

**UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS**



CURSO DE ESPECIALIZACIÓN DE INGENIERIA DE CALIDAD

Diseño e implementación de un plan integral de pruebas manuales y automatizadas para asegurar la calidad de un sistema de comercio electrónico en producción

PRESENTADO POR:

**JENNIFER TATIANA CRUZ CASTRO
MAURICIO DANIEL HERNÁNDEZ JOVEL
NATHALY ADRIANA RODRÍGUEZ MARTÍNEZ**

**PARA OPTAR EL GRADO DE:
INGENIERO DE SISTEMAS INFORMÁTICOS**

CIUDAD UNIVERSITARIA, DICIEMBRE DE 2025

UNIVERSIDAD DE EL SALVADOR

RECTOR:

M.Sc. JUAN ROSA QUINTANILLA

SECRETARIO GENERAL:

LIC. PEDRO ROSALÍO ESCOBAR CASTANEDA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO:

ING. LUIS SALVADOR BARRERA MANCÍA

SECRETARIO:

ARQ. RAÚL ALEXANDER FABIÁN ORELLANA

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

DIRECTOR:

ING. CÉSAR AUGUSTO GONZÁLEZ RODRÍGUEZ

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA ESCUELA DE
INGENIERÍA DE SISTEMAS INFORMÁTICOS

Curso de especialización previo a la opción al grado de:

INGENIERO DE SISTEMAS INFORMÁTICOS

Título:

CURSO DE ESPECIALIZACIÓN EN INGENIERÍA DE CALIDAD

DISEÑO E IMPLEMENTACIÓN DE UN PLAN INTEGRAL DE PRUEBAS
MANUALES Y AUTOMATIZADAS PARA ASEGURAR LA CALIDAD DE
UN SISTEMA DE COMERCIO ELECTRÓNICO EN PRODUCCIÓN

Presentado por:

JENNIFER TATIANA CRUZ CASTRO
MAURICIO DANIEL HERNÁNDEZ JOVEL
NATHALY ADRIANA RODRÍGUEZ MARTÍNEZ

Curso de Especialización aprobado por:

Docente Asesor:

ING. SANDRA GUADALUPE ROMERO HERNÁNDEZ

SAN SALVADOR, DICIEMBRE 2025

Curso de Especialización aprobado por:

Docente asesor:

ING. SANDRA GUADALUPE ROMERO HERNÁNDEZ

Agradecimientos

Agradezco profundamente a Dios por mantenerme firme en cada paso durante este proceso, por su fidelidad en los momentos más difíciles y por darme fuerzas cuando sentí que ya no podía más. Toda la gloria y honra sean a Él.

Infinitas gracias para mi mami, Griselda Rodríguez, por creer en mí, por sus palabras llenas de amor y sus oraciones, que sé que han sido mi soporte constante. A mi abuelita, tías, primas y primos por su apoyo incondicional, y un agradecimiento especial hasta el cielo a mi tío y abuelito.

A mis amigos de la Universidad, Briseida, Gustavo, Gonzalo, Dennis, Sofia, Magaly y Marcelo, gracias por acompañarme en este camino, sin ustedes estos años no habrían sido iguales; y a mis amigas Karen y Odaly, por siempre animarme desde antes de entrar a la U; los quiero y los llevo siempre en mi corazón.

Agradezco grandemente a mis mejores amigos Karla, Stephani y Agustín, por ser mi apoyo cuando sentía que ya no podía más, los quiero muchísimo, que me sean eternos.

A mis compañeros de tesina, Jennifer y Mauricio, gracias por permitirme ser parte del equipo. Con esfuerzo, dedicación y desvelos logramos llegar juntos hasta este punto, mis mejores deseos para ustedes.

Extiendo mi agradecimiento a mi asesora por su tiempo y guía durante todo este proceso, y a los jurados, por su tiempo y atención a nuestro proyecto.

Y, por último, pero no menos importante, a mí misma, por todo el esfuerzo entregado, por no rendirme a pesar de las dificultades y por mantenerme firme hasta estar hoy a un paso de convertirme en ingeniera. Ha sido un proceso de altos y bajos, pero lo he logrado. Gracias.

Nathaly Adriana Rodríguez

Primero que nada, quiero agradecer a Dios por permitirme llegar hasta esta etapa de mi vida y darme la sabiduría para culminar los diversos desafíos intelectuales y emocionales que ha conllevado.

A mis padres, José Mauricio Hernández Rivas y Bera Lot Jovel Rodríguez, quienes siempre brindaron su apoyo incondicional, siempre escuchándome con paciencia, cariño y por su inmenso amor, por enseñarme que los sueños se alcanzan con esfuerzo y fé.

A mis hermanas, Bera Vanessa Hernández Jovel y Sofía Alejandra Hernández Jovel, quienes me acompañaron en tantas noches de desvelo de estudio o de tareas, quedándose a veces sin mi compañía para salidas, llevándome comida a la casa y dándome palabras de aliento para superar situaciones.

Quiero también agradecerme, por cada desvelo, por los planes cancelados, por las salidas no hechas, por las preocupaciones llevadas, por continuar a pesar de sentir que no lo lograría y por no rendirme, confiando en mis capacidades. Reconozco mi esfuerzo y mi valentía para seguir adelante, incluso cuando las dudas me invadían.

A mis amigos que hice en la Universidad Raúl, Eduardo (Jocen), Aldair, Jennifer, quienes durante estos años en más de una ocasión estuvieron codo a codo en las pruebas e hicieron en algunas ocasiones más fácil el camino en ellas. Gracias por los momentos de amistad que se convirtieron en recuerdos valiosos.

A mis compañeras de grupo, Jennifer y Nathaly, porque a pesar del cansancio, las enfermedades y las dificultades, nuestra determinación y apoyo mutuo nos permitieron superar los obstáculos, siempre acompañados de risas y del compromiso de alcanzar juntos la meta

Finalmente, quiero expresar mi agradecimiento a nuestra asesora y a los jurados, por su apoyo y tiempo fundamental para la culminación exitosa de este proyecto.

Mauricio Daniel Hernández Jovel

Primeramente, quiero agradecer a Dios por permitirme llegar hasta esta etapa de mi vida. Su guía ha sido fundamental en cada paso, especialmente en aquellos momentos en los que sentía que no lo lograría. Sin Él, este logro no habría sido posible, pues me dio la fortaleza, sabiduría y la esperanza necesaria para continuar.

A mi mamá, Claudia Castro, quien nunca dudó de mí, ni siquiera en los momentos más inciertos, quien me apoyó y alentó en cada decisión que tomé, escuchándome siempre con paciencia, cariño, por su amor incondicional, por enseñarme que los sueños no se cumplen por suerte, sino con esfuerzo y fe.

A mi abuela, Marta Alarcón, quien me acompañó en tantas noches de desvelo, quedándose a mi lado hasta quedarse dormida, llevándome comida al escritorio, masajeando mis hombros para ayudarme a relajarme, por esas veces en la que se sentaba a escuchar mis preocupaciones, brindándome palabras de aliento. Su dedicación fue un refugio en los momentos difíciles.

Quiero también agradecerme, por cada desvelo, por las veces que comí tarde, por continuar a pesar de sentir que no lo lograría, por no rendirme, confiando en mis capacidades. Reconozco mi esfuerzo y mi valentía para seguir adelante, incluso cuando las dudas me invadían.

Agradecer a mi novio, amigo y confidente, Aldair Villalobos por haber compartido gran parte de la carrera conmigo, por su paciencia, por escucharme sin juzgarme cuando alguna lágrima se escapaba, por sus enseñanzas, consejos, regaños y risas, por toda la ayuda que me brindó en este largo recorrido. Gracias por ser mi compañero en este viaje, por motivarme a seguir y por recordarme que no estaba sola.

A mis compañeros y amigos de estudio (Eduardo, Raúl, Mauricio, Mauri, Héctor, Carlos), quienes durante estos años escucharon mis quejas y me regalaron risas que hicieron más llevadero el camino. Gracias por los momentos de amistad que se convirtieron en recuerdos valiosos.

A mis compañeros de grupo, porque a pesar del cansancio, las enfermedades y las dificultades, nuestra determinación y apoyo mutuo nos permitieron superar los obstáculos, siempre acompañados de risas y del compromiso de alcanzar juntos la meta.

Finalmente, quiero expresar mi agradecimiento a nuestra asesora y a los jurados, por su apoyo y tiempo fundamental para la culminación exitosa de este proyecto.

Jennifer Tatiana Cruz

Tabla de contenido

| | |
|--|----|
| 1. Resumen | 19 |
| 2. Introducción..... | 20 |
| 3. Objetivos..... | 21 |
| 3.1. Objetivo General..... | 21 |
| 3.2. Objetivos específicos..... | 21 |
| 4. Capitulo I: Generalidades | 23 |
| 4.1. Fundamentos de la calidad del software..... | 23 |
| 4.2. Pruebas de software: conceptos esenciales..... | 24 |
| 4.3. Enfoque de ejecución de pruebas | 25 |
| 4.4. Niveles de pruebas..... | 26 |
| 4.4.1. Pruebas unitarias..... | 26 |
| 4.4.2. Pruebas de integración..... | 26 |
| 4.4.3. Pruebas del sistema..... | 26 |
| 4.4.4. Pruebas de aceptación..... | 27 |
| 4.5. Tipos de pruebas | 27 |
| 4.5.1. Pruebas funcionales | 27 |
| 4.5.2. Pruebas no funcionales | 27 |
| 4.5.3. Pruebas de regresión..... | 28 |
| 4.5.4. Pruebas manuales | 28 |
| 4.5.5. Pruebas automatizadas..... | 29 |
| 5. Capitulo II: Plan de pruebas | 31 |
| 5.1. Definición de plan de pruebas | 31 |
| 5.2. Alcance del plan de pruebas | 31 |
| 5.3. Módulos no considerados | 32 |
| 5.4. Selección de pruebas automatizadas..... | 33 |
| 5.5. Resumen visual..... | 34 |
| 5.6. Criterios para la selección de herramientas | 34 |
| 5.7. Estándar IEEE-829 | 35 |
| 5.8. Riesgos | 37 |
| 5.9. Características que se probarán | 38 |

| | |
|---|----|
| 5.10. Alcance (estrategia) | 38 |
| 5.11. Criterios de aprobación/fallo/ejecución con hallazgos | 39 |
| 5.12. Entregables de prueba..... | 40 |
| 5.13. Preparación técnica y metodológica | 41 |
| 5.14. Ambientes y recursos de prueba | 42 |
| 5.14.1. Recurso de software/hardware..... | 42 |
| 5.14.2. Entorno técnico..... | 43 |
| 5.14.3. Instrumentos técnicos del proceso de testing | 43 |
| 5.14.4. Configuración de herramientas de pruebas | 44 |
| 5.15. Responsabilidades | 48 |
| 5.16. Calendario..... | 50 |
| 5.17. Plan de riesgos y contingencias..... | 52 |
| 5.17.1. Introducción..... | 52 |
| 5.17.2. Alcance | 52 |
| 5.17.3. Objetivos..... | 52 |
| 5.17.4. Equipo de trabajo y roles | 53 |
| 5.17.5. Criterios de evaluación de riesgos | 53 |
| 5.17.6. Plan de comunicación..... | 53 |
| 5.17.7. Identificación y evaluación de riesgos..... | 54 |
| 5.17.8. Matriz de probabilidad e impacto | 56 |
| 5.18. Aprobaciones | 56 |
| 6. Capitulo III: Caso de estudio | 58 |
| 6.1. Antecedentes..... | 58 |
| 6.2. Modelado de negocio..... | 59 |
| 6.3. Reglas de negocio..... | 61 |
| 6.4. Necesidades del negocio..... | 62 |
| 6.5. Levantamiento de requerimientos de pruebas | 64 |
| 6.5.1. Criterios para el levantamiento de requerimientos | 64 |
| 6.6. Pruebas funcionales y no funcionales..... | 66 |
| 6.6.1. Requerimientos funcionales | 66 |
| 6.6.2. Requerimientos no funcionales | 70 |

| | |
|---|-----|
| 6.7. Casos a automatizar | 71 |
| 6.8. Casos que no se van automatizar. | 76 |
| 6.9. Cobertura de pruebas | 77 |
| 6.10. Alcance de pruebas | 79 |
| 6.11. Estrategia de pruebas | 80 |
| 6.11.1. Enfoque de ejecución: pruebas manuales vs automatizadas..... | 81 |
| 6.12. Análisis de riesgos | 82 |
| 6.13. Roles y responsabilidades..... | 83 |
| 6.14. Calendario de pruebas | 84 |
| 6.15. Ejecución CI/CD | 85 |
| 6.16. Entorno de prueba..... | 88 |
| 6.17. Registro de ejecución de pruebas | 91 |
| 6.18. Entregables de pruebas automatizada y no automatizadas | 92 |
| 6.19. Matriz de pruebas | 93 |
| 6.20. Test Readiness Review (TRR)..... | 103 |
| 6.21. Estimaciones..... | 105 |
| 6.22. Métricas de pruebas | 108 |
| 6.23. Criterios de entrada y salida | 109 |
| 6.24. Procedimiento para la ejecución de prueba | 111 |
| 6.25. Métricas de pruebas Resultados de ejecución de las pruebas..... | 114 |
| 6.26. Resultados de ejecución de las pruebas..... | 115 |
| 6.27. Pruebas de código | 192 |
| 6.27.1. Análisis de calidad del código | 193 |
| 6.27.2. Conteo de incidencias..... | 195 |
| 6.27.3. Rango de métricas | 195 |
| 6.28. Informe de defectos/hallazgos encontrados..... | 198 |
| 6.28.1. Resumen | 198 |
| 6.28.2. Objetivo | 198 |
| 6.28.3. Metodología..... | 199 |
| 6.28.4. Hallazgos | 200 |
| 6.28.5. Defectos más críticos y su impacto | 206 |

| | |
|--|-----|
| 6.28.6. Patrones de error identificados | 207 |
| 6.28.7. Análisis de hallazgos de los módulos con bajo desempeño..... | 209 |
| 6.28.8. Conclusiones..... | 209 |
| 6.28.9. Recomendaciones | 210 |
| 6.29. Informe de cierre de pruebas | 213 |
| 6.29.1. Resumen | 213 |
| 6.29.2. Objetivo | 214 |
| 6.29.3. Alcance de las pruebas..... | 214 |
| 6.29.4. Estrategia y metodología | 215 |
| 6.29.5. Consolidado de resultados | 216 |
| 6.29.6. Resultados de las pruebas por Sprint..... | 219 |
| 6.29.7. Resultados por módulos | 222 |
| 6.29.9. Resultados globales | 228 |
| 6.29.10. Análisis de métricas | 229 |
| 6.29.11. Conclusiones..... | 230 |
| 6.29.12. Recomendaciones sobre los casos ejecutados | 231 |
| 7. Recomendaciones | 232 |
| 8. Lecciones aprendidas..... | 233 |
| 9. Conclusiones..... | 234 |
| 10. Carta de Salida..... | 236 |
| 11. Referencias | 238 |
| 12. Anexos | 239 |
| 12.1. Anexo 1: Resultados del análisis estático de código | 239 |
| 12.2. Anexo 2: Análisis de caja negra..... | 244 |
| 12.3. Anexo 3: Diagramas y flujos del sistema | 245 |
| 12.4. Anexo 4: Requerimientos, casos de prueba y resultados..... | 253 |
| 12.5. Anexo 5: Presentación de acta de aceptación de matriz de pruebas y resultado ... | 277 |
| 13. Glosario | 279 |

Índice de tablas

| | |
|--|----|
| Tabla 1. Comparativa entre pruebas manuales y automatizadas. | 25 |
| Tabla 2. Módulos considerados en el plan de pruebas (Parte I). | 31 |
| Tabla 3. Módulos considerados en el plan de pruebas (Parte II). | 32 |
| Tabla 4. Módulos no incluidos en el plan de pruebas. | 32 |
| Tabla 5. Número de casos a automatizables/no automatizables. | 34 |
| Tabla 6. Riesgos potenciales durante la planificación y ejecución del testing..... | 37 |
| Tabla 7. Características a probar..... | 38 |
| Tabla 8. Recurso de software y hardware a utilizar durante el proceso de pruebas. | 42 |
| Tabla 9. Resumen de componentes tecnológicos del sistema. | 43 |
| Tabla 10. Herramientas de pruebas a utilizar y sus versiones. | 43 |
| Tabla 11. Calendario para el plan de pruebas. | 50 |
| Tabla 12. Equipo responsable de actividades a desarrollar. | 53 |
| Tabla 13. Identificación de riesgos, mitigación y contingencia (Parte I). | 54 |
| Tabla 14. Identificación de riesgos, mitigación y contingencia (Parte II). | 55 |
| Tabla 15. Registro de aprobaciones. | 56 |
| Tabla 16. Actores principales del sistema e-commerce. | 59 |
| Tabla 17. Detalle de los módulos principales del sistema. | 60 |
| Tabla 18. Análisis FODA..... | 63 |
| Tabla 19. Requerimientos de autenticación de usuarios..... | 66 |
| Tabla 20. Requerimientos por módulo/submódulos – Vista administrador (Parte I). | 67 |
| Tabla 21. Requerimientos por módulo/submódulos – Vista administrador (Parte II). | 68 |
| Tabla 22. Requerimientos por módulo/submódulos – Vista seller. | 69 |
| Tabla 23. Requerimientos por módulo/submódulos – Vista customer..... | 70 |
| Tabla 24. Requerimientos no funcionales del sistema..... | 70 |
| Tabla 25. Casos de prueba con automatización completa (Parte I). | 71 |
| Tabla 26. Casos de prueba con automatización completa (Parte II). | 72 |
| Tabla 27. Casos de prueba con automatización completa (Parte III). | 73 |
| Tabla 28. Casos de prueba con automatización completa (Parte IV). | 74 |
| Tabla 29. Casos de prueba con automatización completa (Parte V). | 75 |
| Tabla 30. Casos de prueba con automatización completa (Parte VI). | 76 |
| Tabla 31. Casos de prueba no automatizables. | 77 |
| Tabla 32. Resumen de tipos de pruebas aplicados..... | 78 |
| Tabla 33. Definición de módulos críticos. | 80 |
| Tabla 34. Roles, responsabilidades y funciones del equipo de trabajo..... | 83 |
| Tabla 35. Diseño de casos de prueba (Parte I). | 93 |
| Tabla 36. Diseño de casos de prueba (Parte II). | 94 |

| | |
|---|-----|
| Tabla 37. Diseño de casos de prueba (Parte III)..... | 95 |
| Tabla 38. Diseño de casos de prueba (Parte IV)..... | 96 |
| Tabla 39. Diseño de casos de prueba (Parte V)..... | 97 |
| Tabla 40. Diseño de casos de prueba (Parte VI)..... | 98 |
| Tabla 41. Diseño de casos de prueba (Parte VII)..... | 99 |
| Tabla 42. Diseño de casos de prueba (Parte VIII)..... | 100 |
| Tabla 43. Diseño de casos de prueba (Parte IX)..... | 101 |
| Tabla 44. Diseño de casos de prueba (Parte X)..... | 102 |
| Tabla 45. Test Readiness Review | 104 |
| Tabla 46. Criterios de entrada..... | 109 |
| Tabla 47. Criterios de salida | 110 |
| Tabla 48. Caso de prueba 01..... | 115 |
| Tabla 49. Caso de prueba 02..... | 116 |
| Tabla 50. Caso de prueba 03..... | 117 |
| Tabla 51. Caso de prueba 04..... | 118 |
| Tabla 52. Caso de prueba 05..... | 119 |
| Tabla 53. Caso de prueba 06..... | 120 |
| Tabla 54. Caso de prueba 07..... | 121 |
| Tabla 55. Caso de prueba 08..... | 122 |
| Tabla 56. Caso de prueba 09..... | 123 |
| Tabla 57. Caso de prueba 10..... | 124 |
| Tabla 58. Caso de prueba 11..... | 125 |
| Tabla 59. Caso de prueba 12..... | 126 |
| Tabla 60. Caso de prueba 13..... | 127 |
| Tabla 61. Caso de prueba 14..... | 128 |
| Tabla 62. Caso de prueba 15..... | 129 |
| Tabla 63. Caso de prueba 16..... | 130 |
| Tabla 64. Caso de prueba 17..... | 131 |
| Tabla 65. Caso de prueba 18..... | 132 |
| Tabla 66. Caso de prueba 19..... | 133 |
| Tabla 67. Caso de prueba 20..... | 134 |
| Tabla 68. Caso de prueba 21..... | 135 |
| Tabla 69. Caso de prueba 22..... | 136 |
| Tabla 70. Caso de prueba 23..... | 137 |
| Tabla 71. Caso de prueba 24..... | 138 |
| Tabla 72. Caso de prueba 25..... | 139 |
| Tabla 73. Caso de prueba 26..... | 140 |
| Tabla 74. Caso de prueba 27..... | 141 |

| | |
|-----------------------------------|-----|
| Tabla 75. Caso de prueba 28..... | 142 |
| Tabla 76. Caso de prueba 29..... | 143 |
| Tabla 77. Caso de prueba 30..... | 144 |
| Tabla 78. Caso de prueba 31..... | 145 |
| Tabla 79. Caso de prueba 32..... | 146 |
| Tabla 80. Caso de prueba 33..... | 147 |
| Tabla 81. Caso de prueba 34..... | 148 |
| Tabla 82. Caso de prueba 35..... | 149 |
| Tabla 83. Caso de prueba 36..... | 150 |
| Tabla 84. Caso de prueba 37..... | 151 |
| Tabla 85. Caso de prueba 38..... | 152 |
| Tabla 86. Caso de prueba 39..... | 153 |
| Tabla 87. Caso de prueba 40..... | 154 |
| Tabla 88. Caso de prueba 41..... | 155 |
| Tabla 89. Caso de prueba 42..... | 156 |
| Tabla 90. Caso de prueba 43..... | 157 |
| Tabla 91. Caso de prueba 44..... | 158 |
| Tabla 92. Caso de prueba 45..... | 159 |
| Tabla 93. Caso de prueba 46..... | 160 |
| Tabla 94. Caso de prueba 47..... | 161 |
| Tabla 95. Caso de prueba 48..... | 162 |
| Tabla 96. Caso de prueba 49..... | 163 |
| Tabla 97. Caso de prueba 50..... | 164 |
| Tabla 98. Caso de prueba 51..... | 165 |
| Tabla 99. Caso de prueba 52..... | 166 |
| Tabla 100. Caso de prueba 53..... | 167 |
| Tabla 101. Caso de prueba 54..... | 168 |
| Tabla 102. Caso de prueba 55..... | 169 |
| Tabla 103. Caso de prueba 56..... | 170 |
| Tabla 104. Caso de prueba 57..... | 171 |
| Tabla 105. Caso de prueba 58..... | 172 |
| Tabla 106. Caso de prueba 59..... | 173 |
| Tabla 107. Caso de prueba 60..... | 174 |
| Tabla 108. Caso de prueba 61..... | 175 |
| Tabla 109. Caso de prueba 62..... | 176 |
| Tabla 110. Caso de prueba 63..... | 177 |
| Tabla 111. Caso de prueba 64..... | 178 |
| Tabla 112. Caso de prueba 65..... | 179 |

| | |
|--|-----|
| Tabla 113. Caso de prueba 66..... | 180 |
| Tabla 114. Caso de prueba 67..... | 181 |
| Tabla 115. Caso de prueba 68..... | 182 |
| Tabla 116. Caso de prueba 69..... | 183 |
| Tabla 117. Caso de prueba 70..... | 184 |
| Tabla 118. Caso de prueba 71..... | 185 |
| Tabla 119. Caso de prueba 72..... | 186 |
| Tabla 120. Caso de prueba 73..... | 187 |
| Tabla 121. Caso de prueba 74..... | 188 |
| Tabla 122. Caso de prueba 75..... | 189 |
| Tabla 123. Caso de prueba 76..... | 190 |
| Tabla 124. Caso de prueba 77..... | 191 |
| Tabla 125. Conteo de incidencias por severidad y tipo. | 195 |
| Tabla 126. Rango de métricas..... | 196 |
| Tabla 127. Defectos clasificados por estado..... | 200 |
| Tabla 128. Detalle de defectos encontrados (Parte I)..... | 201 |
| Tabla 129. Detalle de defectos encontrados (Parte II)..... | 202 |
| Tabla 130. Detalle de defectos encontrados (Parte III)..... | 203 |
| Tabla 131. Detalle de defectos encontrados (Parte IV)..... | 204 |
| Tabla 132. Detalle de defectos encontrados (Parte V)..... | 205 |
| Tabla 133. Defectos con mayor criticidad y su impacto (Parte I). | 206 |
| Tabla 134. Defectos con mayor criticidad y su impacto (Parte II). | 207 |
| Tabla 135. Patrones de error y causas (Parte I). | 208 |
| Tabla 136. Patrones de error y causas (Parte II). | 208 |
| Tabla 137. Patrones de error y causas (Parte II). | 209 |
| Tabla 138. Módulos evaluados en el ciclo de pruebas. | 215 |
| Tabla 139. Consolidado de resultados de casos de prueba (Parte I). | 216 |
| Tabla 140. Consolidado de resultados de casos de prueba (Parte II). | 217 |
| Tabla 141. Consolidado de resultados de casos de prueba (Parte III). | 218 |
| Tabla 142. Resumen de ejecución de casos de pruebas – Sprint 1..... | 219 |
| Tabla 143. Resumen de ejecución de casos de pruebas – Sprint 2..... | 220 |
| Tabla 144. Resumen de ejecución de casos de pruebas – Sprint 3..... | 221 |
| Tabla 145. Resumen de ejecución de casos de pruebas por módulo (Parte I). | 222 |
| Tabla 146. Resumen de ejecución de casos de pruebas por módulo (Parte II). | 223 |
| Tabla 147. Resumen de ejecución de casos de pruebas por módulo (Parte III). | 224 |
| Tabla 148. Resumen de ejecución de casos de pruebas por módulo (Parte IV). | 225 |
| Tabla 149. Resumen de ejecución de casos de pruebas por módulo (Parte V). | 226 |
| Tabla 150. Resumen de ejecución de casos de pruebas por módulo (Parte VI). | 227 |

| | |
|---|-----|
| Tabla 151. Resumen general de ejecución de casos. | 228 |
| Tabla 152. Porcentaje de defectos encontrados por módulos. | 230 |
| Tabla 153. Issues List – SonarQube (Parte I). | 239 |
| Tabla 154. Issues List – SonarQube (Parte II). | 240 |
| Tabla 155. Issues List – SonarQube (Parte III). | 241 |
| Tabla 156. Issues List – SonarQube (Parte IV). | 242 |
| Tabla 157. Issues List – SonarQube (Parte V). | 243 |
| Tabla 158. Requerimientos del sistema (Parte I). | 253 |
| Tabla 159. Requerimientos del sistema (Parte II). | 254 |
| Tabla 160. Requerimientos del sistema (Parte III). | 255 |
| Tabla 161. Requerimientos del sistema (Parte IV). | 256 |
| Tabla 162. Matriz de resultados (Parte I). | 257 |
| Tabla 163. Matriz de resultados (Parte II). | 258 |
| Tabla 164. Matriz de resultados (Parte III). | 259 |
| Tabla 165. Matriz de resultados (Parte IV). | 260 |
| Tabla 166. Matriz de resultados (Parte V). | 261 |
| Tabla 167. Matriz de resultados (Parte VI). | 262 |
| Tabla 168. Matriz de resultados (Parte VII). | 263 |
| Tabla 169. Matriz de resultados (Parte VIII). | 264 |
| Tabla 170. Matriz de resultados (Parte IX). | 265 |
| Tabla 171. Matriz de resultados (Parte X). | 266 |
| Tabla 172. Matriz de resultados (Parte XI). | 267 |
| Tabla 173. Matriz de resultados (Parte XII). | 268 |
| Tabla 174. Matriz de resultados (Parte XII). | 269 |
| Tabla 175. Matriz de resultados (Parte XIV). | 270 |
| Tabla 176. Matriz de resultados (Parte XV). | 271 |
| Tabla 177. Matriz de resultados (Parte XVI). | 272 |
| Tabla 178. Matriz de resultados (Parte XVII). | 273 |
| Tabla 179. Matriz de resultados (Parte XVIII). | 274 |
| Tabla 180. Matriz de resultados (Parte XIX). | 275 |
| Tabla 181. Matriz de resultados (Parte XX). | 276 |

Índice de ilustraciones

| | |
|--|-----|
| Ilustración 1. Instalación de Laravel Dusk como dependencia de desarrollo. | 44 |
| Ilustración 2. Verificación de la versión de PHPUnit instalada..... | 45 |
| Ilustración 3. Verificación de la versión instalada de Selenium en entorno Python..... | 45 |
| Ilustración 4. Inicio del asistente de instalación de OWASP ZAP..... | 46 |
| Ilustración 5. Interfaz principal de Apache JMeter..... | 47 |
| Ilustración 6. Proceso de arranque de SonarQube..... | 47 |
| Ilustración 7. Configuración para análisis con Qualys SSL Labs | 48 |
| Ilustración 8. Matriz de evaluación de riesgos por impacto vs probabilidad. | 56 |
| Ilustración 9. Calendario de pruebas | 84 |
| Ilustración 10. Arquitectura CI/CD. | 85 |
| Ilustración 11. Modificación de repositorio..... | 86 |
| Ilustración 12. Ejecución de pruebas desde Github Actions. | 87 |
| Ilustración 13. Envío de Resultados de cada una de las pruebas..... | 88 |
| Ilustración 14. Interfaz de autenticación del sistema: Importaciones Guadalajara. | 90 |
| Ilustración 15. Interfaz de ordenes: Importaciones Guadalajara. | 90 |
| Ilustración 16. Porcentaje de incidencias por categoría. | 192 |
| Ilustración 17. Porcentaje de issues por severidad. | 193 |
| Ilustración 18. Distribución de defectos por módulo | 200 |
| Ilustración 19. Total de casos de pruebas ejecutados – Sprint 1..... | 219 |
| Ilustración 20. Total de casos por estado – Sprint 1 | 219 |
| Ilustración 21. Total de casos de pruebas ejecutados – Sprint 2..... | 220 |
| Ilustración 22. Total de casos por estado – Sprint 1 | 220 |
| Ilustración 23. Total de casos de pruebas ejecutados – Sprint 3..... | 221 |
| Ilustración 24. Total de casos por estado – Sprint 3. | 221 |
| Ilustración 25. Resumen general de ejecución de casos..... | 228 |
| Ilustración 26. Diagrama de la caja negra. | 244 |
| Ilustración 27. Diagrama de la arquitectura del sistema..... | 245 |
| Ilustración 28. Diagrama de los módulos del sistema | 245 |
| Ilustración 29. Flujo del módulo Customers | 246 |
| Ilustración 30. Flujo del módulo Products | 247 |
| Ilustración 31. Flujo del módulo Users | 248 |
| Ilustración 32. Flujo del módulo Orsers | 249 |
| Ilustración 33. Flujo del módulo Inventory. | 250 |
| Ilustración 34. Flujo del módulo Category. | 251 |
| Ilustración 35. Flujo del módulo crear una orden..... | 252 |
| Ilustración 36. Acta de Aceptación de Pruebas y Resultados - Parte I. | 277 |
| Ilustración 37. Acta de Aceptación de Pruebas y Resultados - Parte II..... | 278 |

Índice de ecuaciones

| | |
|--|-----|
| Ecuación 1. Método de PERT..... | 105 |
| Ecuación 2. Porcentaje de casos de pruebas ejecutados..... | 108 |
| Ecuación 3. Porcentaje de casos de pruebas aprobados..... | 108 |
| Ecuación 4. Porcentaje de casos de pruebas fallidos..... | 108 |
| Ecuación 5. Porcentaje de casos de pruebas fallidos..... | 109 |
| Ecuación 6. Porcentaje de defectos encontrados por módulo..... | 109 |
| Ecuación 7. Resultados de cobertura de pruebas ejecutadas..... | 229 |
| Ecuación 8. Resultados de las pruebas aprobadas..... | 229 |
| Ecuación 9. Resultados de las pruebas fallidas..... | 229 |
| Ecuación 10. Resultados de las pruebas ejecutadas con hallazgos..... | 229 |

1. Resumen

El presente sistema, el cual fue desarrollado y es mantenido por un único desarrollador independiente, sin la implementación de procesos formales que aseguren la calidad. Aunque el sistema está en producción y es completamente funcional *Veasé el diagrama Anexo 3: Diagramas del Sistema*, presenta una baja cobertura de pruebas, ya que solo se han ejecutado pruebas manuales por parte del desarrollador sin documentación técnica adecuada, sin automatización y seguimiento de métricas de calidad, lo cual limita la capacidad para detectar defectos, evaluar rendimiento o anticipar problemas funcionales, lo que representa un riesgo operativo ante cambios, mantenimientos o crecimiento del sistema. Para abordar esta problemática, se propone utilizar el método de la caja negra, que permite analizar el comportamiento del sistema observando únicamente las entradas y salidas, sin acceder a la lógica interna. Esto es útil cuando no se cuenta con documentación técnica o acceso directo al código fuente, permitiendo evaluar funcionalidad, rendimiento y experiencia del usuario *Veasé el diagrama Anexo 2: Diagrama de Caja Negra*.

Este trabajo presenta el diseño e implementación de un plan integral de pruebas para un sistema de comercio electrónico en producción, combinando enfoques manuales y automatizados. El objetivo principal es identificar defectos funcionales, optimizar el rendimiento y garantizar una experiencia de usuario confiable.

La metodología se basó en el estándar IEEE-828, utilizando la metodología ágil Scrum para estructurar las actividades en cuatro sprints. Se desarrollaron suites de pruebas manuales, scripts automatizados y pruebas de carga con JMeter, cubriendo los módulos críticos del sistema: productos, órdenes, inventario, clientes y usuarios.

Los resultados incluyen la ejecución de una matriz de pruebas funcionales, la detección de defectos en procesos clave, la obtención de métricas de rendimiento (tiempos de respuesta menores a 2 segundos y tasa de error inferior al 0,5 %) y un incremento significativo en la cobertura de pruebas.

Como conclusión, se establece que la integración de pruebas manuales y automatizadas en proyectos con recursos limitados es factible y altamente beneficiosa, permitiendo mejorar la calidad técnica del software, reducir riesgos en producción y generar confianza en los usuarios finales.

2. Introducción

En la actualidad, el comercio electrónico se ha consolidado como uno de los pilares del crecimiento económico global. La necesidad de plataformas seguras, eficientes y confiables ha generado un aumento en la complejidad de los sistemas, lo que exige procesos de aseguramiento de calidad más rigurosos. Sin embargo, muchos proyectos de desarrollo, especialmente aquellos llevados a cabo por programadores independientes o equipos reducidos, enfrentan limitaciones en la incorporación de prácticas formales de prueba, lo cual incrementa el riesgo de errores en producción, problemas de rendimiento y baja satisfacción del usuario final.

El presente trabajo tiene como propósito diseñar e implementar un plan integral de pruebas de software que combine enfoques manuales y automatizados, aplicados a un sistema de comercio electrónico ya en producción. La propuesta se fundamenta en metodologías ágiles, particularmente Scrum, y en estándares internacionales de calidad como ISO/IEC 25010 e IEEE-828, que permiten establecer métricas objetivas para evaluar la funcionalidad, seguridad, usabilidad y rendimiento del sistema.

La investigación aborda tanto el análisis del contexto actual del sistema, como la definición de requerimientos, la selección de herramientas de prueba y la elaboración de una matriz de casos aplicados a los módulos críticos (productos, órdenes, inventario, clientes y usuarios). Con ello se busca generar evidencia empírica de la mejora en la calidad del software, reducir defectos en producción, optimizar el rendimiento operativo y ofrecer un modelo replicable para proyectos similares.

En suma, esta tesina no solo se orienta a garantizar la estabilidad y confiabilidad del sistema en estudio, sino que también pretende aportar un marco estratégico y práctico que fortalezca la competitividad de desarrolladores y pequeñas empresas en el mercado tecnológico

3. Objetivos

3.1. Objetivo General

Diseñar e implementar un plan integral de pruebas de calidad para un sistema de comercio electrónico, combinando pruebas manuales y automatizadas en módulos críticos. Con el fin de identificar defectos funcionales, optimizar el rendimiento operativo y garantizar una experiencia de usuario fluida, fortaleciendo la estabilidad del sistema y contribuyendo a la confianza del usuario final.

3.2. Objetivos específicos

1. Identificar y priorizar los flujos de negocio de los módulos críticos con el fin de focalizar el esfuerzo de prueba en las funcionalidades de mayor valor y exposición.
2. Diseñar y documentar suites de pruebas automatizadas que cubran los requisitos funcionales clave y los escenarios de uso más frecuentes, asegurando la trazabilidad requisito
3. Desarrollar e implementar pruebas automatizadas que permitan validar procesos repetitivos y de alto impacto, con el fin de reducir el tiempo de prueba, aumentar la cobertura y detectar errores de forma oportuna en cada actualización del sistema.
4. Evaluar el rendimiento y la estabilidad del sistema bajo diferentes condiciones de uso, mediante pruebas de carga y estrés, asegurando tiempos de respuesta óptimos y una navegación fluida para el usuario final.
5. Verificar la usabilidad, seguridad y consistencia de la interfaz de usuario, garantizando que los procesos sean intuitivos, seguros y coherentes, con el fin de fortalecer la confianza y satisfacción del usuario en cada interacción con la plataforma.
6. Redactar un informe que exponga los resultados obtenidos del sistema E-commerce para mejorar la toma de decisiones a futura y generar oportunidades de mejora.

CAPITULO I

GENERALIDADES DE LA INGENIERIA DE CALIDAD

4. Generalidades

4.1. Fundamentos de la calidad del software

La calidad del software constituye un eje fundamental dentro de la ingeniería de sistemas, ya que de ella depende que los productos digitales puedan cumplir con los requerimientos establecidos y responder a las expectativas de los usuarios. En términos generales, se entiende como el grado en que un sistema, componente o proceso satisface tanto los requisitos explícitos, funcionales y no funcionales, como las necesidades implícitas que emergen en el uso cotidiano. De acuerdo con la definición del IEEE, la calidad del software refleja no solo el cumplimiento de especificaciones técnicas, sino también la capacidad de proporcionar confianza, seguridad y valor en la práctica real.

El concepto de calidad en el software no es absoluto, sino multidimensional. Tradicionalmente, se han identificado diferentes perspectivas que ayudan a comprenderlo: desde el enfoque orientado al usuario, que la mide a partir de la experiencia y satisfacción; hasta el enfoque basado en el producto, que examina atributos como la fiabilidad, la eficiencia y la usabilidad. Cada una de estas perspectivas aporta una visión complementaria y necesaria para evaluar de forma integral la calidad de un sistema de software.

En la práctica, la gestión de la calidad se articula a través de actividades de prevención, detección y corrección. La prevención busca establecer normas, lineamientos y buenas prácticas desde las primeras fases de desarrollo, con el fin de evitar la introducción de errores. La detección, por su parte, se centra en inspecciones, revisiones y pruebas sistemáticas que permiten identificar defectos antes de la entrega del producto al usuario final. Finalmente, la corrección consiste en la eliminación de errores detectados y en la incorporación de mecanismos de mejora continua que fortalezcan el producto a lo largo de su ciclo de vida.

El desarrollo de la ingeniería de software ha permitido contar con modelos y estándares que proporcionan un marco de referencia objetivo para evaluar y garantizar la calidad. Entre ellos, la norma ISO/IEC 25010 destaca por definir ocho características fundamentales: adecuación funcional, rendimiento y eficiencia, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad. Cada una de estas dimensiones aporta criterios medibles que facilitan la evaluación del producto.

La importancia de garantizar la calidad del software radica en los beneficios tangibles que genera. Un producto de calidad reduce costos de mantenimiento, previene fallos críticos, mejora la satisfacción de los usuarios y asegura la continuidad operativa del negocio. Por ello, adoptar prácticas de calidad desde el inicio del desarrollo no es un lujo, sino una necesidad que asegura la sostenibilidad y competitividad del sistema en el mercado.

En el caso particular de este proyecto, los fundamentos de la calidad del software se convierten en el punto de partida para el diseño del plan de pruebas, pero no solo como conceptos teóricos, sino como criterios que guiaron decisiones concretas. A partir de estos principios, y en conjunto con el usuario final, se identificaron los módulos que tenían mayor impacto en las operaciones diarias, como Products, Orders, Inventory y Login. Esto permitió enfocar el esfuerzo en las áreas donde la fiabilidad, la adecuación funcional y la seguridad son más críticas para el negocio. De esta manera, el proyecto demuestra que incluso en contextos de recursos limitados, como el de un desarrollador independiente, es posible aplicar prácticas reconocidas y orientarlas a las necesidades reales del sistema para garantizar estabilidad, eficiencia y una mejor experiencia de usuario.

4.2. Pruebas de software: conceptos esenciales

Las pruebas de software constituyen una de las actividades más importantes dentro del ciclo de vida de desarrollo, pues representan el mecanismo mediante el cual se valida que un sistema cumple con los requerimientos establecidos y se comporta conforme a lo esperado en diversos escenarios. Su propósito no se limita únicamente a encontrar defectos, sino también a confirmar que el software satisface las necesidades del usuario final y que ofrece un nivel aceptable de calidad, fiabilidad y rendimiento.

El valor de las pruebas radica en su capacidad para reducir la incertidumbre. Cada caso ejecutado, ya sea de forma manual o automatizada, brinda evidencia acerca del estado del sistema, permitiendo identificar errores antes de que lleguen a producción. Esto genera un impacto directo en la eficiencia y en la reducción de costos, ya que reparar un defecto en etapas tempranas resulta considerablemente menos oneroso que corregirlo una vez que el producto está en manos de los usuarios. Asimismo, las pruebas incrementan la confianza de todas las partes interesadas, al ofrecer resultados medibles sobre el nivel de cumplimiento y la estabilidad del software.

Las pruebas también permiten clasificar la evaluación del sistema en diferentes niveles. En una primera capa se encuentran las pruebas unitarias, que verifican de manera aislada las funciones o métodos individuales. Posteriormente, las pruebas de integración validan la comunicación y cooperación entre diferentes módulos, asegurando que trabajen en conjunto sin inconsistencias. Más adelante, las pruebas de sistema evalúan el producto completo bajo condiciones similares a las del entorno real, comprobando su funcionamiento de extremo a extremo.

De manera complementaria, el aseguramiento de la calidad contempla distintos tipos de pruebas según los objetivos perseguidos. Las pruebas funcionales, por ejemplo, garantizan que las características del sistema se ejecuten de acuerdo con los requerimientos definidos. En cambio, las pruebas no funcionales evalúan aspectos como el rendimiento, la seguridad, la escalabilidad y la usabilidad, factores que inciden directamente en la experiencia del

usuario. Existen además pruebas de regresión, que buscan confirmar que las nuevas modificaciones no afecten funcionalidades previamente desarrolladas, y pruebas exploratorias, en las que el tester interactúa libremente con la aplicación para descubrir comportamientos inesperados.

El carácter esencial de las pruebas radica en que abarcan tanto la dimensión técnica como la experiencia del usuario. Por un lado, verifican la lógica interna y la estabilidad del software; por el otro, garantizan que el producto sea intuitivo, seguro y confiable. De esta forma, las pruebas no deben considerarse un proceso aislado o secundario, sino una práctica continua e integrada al desarrollo, capaz de generar evidencia concreta sobre la calidad del sistema.

En el caso del sistema evaluado, estos conceptos orientaron la forma en que se diseñó la estrategia de pruebas. La distinción entre pruebas funcionales, no funcionales y de regresión permitió estructurar un plan que cubriera los comportamientos principales del sistema, especialmente aquellos señalados por el usuario como críticos. Así, cada caso de prueba no solo corroboró requisitos documentados, sino también situaciones reales de uso identificadas durante la fase de levantamiento con el desarrollador.

4.3. Enfoque de ejecución de pruebas

El enfoque de ejecución de pruebas representa una etapa importante dentro del proceso de aseguramiento de la calidad del software, pues permite verificar que el sistema cumple con los requerimientos establecidos y que funciona de manera adecuada bajo diferentes escenarios. Este proceso puede realizarse de forma manual o automatizada, según los objetivos del proyecto, el tiempo disponible y los recursos técnicos con los que se cuente.

- **Comparativa:** Para elegir el enfoque adecuado de pruebas, es importante comprender las diferencias clave entre las pruebas manuales y las automatizadas.

| Criterio | Pruebas manuales | Pruebas automatizadas |
|-------------------------------|---------------------------|------------------------|
| Costo Inicial | Bajo | Alto |
| Velocidad de ejecución | Lenta | Rápida |
| Escalabilidad | Limitada | Alta |
| Detección UX/UI | Alta | Baja |
| Detección de errores | Usabilidad, diseño | Regresión, rendimiento |
| Mantenimiento | Bajo | Alto |
| Adaptabilidad | Alta (cambios frecuentes) | Requiere mantenimiento |
| Reusabilidad | Baja | Alta |

Tabla 1. Comparativa entre pruebas manuales y automatizadas.

En ambos enfoques, este tipo de pruebas son complementarios. Por un lado, tenemos las pruebas manuales siendo valiosas en fases exploratorias o cuando se requiere juicio humano, mientras que las pruebas automatizadas son ideales para asegurar consistencia en pruebas repetitivas y facilitar la integración continua en entornos de desarrollo ágil.

En nuestro caso, la elección del enfoque surgió de las condiciones reales: el tiempo disponible, los recursos técnicos y la necesidad del usuario de obtener resultados confiables. Esto llevó a combinar pruebas manuales con pruebas automatizadas. Esta combinación permitió optimizar el esfuerzo y reducir tiempos sin comprometer la calidad.

4.4. Niveles de pruebas

El proceso de pruebas de software se organiza en diferentes niveles, los cuales permiten evaluar la calidad del sistema de manera progresiva. Cada nivel tiene un propósito específico que va desde la validación de pequeñas unidades de código hasta la verificación del sistema completo en condiciones cercanas a las reales.

4.4.1. Pruebas unitarias

Se enfocan en validar componentes individuales del código (funciones, métodos o clases) de manera aislada. Generalmente son ejecutadas por los desarrolladores para confirmar que cada pieza funciona correctamente de forma independiente.

4.4.2. Pruebas de integración

Evalúan la interacción entre módulos o componentes, verificando que al combinarse trabajen de forma adecuada. Este nivel permite detectar problemas en las conexiones entre partes del software.

4.4.3. Pruebas del sistema

Consisten en verificar el comportamiento del sistema como un todo, comprobando el funcionamiento de extremo a extremo en un entorno que simula el uso real. Se busca garantizar que todas las funcionalidades operen juntas sin conflictos.

4.4.4. Pruebas de aceptación

Validan que el software cumpla con los requerimientos y expectativas del cliente o usuario final. En este nivel suelen participar stakeholders o usuarios clave, quienes confirman si el producto está listo para su implementación o despliegue.

4.5. Tipos de pruebas

El proceso de pruebas de software abarca diversas categorías que evalúan el sistema desde diferentes perspectivas. Estas pruebas buscan garantizar que el producto cumpla con los requisitos establecidos, funcione correctamente y que mantenga su calidad incluso cuando se realizan cambios o mejoras. A continuación, se describen los principales tipos de pruebas relevantes para este trabajo.

4.5.1. Pruebas funcionales

Las pruebas funcionales tienen como objetivo verificar que el software realice correctamente las funciones para las cuales fue diseñado, de acuerdo con los requisitos establecidos en el análisis y la especificación del sistema. En este tipo de pruebas, se evalúa el comportamiento del sistema ante entradas específicas y se observan las salidas generadas para comprobar que coincidan con los resultados esperados.

Objetivos principales

- Validar que las funcionalidades cumplan con los requerimientos del usuario.
- Detectar errores en el comportamiento del sistema.
- Asegurar que cada componente interactúe correctamente con los demás.

4.5.2. Pruebas no funcionales

Las pruebas no funcionales se centran en evaluar características relacionadas con la calidad del software, más allá de sus funcionalidades específicas. Evalúan cómo se comporta el sistema más allá de sus funciones básicas, verifican si el sistema cumple con criterios de calidad como rendimiento, seguridad, usabilidad, escalabilidad, etc.

Objetivos clave

- Medir atributos de calidad del sistema.
- Identificar cuellos de botella, vulnerabilidades o problemas de experiencia de usuario.
- Asegurar que el sistema sea robusto, confiable y eficiente bajo distintas condiciones

4.5.3. Pruebas de regresión

Las pruebas de regresión consisten en ejecutar nuevamente los casos de prueba que ya habían sido validados anteriormente, con el fin de comprobar que los cambios recientes en el sistema no afecten negativamente las funcionalidades existentes. Esto es importante en entornos donde se realizan modificaciones continuas, como en metodologías ágiles.

- Aseguran que el sistema no se rompa con nuevas actualizaciones.
- Permiten validar que los flujos críticos siguen siendo estables.
- Son importantes para mantener la reproducibilidad y confianza en entornos auditables.

4.5.4. Pruebas manuales

Las pruebas manuales son aquellas en las que los casos de prueba son ejecutados por un tester humano, sin el uso de herramientas automáticas. El tester interactúa directamente con la aplicación, siguiendo un conjunto de pasos predefinidos para verificar el comportamiento esperado del sistema.

Características principales:

- **Flexibilidad:** Permite detectar errores visuales, de usabilidad y experiencia del usuario que podrían ser ignorados por pruebas automáticas.
- **Costo inicial bajo:** No requiere herramientas o scripts complejos para iniciar.
- **Dependencia del humano:** Requiere atención al detalle y tiempo, por lo que puede ser propenso a errores humanos.

Ventajas:

- Ideal para pruebas exploratorias, pruebas de interfaz gráfica y casos únicos o poco frecuentes.
- Útil en etapas tempranas del desarrollo donde los cambios son constantes.

Desventajas:

- Repetitivas y lentas.
- Escalabilidad limitada.
- Difíciles de mantener en proyectos grandes o de larga duración.

4.5.5. Pruebas automatizadas

Las pruebas automatizadas se realizan mediante el uso de scripts o herramientas que ejecutan casos de prueba sin la intervención humana directa. Estas pruebas se programan previamente y se ejecutan de forma repetitiva y controlada, siendo estos muy útiles para proyectos grandes o con necesidades frecuentes de regresión.

Características principales:

- Requiere inversión inicial para desarrollar y mantener scripts.
- Alta velocidad de ejecución, útil para pruebas repetitivas o regresiones frecuentes.
- Permite una alta cobertura y ejecución constante sin fatiga humana

Herramientas comunes:

- Jest, Mocha, JUnit (para pruebas unitarias)
- Selenium, Cypress, Playwright (para pruebas de interfaz)
- Postman/Newman, SoapUI (para pruebas de APIs)

Ventajas

- Ahorro de tiempo a largo plazo.
- Reutilizables y escalables.
- Resultados consistentes y trazables.
- Mejor cobertura de pruebas al automatizar múltiples escenarios.

Desventajas

- Menor capacidad para detectar errores
- Requieren conocimientos técnicos.
- Alta inversión en el desarrollo de los scripts.
- Mantenimiento constante ante cambios en el sistema

CAPITULO II

PLAN DE PRUEBAS

5. Plan de pruebas

5.1. Definición de plan de pruebas

El plan de pruebas constituye el documento técnico que guía la validación del sistema desarrollado. Su propósito es verificar que los módulos incluidos cumplen con los requisitos funcionales, de seguridad, rendimiento y usabilidad, definidos mediante una estrategia estructurada que permita reproducibilidad, trazabilidad y defensa académica.

5.2. Alcance del plan de pruebas

El alcance del plan de pruebas se definió a partir de un análisis realizado junto al usuario final, priorizando los módulos que forman parte de los procesos más importantes del sistema. La selección tomó en cuenta aspectos como: qué funcionalidades se utilizan con mayor frecuencia, cuáles tienen mayor impacto en las operaciones diarias y en qué áreas se habían identificado riesgos durante el uso del sistema. De esta forma, los módulos incluidos en la Tabla 2 representan flujos clave como la gestión de productos, órdenes, inventario, usuarios y autenticación. Validarlos garantiza que las tareas centrales del negocio se ejecuten de forma estable y confiable.

Los tipos de prueba definidos: funcionales, seguridad, rendimiento, integración, concurrencia y compatibilidad, se eligieron para cubrir las necesidades reales detectadas durante este análisis, asegurando que los módulos críticos respondan adecuadamente bajo diferentes condiciones.

| Módulo | Función | Uso |
|-----------|---|---|
| Products | Permite registrar, editar, buscar y gestionar productos en el sistema. | Usado por el administrador para mantener actualizado el catálogo. |
| Orders | Registra y gestiona órdenes de productos realizadas por los clientes. | Permite seguimiento, edición y generación de reportes de ventas. |
| Inventory | Administra almacenes, stock disponible y movimientos de inventario. | Control logístico del inventario y consulta de historial. |
| Users | Permite registrar y gestionar usuarios del sistema con distintos roles. | Solo accesible para administradores. |
| Customers | Gestiona el registro y edición de clientes asociados a órdenes de compra. | Base de datos de clientes para vinculación con ventas. |

Tabla 2. Módulos considerados en el plan de pruebas (Parte I).

| Módulo | Función | Uso |
|----------|---|--|
| Category | Permite crear y organizar categorías de productos. | Facilita la clasificación del catálogo de productos. |
| Purchase | Registra órdenes de compra para abastecimiento desde proveedores. | Mantiene el inventario mediante adquisiciones externas. |
| Login | Permite la autenticación de usuarios en el sistema. | Accesible para todos los roles; habilita funcionalidades personalizadas. |

Tabla 3. Módulos considerados en el plan de pruebas (Parte II).

5.3. Módulos no considerados

El plan de pruebas se diseñó en función de los módulos que el usuario identificó como esenciales para el funcionamiento diario del sistema. Esta selección permitió concentrar los esfuerzos en los procesos que sostienen las operaciones principales, optimizando tiempo y recursos en función del impacto real. Por lo tanto, algunos módulos complementarios fueron excluidos del alcance de esta fase. Su exclusión se fundamenta en tres criterios definidos conjuntamente con el usuario:

- **Uso operativo real** (frecuencia y relevancia en el día a día)
- **Impacto funcional** (contribución directa a los flujos críticos)
- **Riesgo asociado** (potencial de afectar la continuidad del sistema)

A continuación, se describen brevemente los módulos excluidos del plan de pruebas en esta etapa:

| Módulo | Función | Motivo |
|-----------------|---|---|
| Seller | Permite gestionar el listado de los vendedores registrados en el sistema. | No forma parte de los flujos críticos definidos para esta fase del plan de pruebas |
| States | El módulo tiene como propósito la administración de los Estados geográficos del país | Módulo auxiliar sin impacto directo en los flujos funcionales principales |
| Suppliers | Gestiona la información del listado de proveedores | No se encuentra vinculado directamente con los procesos de compra o inventario validados en esta etapa. |
| Coupons | Permite la creación de códigos para cupones de descuento. | No se consideró relevante para los flujos funcionales priorizados en el presente plan. |
| Purchase Orders | Muestra un registro de las compras de productos que se han hecho. Misma función del módulo de inventario. | Flujo dependiente de proveedores y almacenes no incluidos en el entorno de prueba actual. |

Tabla 4. Módulos no incluidos en el plan de pruebas.

5.4. Selección de pruebas automatizadas

Al definir los casos de prueba, es importante determinar cuáles son más adecuados para llevar a cabo la automatización y cuales para la ejecución manual. Esta clasificación se basa en criterios como la complejidad de la validación, la frecuencia de ejecución, la necesidad de validaciones visuales, entre otras.

- **Criterio para la automatización**

Se priorizó la automatización de casos de prueba críticos, repetitivos, complejos, técnicos y de seguridad, dejando manuales los que serán simples, rápidos o dependían de inspección visual o externa:

- 1. Flujos críticos y repetitivos**

- Aquellos casos de prueba que forman parte de los flujos centrales del sistema, para asegurar la funcionalidad principal del mismo.

- 2. Escenarios complejos**

- Aquellos casos que implican múltiples combinaciones de datos o escenarios que se llevarían un tiempo o propensos a errores si se hicieran manualmente.

- 3. Seguridad**

- Aquellos casos críticos de seguridad que necesitan garantizar la protección y cumplimiento de políticas de seguridad.

- 4. Validación de la integridad de datos y en el flujo de negocio**

- Aquellos casos que validan que la información en la interfaz coincida con la base de datos, así como procesos críticos de negocio

- 5. Pruebas técnicas sin dependencia a una revisión visual**

- Se automatizan pruebas que pueden evaluarse mediante herramientas o scripts, sin necesidad de inspección manual de UI o documentos.

5.5. Resumen visual

| Módulo | Total casos | Automatizables | No automatizables |
|--------------|-------------|----------------|-------------------|
| Login | 6 | 5 | 1 |
| Products | 10 | 7 | 3 |
| Orders | 11 | 8 | 3 |
| Inventory | 6 | 4 | 2 |
| Customer | 7 | 6 | 1 |
| Users | 6 | 6 | 0 |
| Category | 6 | 5 | 1 |
| Purchase | 5 | 5 | 0 |
| Seguridad | 15 | 14 | 1 |
| Rendimiento | 2 | 2 | 0 |
| Concurrencia | 1 | 0 | 1 |
| UI/UX | 2 | 2 | 0 |
| Total | 77 | 64 | 13 |

Tabla 5. Número de casos a automatizables/no automatizables.

5.6. Criterios para la selección de herramientas

En el proceso de aseguramiento de la calidad del software, las herramientas de testing juegan un papel fundamental para garantizar que un sistema cumpla con los requisitos funcionales y no funcionales establecidos. La correcta selección de herramientas depende de factores como el tipo de proyecto, el lenguaje de programación utilizado, los recursos disponibles y el nivel de madurez del equipo.

Para la ejecución de las pruebas, se ha seleccionado un conjunto de herramientas para la automatización, seguridad y rendimiento que ofrecen una cobertura completa, desde la funcionalidad y la interfaz hasta el rendimiento y la seguridad del sistema:

- 1. Compatibilidad con el ecosistema de Laravel:** Se priorizaron herramientas que se integren de manera nativa o fluida con el framework del sistema, con el objetivo de minimizar la curva de aprendizaje, facilitar su configuración y asegurar el correcto funcionamiento entre las herramientas y el código.
- 2. Cobertura del tipo de prueba:** Se buscó un conjunto de herramientas que ofreciera una cobertura en las áreas de pruebas necesarias, abarcando diferentes tipos de pruebas: funcionales, de rendimiento, de seguridad, y de interfaz de usuario.
- 3. Costo y permiso de uso:** Se evaluó el licenciamiento de cada herramienta. Se considero aquellas opciones de código abierto y gratuitas que permiten mantener los costos del proyecto.

4. **Comunidad y soporte:** La existencia de una comunidad activa, documentación y soporte técnico (en caso de ser necesario) fue un factor importante para garantizar que se puedan resolver problemas y que las herramientas se mantengan actualizadas.

La evaluación de las herramientas se realizó en función de los criterios de selección, resultando en la siguiente elección final:

1. **PHPUnit y Laravel Dusk:** Seleccionados por su integración nativa con Laravel. Siendo herramientas que garantizan la compatibilidad y flujo de trabajo optimizado para pruebas unitarias y E2E. Son de código abierto.
2. **JMeter:** Tiene la capacidad para simular una alta carga de usuarios y su uso extendido para pruebas de rendimiento. Es de código abierto.
3. **OWASP ZAP y Qualys SSL Labs:** Seleccionadas por ser altamente efectivas en sus respectivos campos. OWASP es de código abierto y Qualys SSL gratuito.
4. **SonarQube:** Seleccionada por su eficiencia en el análisis estático del código, permitiendo identificar errores y vulnerabilidades.

5.7. Estándar IEEE-829

El plan de pruebas se fundamenta en el estándar IEEE-829, también conocido como Standard for Software Test Documentation, el cual define un marco formal para la planificación, ejecución y registro de las actividades de prueba en proyectos de software. Este estándar fue desarrollado por el Institute of Electrical and Electronics Engineers (IEEE) con el propósito de establecer una estructura uniforme que garantice la trazabilidad, la consistencia y la calidad de los procesos de testing.

La principal contribución del IEEE-829 radica en que proporciona un conjunto de documentos y lineamientos que permiten organizar de manera sistemática la información generada durante el ciclo de vida de las pruebas.

A través de este estándar, se promueve la transparencia, la reproducibilidad y la posibilidad de auditar los resultados, elementos esenciales tanto en entornos empresariales como académicos. El IEEE-829 define ocho documentos principales que estructuran el proceso de pruebas:

1. **Plan de pruebas:** especifica el alcance, los objetivos, los recursos, el calendario, los riesgos y las responsabilidades del equipo de pruebas.
2. **Diseño de casos de prueba:** describe las condiciones, datos y criterios que se usarán para evaluar cada funcionalidad del sistema.
3. **Especificación de casos de prueba:** detalla los pasos de ejecución, entradas, resultados esperados y criterios de aceptación de cada caso.
4. **Procedimiento de prueba:** establece la secuencia de acciones que deben seguirse para ejecutar los casos de prueba.
5. **Registro de resultados:** documenta la evidencia de cada ejecución, incluyendo fecha, entorno, incidentes y observaciones.
6. **Informe de incidentes:** recoge los defectos o desviaciones detectadas durante las pruebas, junto con su severidad e impacto.
7. **Resumen de resultados:** presenta una síntesis global del proceso de pruebas, resultados obtenidos, métricas y conclusiones.
8. **Análisis de variaciones o desviaciones:** documenta cambios relevantes entre versiones o elementos sometidos a prueba.

Estos componentes no solo garantizan una documentación completa, sino que también aseguran la trazabilidad entre los requerimientos del sistema, los casos de prueba y los resultados obtenidos. Al seguir este modelo, cada actividad puede rastrearse desde su planificación hasta su cierre, facilitando la evaluación de la cobertura y la detección de áreas susceptibles de mejora.

En el contexto del presente proyecto, la adopción del estándar IEEE-829 permite estructurar el plan de pruebas de manera ordenada y auditable, integrando tanto las pruebas manuales como las automatizadas bajo un mismo marco metodológico. Por lo que, se establecieron relaciones directas entre los requerimientos funcionales, los módulos del sistema, los tipos de prueba aplicados y las evidencias obtenidas.

Además, la aplicación de este estándar contribuye a la creación de un documento actualizado, en el que se almacenan los reportes y resultados de cada prueba. Esta práctica permite mantener la coherencia entre las fases del proyecto, asegura la calidad de los registros y facilita el análisis comparativo entre pruebas.

En resumen, el estándar IEEE-829 no solo proporciona una guía formal para la documentación de las pruebas, sino que también constituye una herramienta esencial para la mejora continua, la gestión de riesgos y la toma de decisiones informadas dentro del proceso de aseguramiento de calidad. La aplicación del estándar IEEE-829 no solo aporta estructura documental, sino que también fortalece la trazabilidad entre pruebas, resultados y decisiones técnicas, consolidando así la calidad del proceso de validación.

5.8. Riesgos

Durante la planificación y ejecución del plan de pruebas pueden surgir factores que comprometan la calidad del sistema, la estabilidad del entorno o la trazabilidad de los resultados. Por ello, resulta fundamental identificar oportunamente los riesgos que puedan afectar el cumplimiento de los objetivos de testing.

Estos riesgos pueden clasificarse en categorías técnicas, operativas, de seguridad, de planificación y de requerimientos. Su impacto puede reflejarse en la ejecución del plan, la confiabilidad de los resultados o la consistencia de los entregables.

A continuación, se presentan algunos de los riesgos potenciales que podrían surgir en el transcurso del proceso de testing:

| Tipo de riesgo | Descripción | Impacto | Probabilidad |
|----------------|--|---------|--------------|
| Técnico | Inestabilidad del entorno de pruebas debido a configuraciones incompletas o incompatibles. | Alto | Media |
| Seguridad | Presencia de vulnerabilidades críticas no corregidas que impidan ejecutar pruebas sin comprometer datos. | Alto | Media |
| Planificación | Retrasos en la entrega de versiones necesarios para iniciar las pruebas. | Medio | Alta |
| Requerimientos | Cambios de último momento en funcionalidades clave que invaliden los casos de prueba definidos. | Medio | Media |
| Rendimiento | Falta de disponibilidad de tiempo o herramientas necesarias para ejecutar pruebas. | Alto | Baja |

Tabla 6. Riesgos potenciales durante la planificación y ejecución del testing

Los riesgos ejemplificados en esta sección corresponden a factores que podrían comprometer la ejecución del plan de pruebas, afectando la estabilidad del entorno, la disponibilidad de recursos o la consistencia de los entregables. Las estrategias de mitigación, seguimiento y contingencia asociadas a estos riesgos se desarrollan en el apartado “Plan de riesgos y contingencias”, donde se especifica cada escenario identificado.

De manera general, una adecuada gestión de riesgos permite fortalecer la fiabilidad del proceso de pruebas, reducir imprevistos y garantizar que el sistema cumpla con los estándares de calidad establecidos.

5.9. Características que se probarán

Las características que se evaluarán dentro del presente plan de pruebas corresponden a los módulos más relevantes del sistema de comercio electrónico. La selección se realizó con base en su impacto en el negocio, y el nivel del riesgo asociado a su funcionamiento impacto en el negocio, la frecuencia de uso esperada por el usuario final, y el nivel de riesgo asociado a su funcionamiento.

A continuación, se resumen las principales características y el tipo de prueba aplicable a cada una:

| Módulo | Tipo de prueba | Objetivo principal |
|-----------|---------------------------------------|---|
| Login | Funcional y seguridad | Validar el proceso de autenticación y control de acceso de usuarios. |
| Products | Funcional, UI y rendimiento | Verificar la correcta gestión del catálogo de productos y su presentación. |
| Orders | Funcional, integración y regresión | Comprobar el flujo de órdenes de inicio a fin. |
| Inventory | Funcional, concurrencia y rendimiento | Validar el control de stock y la sincronización de movimientos de inventario. |
| Customers | Funcional y datos | Comprobar el registro y mantenimiento de información de clientes. |
| Users | Seguridad y permisos | Verificar los niveles de acceso y la correcta asignación de roles. |
| Category | Funcional | Validar la organización de productos por categorías. |
| Purchase | Integración y datos | Evaluar la gestión de compras. |

Tabla 7. Características a probar.

Estas características representan los flujos de negocio más críticos del sistema, por lo que su validación integral contribuye a garantizar la estabilidad operativa, la confiabilidad de los datos y la seguridad del entorno en producción.

5.10. Alcance (estrategia)

El presente proyecto se centra en el diseño e implementación progresiva de un plan integral de pruebas de calidad que combine métodos manuales y automatizados, aplicado específicamente a los módulos críticos del sistema de comercio electrónico en producción. La estrategia se desarrolló bajo la metodología Scrum ligero, estructurada en tres sprints de trabajo consecutivos, orientados a garantizar la trazabilidad, la eficiencia en la ejecución y la mejora continua de los procesos de verificación.

Durante la primera etapa, se realizó el levantamiento de requerimientos de pruebas y la identificación de flujos de negocio prioritarios, con el propósito de concentrar los esfuerzos en las funcionalidades de mayor valor operativo. Con base en ello, se diseñó una suite manual de casos de prueba que permitió validar las principales operaciones del sistema, incluyendo la gestión de productos, órdenes, inventario, clientes y usuarios.

En una segunda fase, la estrategia evolucionó hacia la automatización de pruebas, con el objetivo de optimizar los tiempos de ejecución, ampliar la cobertura y facilitar la detección temprana de defectos. Los resultados de cada ejecución fueron documentados en una plantilla estándar y utilizando un control de seguimiento en Excel.

La estrategia también contempló la implementación de métricas de calidad para evaluar de manera objetiva aspectos como la adecuación funcional, el rendimiento, la seguridad, la usabilidad y la mantenibilidad del sistema. Estas acciones permiten monitorear el comportamiento del sistema durante las pruebas y facilitar el análisis comparativo entre versiones o entregables.

Asimismo, se desarrolló un repositorio central de documentación viva, donde se registraron los resultados, incidencias y hallazgos de cada iteración, con el fin de garantizar la trazabilidad y facilitar la mejora continua. Esta práctica se complementó con la ejecución de pruebas de carga, seguridad y regresión, asegurando una visión integral del desempeño del sistema en condiciones reales.

5.11. Criterios de aprobación/fallo/ejecución con hallazgos

Para garantizar la objetividad en la evaluación del sistema y asegurar que las pruebas cumplan con los estándares de calidad establecidos, se definieron criterios específicos de aprobación y fallo, aplicables a todos los módulos críticos evaluados. Estos criterios permiten determinar de forma medible cuándo un caso de prueba, un conjunto de pruebas o una iteración completa puede considerarse exitosa.

- Un caso de prueba aprobado cuando los resultados obtenidos durante la ejecución coinciden con los resultados esperados definidos en la documentación técnica. Cada funcionalidad validada deberá cumplir con las condiciones establecidas en los requerimientos funcionales y no funcionales, sin generar errores visibles, interrupciones, comportamientos inesperados o pérdida de datos.
- Un caso de prueba se considerará fallido cuando el sistema presente desviaciones significativas respecto al comportamiento esperado. Esto incluye errores funcionales, fallas en la validación de datos, tiempos de respuesta que superen los

límites establecidos, inconsistencias en la interfaz de usuario, fallos de seguridad o cualquier defecto que impida la correcta ejecución del flujo de negocio.

- Un caso de prueba ejecutado con hallazgos cuando, aun cumpliendo con los resultados esperados, se identifican situaciones que no constituyen un fallo como tal, pero sí requieren atención y documentación.

5.12. Entregables de prueba

Los entregables de prueba constituyen el conjunto de documentos, artefactos y evidencias generadas durante la planificación, ejecución y cierre del proceso de aseguramiento de calidad. Su propósito es garantizar la trazabilidad del trabajo realizado, facilitar la interpretación de resultados y servir como soporte técnico ante revisiones o auditorías posteriores.

En el marco de este proyecto, los entregables de prueba se clasifican en tres etapas principales: planificación, ejecución y documentación de resultados.

Durante la etapa de planificación, los entregables comprenden la definición del Plan de Pruebas, el cual establece el alcance, objetivos, metodología, criterios de aceptación, riesgos y responsabilidades del equipo.

También se incluyen los documentos de levantamiento de requerimientos funcionales y no funcionales, las matrices de riesgo \times impacto, los casos de uso y la matriz de trazabilidad que vincula cada requerimiento con sus respectivas pruebas. Estos artefactos permiten garantizar que la cobertura de pruebas sea completa y alineada con los objetivos del sistema.

En la etapa de ejecución, los entregables corresponden a los recursos generados durante la aplicación práctica de las pruebas manuales y automatizadas. Entre ellos se incluyen las suites de pruebas manuales documentadas en hojas de ejecución, los scripts automatizados desarrollados con herramientas seleccionadas, los reportes automáticos de resultados, los registros de defectos detectados, y las evidencias gráficas (capturas de pantalla, logs y archivos de salida) que confirman el comportamiento observado del sistema. Además, se entrega la configuración del ambiente de pruebas, donde se detalla la infraestructura, los entornos de ejecución y las versiones de las herramientas utilizadas.

Finalmente, en la etapa de cierre, los entregables se orientan a la consolidación y análisis de la información recolectada. Esta fase incluye el Informe Final de Pruebas, que presenta el resumen de los resultados obtenidos, la clasificación de defectos por severidad, la evaluación de métricas de rendimiento (tiempos de respuesta, tasa de error, cobertura alcanzada) y las recomendaciones técnicas para la mejora continua del sistema.

En conjunto, estos entregables constituyen la evidencia formal del proceso de aseguramiento de calidad implementado. Además, permiten demostrar que las pruebas fueron planificadas, ejecutadas y documentadas de manera sistemática, contribuyendo así a la mejora de la confiabilidad, mantenibilidad y rendimiento del sistema de comercio electrónico evaluado.

5.13. Preparación técnica y metodológica

La ejecución del presente plan de pruebas requirió una preparación previa tanto en aspectos técnicos como metodológicos, con el objetivo de garantizar la rigurosidad y trazabilidad exigidas. El equipo reforzó sus conocimientos en el uso del entorno de desarrollo, la comprensión de la arquitectura del sistema y la familiarización con las herramientas empleadas para la gestión, ejecución de pruebas y revisión de conceptos relacionados con los tipos de pruebas, diseño de casos de prueba, gestión de incidencias y validación de resultados.

- **Preparación técnica**

Se estudiaron y aplicaron herramientas especializadas para la automatización de pruebas, la validación funcional y la documentación estructurada de resultados. Se familiarizó con los procedimientos para automatizar casos de prueba y ejecutar pruebas manuales de manera sistemática, asegurando la repetibilidad de los resultados. Las pruebas fueron estructuradas con pasos claros, datos de entrada precisos y resultados esperados documentados. Se priorizaron los casos críticos y aquellos que impactan funcionalidades principales, y se registraron todas las ejecuciones para garantizar trazabilidad y control sobre los resultados obtenidos.

Adicionalmente, se establecieron los ambientes de prueba controlados y se configuraron datos representativos de escenarios reales, permitiendo evaluar correctamente la funcionalidad, la gestión de errores y el comportamiento del sistema bajo diferentes condiciones de uso.

- **Preparación metodológica**

Desde el enfoque metodológico, se establecieron lineamientos para el diseño, ejecución y documentación de pruebas:

- **Diseño de casos de prueba:** Se revisaron criterios para estructurar cada caso con precondiciones, pasos de ejecución, datos de entrada, resultados esperados y criterios de aceptación, asegurando la trazabilidad con los requerimientos del sistema.

- **Documentación técnica estructurada:** Se definieron instrucciones para registrar resultados, incidencias y evidencias de manera clara y organizada, facilitando la replicación del proceso y la revisión académica.
- **Delimitación del alcance:** Se definieron los módulos incluidos y excluidos, así como los casos automatizados, no automatizados y parcialmente cubiertos, con justificación técnica para cada decisión.

5.14. Ambientes y recursos de prueba

El proceso de pruebas requiere ambientes claramente definidos que permitan validar el correcto funcionamiento del sistema en diferentes etapas del desarrollo. Cada ambiente posee características específicas que contribuyen a la identificación oportuna de incidencias, la verificación de la integración entre módulos y la reproducción de escenarios.

5.14.1. Recurso de software/hardware

En relación con el ambiente y las herramientas de prueba, es importante considerar los requisitos técnicos que garanticen el adecuado desempeño durante la ejecución del proceso de testing para asegurar que las pruebas se realicen en condiciones estables y compatibles con el entorno de desarrollo. A continuación, se detallan las condiciones y configuraciones para el hardware, software y servicios necesarios en las pruebas.

| Recurso | PC 1 | PC 2 | PC 3 |
|-------------------|--------------------------|--------------------------|--------------------------|
| Tipo | Laptop | Laptop | Desktop |
| Procesador | 12th Gen Intel Core i3 | Intel Core i7-855U | Intel i5-10400f |
| RAM | 8 GB | 12 GB | 32 GB |
| SO | Windows 11 | Windows 11 | Windows 10 |
| Chrome | Versión 141.0.7390.55 | Versión 140.0.7339.20 | Versión 140.0.7339.20 |
| Edge | Versión 141.0.3537.57 | Versión 140.0.7339.20 | Versión 140.0.7339.20 |
| Firefox | Versión 143.04 | Versión 141.0.3537.57 | Versión 141.0.3537.57 |

Tabla 8. Recurso de software y hardware a utilizar durante el proceso de pruebas.

5.14.2. Entorno técnico

A continuación, se detalla el entorno técnico del sistema, incluyendo las tecnologías, versiones y funciones críticas de cada componente. Esta información permite entender la base tecnológica que soporta el desarrollo y operación del proyecto.

| Componente | Tecnología | Versión |
|---------------|------------------|---------|
| Back-end | PHP (Laravel) | 8.2.28 |
| Base de datos | MySQL | 8.0 |
| Front-end | Blade, JS y HTML | 8.2.12 |
| Paquetes | Composer | 2.8.9 |
| Proveedor | Hostinger | - |
| Servidor | LiteSpeed | - |

Tabla 9. Resumen de componentes tecnológicos del sistema.

5.14.3. Instrumentos técnicos del proceso de testing

Durante el proceso de validación del sistema, se emplearon diversas herramientas especializadas para cubrir pruebas unitarias, funcionales, de rendimiento, seguridad y automatización, lo cual permiten realizar evaluaciones, detectar defectos de manera temprana y garantizar que los diferentes componentes del sistema cumplieran con los estándares de calidad establecidos. A continuación, se resume las tecnologías utilizadas, indicando su propósito y versión correspondiente.

| Herramienta | Tipo de pruebas | Versión/Configuración |
|-----------------------------|-------------------------------------|--|
| PHPUnit | Unitarias | Versión 10.4.1. PHPUnit instalación por defecto. |
| Laravel Dusk | E2E y UI | Versión 8.3.3. Instalada vía Composer. |
| JMeter | Carga y Rendimiento | Versión 5.6.3. Instalado desde la máquina física. |
| Selenium WebDriver | Automatización Multi-Navegador | Versión 4.35.0 Configurado manual. |
| OWASP ZAP | Seguridad | Versión 2.16.1. Instalado desde la máquina física. |
| SonarQube | Análisis de código | Versión 25.3.0.1 Configuración manual. |
| Qualys SSL Labs | Seguridad SSL/TLS | No requiere instalación local |
| Excel (OneDrive/Sharepoint) | Diseño y control de casos de prueba | Recurso de apoyo. Almacenamiento y gestión colaborativa para evidencia y la documentación complementaria |

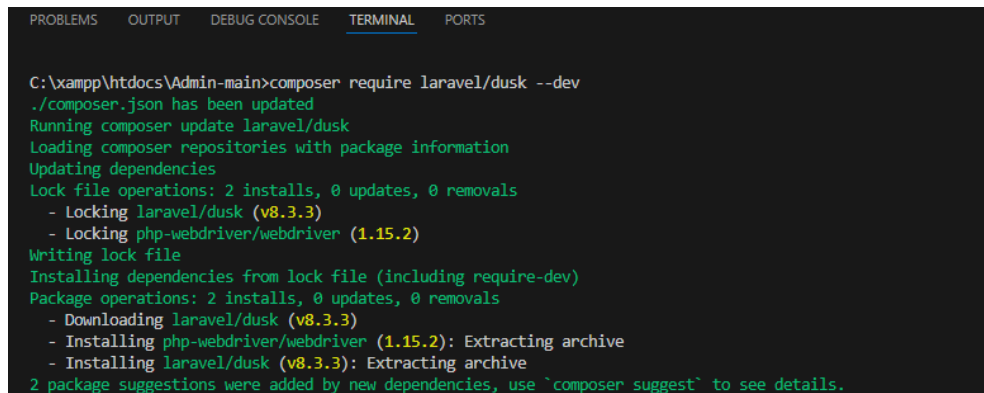
Tabla 10. Herramientas de pruebas a utilizar y sus versiones.

5.14.4. Configuración de herramientas de pruebas

Con el objetivo de ejecutar pruebas bajo condiciones controladas, se detallan los procedimientos de instalación y configuración de las herramientas seleccionadas. Se describe cómo se integraron técnicamente en el proyecto, incluyendo rutas de instalación, ajustes por entorno y consideraciones necesarias para su funcionamiento correcto.

- **Laravel Dusk**

1. **Agregando dependencia:** Se incorpora Laravel Dusk al proyecto mediante Composer como dependencia de desarrollo: *composer require laravel/dusk --dev*



```
C:\xampp\htdocs\Admin-main>composer require laravel/dusk --dev
./composer.json has been updated
Running composer update laravel/dusk
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking laravel/dusk (v8.3.3)
  - Locking php-webdriver/webdriver (1.15.2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading laravel/dusk (v8.3.3)
  - Installing php-webdriver/webdriver (1.15.2): Extracting archive
  - Installing laravel/dusk (v8.3.3): Extracting archive
2 package suggestions were added by new dependencies, use `composer suggest` to see details.
```

Ilustración 1. Instalación de Laravel Dusk como dependencia de desarrollo.

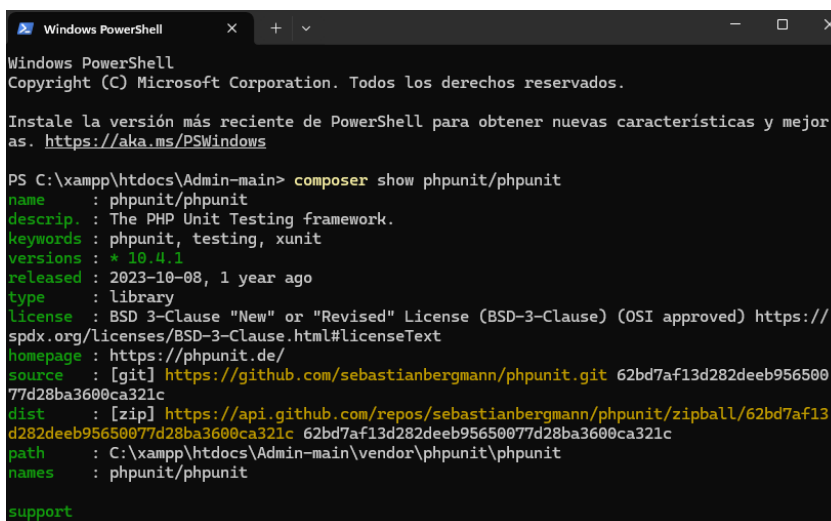
2. **Inicializar Dusk:** Se ejecuta el comando Artisan para instalar Dusk y generar la estructura de pruebas de navegador: *php artisan dusk:install*.
3. **Directorios creados:** Esto crea el directorio tests/Browser, junto con archivos base como ExampleTest.php y la clase DuskTestCase.php

- **PHPUnit**

1. **Verificación de instalación por defecto:** Laravel incluye PHPUnit como dependencia de desarrollo. Para confirmar su presencia, se ejecuta *composer show phpunit/phpunit*
2. **Instalación manual (si no está presente):** En caso de que no esté incluido, se puede instalar con: *composer require --dev phpunit/phpunit*

3. Archivos para pruebas: Laravel incorpora dos elementos fundamentales para la ejecución de pruebas:

- *phpunit.xml*: ubicado en la raíz del proyecto, define la configuración del entorno de pruebas.
- *tests/*: carpeta que contiene los casos de prueba, organizados en Feature y Unit.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\xampp\htdocs\Admin-main> composer show phpunit/phpunit
name      : phpunit/phpunit
descrip.  : The PHP Unit Testing framework.
keywords  : phpunit, testing, xunit
versions  : * 10.4.1
released  : 2023-10-08, 1 year ago
type      : library
license   : BSD 3-Clause "New" or "Revised" License (BSD-3-Clause) (OSI approved) https://
spx.org/licenses/BSD-3-Clause.html#licenseText
homepage  : https://phpunit.de/
source    : [git] https://github.com/sebastianbergmann/phpunit.git 62bd7af13d282deeb956500
77d28ba3600ca321c
dist      : [zip] https://api.github.com/repos/sebastianbergmann/phpunit/zipball/62bd7af13
d282deeb95650077d28ba3600ca321c 62bd7af13d282deeb95650077d28ba3600ca321c
path      : C:\xampp\htdocs\Admin-main\vendor\phpunit\phpunit
names     : phpunit/phpunit
support
```

Ilustración 2. Verificación de la versión de PHPUnit instalada.

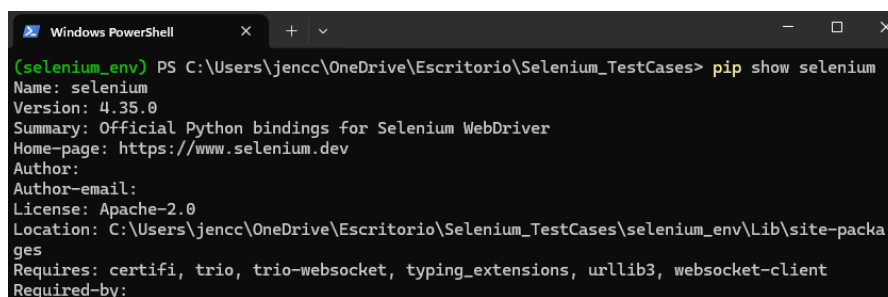
4. Configuración de entorno de pruebas: Desde el archivo phpunit.xml se define la conexión a la base de datos, el mailer y otras variables de entorno necesarias.

• Selenium WebDriver

1. Instalar el controlador: Se descarga los drivers compatibles con el sistema operativo a utilizar. En este caso, se empleó el driver correspondiente para Chrome.

2. Instalar Selenium como librería: Para la instalación de Selenium con Python se requiere ejecutar lo siguiente:

pip install selenium



```
Windows PowerShell
(selenium_env) PS C:\Users\jenc\OneDrive\Escritorio\Selenium_TestCases> pip show selenium
Name: selenium
Version: 4.35.0
Summary: Official Python bindings for Selenium WebDriver
Home-page: https://www.selenium.dev
Author:
Author-email:
License: Apache-2.0
Location: C:\Users\jenc\OneDrive\Escritorio\Selenium_TestCases\selenium_env\Lib\site-packa
ges
Requires: certifi, trio, trio-websocket, typing_extensions, urllib3, websocket-client
Required-by:
```

Ilustración 3. Verificación de la versión instalada de Selenium en entorno Python.

- **Owasp ZAP**

1. **Descarga e instalación** Se descarga el instalador ejecutable (.exe) desde el sitio oficial de OWASP ZAP y se procede con la instalación en la máquina física.

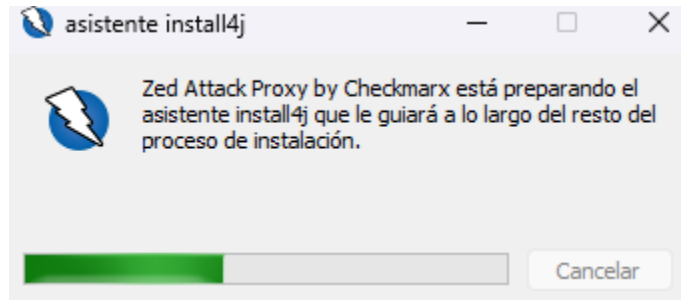


Ilustración 4. Inicio del asistente de instalación de OWASP ZAP.

2. **Inicio de sesión:** Al iniciar la herramienta por primera vez, se solicita al usuario si desea guardar la sesión de análisis.
3. **Configuración del proxy del navegador** Para interceptar y analizar el tráfico HTTP/HTTPS, es necesario configurar el proxy del navegador:
 - **Chrome y Microsoft Edge:** Se realiza desde la configuración de red de Windows en la sección de Sistema y Rendimiento
 - **Firefox:** Se configura directamente desde el navegador en la sección de Configuración de red.

- **Jmeter**

1. **Descarga del software:** Se descarga la versión estable de Apache JMeter desde el sitio oficial.
2. **Descompresión y ubicación:** Se descomprime el archivo en una ruta accesible.
3. **Requisitos previos:** JMeter requiere tener instalado JDK
4. **Ejecución de JMeter:** Desde la carpeta descomprimida, se ejecuta el archivo:
 - **En Windows:** bin\jmeter.bat

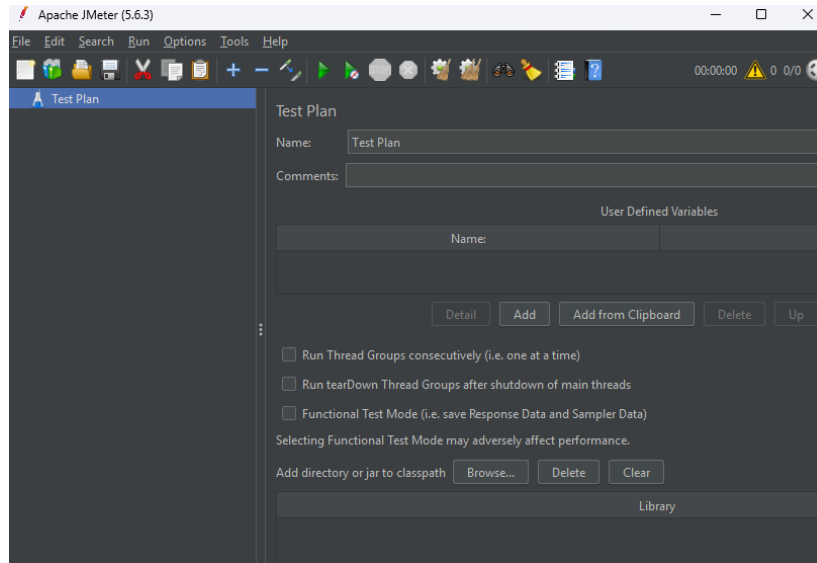


Ilustración 5. Interfaz principal de Apache JMeter.

- **SonarQube**

1. **Instalación y configuración:** Se despliega SonarQube en servidor local o nube.
2. **Integración con proyecto:** Se añade archivo sonar-project.properties en el repositorio.
3. **Ejecución de análisis:** Se corre sonar-scanner para enviar resultados a SonarQube.
4. **Visualización y seguimiento:** Los hallazgos se revisan en el dashboard de SonarQube.

```

C:\WINDOWS\system32\cmd. x + v
Starting SonarQube...
2025.11.18 22:02:16 INFO app[][o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\jenc\OneDrive\Escritorio\Sonar\sonarqube-25.3.0.104237\sonarqube-25.3.0.104237\temp
2025.11.18 22:02:16 INFO app[][o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:{}]
2025.11.18 22:02:16 INFO app[][o.s.a.ProcessLauncherImpl] Launch process[ELASTICSEARCH] from [C:\Users\jenc\OneDrive\Escritorio\Sonar\sonarqube-25.3.0.104237\sonarqube-25.3.0.104237\elasticsearch]
C:\Program Files\Java\jdk-17\bin\java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=./bin/elasticsearch -Dcli.libs=lib/tools/server-cli -Des.path.home=C:\Users\jenc\OneDrive\Escritorio\Sonar\sonarqube-25.3.0.104237\sonarqube-25.3.0.104237\elasticsearch -Des.path.conf=C:\Users\jenc\OneDrive\Escritorio\Sonar\sonarqube-25.3.0.104237\sonarqube-25.3.0.104237\temp\conf\es -Des.distribution.type=tar -cp C:\Users\jenc\OneDrive\Escritorio\Sonar\sonarqube-25.3.0.104237\sonarqube-25.3.0.104237\elasticsearch\lib\*;C:\Users\jenc\OneDrive\Escritorio\Sonar\sonarqube-25.3.0.104237\sonarqube-25.3.0.104237\elasticsearch\lib\cli-launcher\* org.elasticsearch.launcher.CliToolLauncher
2025.11.18 22:02:16 INFO app[][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in order to avoid potential conflicts

```

Ilustración 6. Proceso de arranque de SonarQube

- **Qualys SSL Labs**

- 1. Preparar el servidor y certificado**

- Instalar un certificado SSL/TLS válido emitido por una autoridad de confianza.
- Asegurar de incluir la cadena completa de certificados intermedios.
- Configurar claves.

- 2. Configurar protocolos seguros**

- 3. Configurar cifrados recomendados**

- 4. Verificar la configuración con SSL Labs**

- Accede a Qualys SSL Labs.
- Ingresa el dominio del servidor

Qualys SSL Labs

Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > cuscatec.cuscatec.com

SSL Report: cuscatec.cuscatec.com

Assessed on: Thu, 23 Oct 2025 12:27:52 UTC | [Hide](#) | [Clear cache](#)

[Scan Another >>](#)

| | Server | Test time | Grade |
|---|---|---|-------|
| 1 | 212.85.28.193 Ready | Thu, 23 Oct 2025 12:24:32 UTC Duration: 97.883 sec | A |
| 2 | 2a02:4780:1:549:0:77c:3469:5 Ready | Thu, 23 Oct 2025 12:26:10 UTC Duration: 102.64 sec | A |

SSL Report v2.4.1

Copyright © 2009-2025 [Qualys, Inc.](#) All Rights Reserved. [Privacy Policy](#)

[Try Qualys for free!](#) Experience the award-winning [Qualys Cloud Platform](#) and the entire collection of [Qualys Cloud Apps](#), including [certificate security](#) solutions. [Terms and Conditions](#)

Ilustración 7. Configuración para análisis con Qualys SSL Labs

5.15. Responsabilidades

La correcta ejecución del plan de pruebas requiere una distribución clara de funciones que garantice la coordinación entre las etapas de planificación, diseño, ejecución y documentación. Para ello, se definieron los roles y responsabilidades del equipo de trabajo conforme al marco de la metodología ágil Scrum, adaptada al contexto académico y operativo del proyecto. El equipo está conformado por tres integrantes, quienes asumieron funciones técnicas y de gestión de manera colaborativa, manteniendo una comunicación continua y una documentación viva del proceso.

A continuación, se describen los roles principales y sus responsabilidades dentro del proyecto:

- **Product Owner (Propietario del producto)**

Responsable de definir los objetivos del proyecto y priorizar los flujos de negocio a validar. Supervisa el cumplimiento del plan de trabajo, aprueba los entregables de cada sprint y garantiza que las pruebas estén alineadas con los requerimientos funcionales del sistema de comercio electrónico. Además, participa en la validación de resultados y en la aprobación de las correcciones antes de su implementación.

- **Scrum Master (Facilitador del proceso)**

Encargado de asegurar que la metodología Scrum se aplique correctamente durante los cuatro sprints de ejecución. Su función principal es coordinar las reuniones de planificación, revisión y retrospectiva, mantener la trazabilidad del progreso y gestionar los riesgos o bloqueos que puedan afectar el cumplimiento de los objetivos. También vela por la integración de las herramientas de automatización y por el control de calidad del flujo de trabajo.

- **Equipo de QA (Quality Assurance Team)**

Desempeña la función técnica de diseñar, ejecutar y documentar las pruebas manuales y automatizadas. Entre sus responsabilidades se incluyen la elaboración de la matriz de pruebas, la creación de casos de prueba funcionales, la configuración del ambiente de pruebas, la implementación de scripts automatizados, la ejecución de pruebas de carga y rendimiento, y el registro de defectos y métricas en los reportes correspondientes. Además, el equipo QA analiza los resultados, genera dashboards de indicadores y formula recomendaciones para la mejora continua del sistema.

- **Desarrollador del Sistema**

Aunque el sistema analizado pertenece a un desarrollador independiente, este rol se considera dentro del alcance del proyecto como colaborador técnico externo. Su responsabilidad consiste en atender los reportes de defectos emitidos por el equipo QA, corregir las incidencias críticas y garantizar que las nuevas versiones del sistema sean entregadas para su validación.

- **Asesor Académico**

El asesor del proyecto cumple una función supervisora y de acompañamiento metodológico. Se encarga de revisar y aprobar el cumplimiento de los lineamientos académicos, validar la coherencia de los resultados y orientar la integración de los estándares de calidad y la documentación técnica.

5.16. Calendario

El calendario del plan de pruebas se organizó conforme a la metodología Scrum ligero, dividiendo el trabajo en Sprint semanales. Esta planificación permite estructurar las actividades de manera ordenada, asignando tiempos para la elaboración teórica, la ejecución de pruebas y la documentación de resultados. Cada sprint se diseñó considerando la complejidad de las tareas y la disponibilidad de recursos, con el objetivo de mantener un flujo constante de trabajo y asegurar la entrega oportuna de cada etapa del proyecto.

| Sprint | Actividades principales | Duración estimada |
|---|--|--|
| Sprint 1 - Capítulo I: Generalidades | Elaboración del marco teórico: fundamentos de la calidad del software, conceptos de pruebas, niveles y tipos de testing. | 1.5 semanas (23/09/2025 – 01/10/2025) |
| Sprint 2 - Capítulo II: Plan de Pruebas | Desarrollo de la estructura técnica del Plan de Pruebas, integrando los apartados teóricos, metodológicos y operativos necesarios para su ejecución. | 1.5 semanas (01/10/2025 – 09/10/2025) |
| Sprint 3 - Capítulo III: Ejecución y documentación técnica | Implementación de pruebas manuales y automatizadas, recolección de evidencias y elaboración del informe final. | 2 semanas (09/10/2025 – 24/10/2025) |
| Sprint 4 - Capítulo III: Análisis y cierre del proyecto | Consolidación de resultados, análisis de métricas, revisión académica y presentación de conclusiones. | 1.5 semanas (24/10/2025 – 03/11/2025) |

Tabla 11. Calendario para el plan de pruebas.



 **Ecommerce**

PLAN DE RIESGOS Y CONTINGENCIAS

DISEÑO E IMPLEMENTACIÓN DE UN PLAN INTEGRAL DE PRUEBAS MANUALES Y AUTOMATIZADAS PARA ASEGURAR LA CALIDAD DE UN SISTEMA DE COMERCIO ELECTRÓNICO EN PRODUCCIÓN

VERSION 1.0

| Fecha | Autor | Contenido |
|--------------|---|---|
| 7/10/2025 | Jennifer Tatiana Cruz Castro Nathaly Adriana Rodríguez Martínez Mauricio Daniel Hernández Jovel | Introducción Alcance Objetivos Equipo de trabajo y roles Criterios de evaluación de riesgos Plan de comunicación Identificación y evaluación de riesgos |

5.17. Plan de riesgos y contingencias

5.17.1. Introducción

El presente plan tiene como propósito identificar, evaluar y gestionar los riesgos que puedan surgir durante la ejecución de los casos de pruebas del sistema E-commerce. Su propósito es garantizar la continuidad, calidad y confiabilidad del proceso de ejecución de los casos definidos, incluso ante eventos imprevistos que puedan afectar el desarrollo del plan de pruebas.

5.17.2. Alcance

- El plan aplica a las pruebas realizadas sobre la versión final del sistema web, abarcando los siguientes tipos de pruebas:
 - Pruebas funcionales
 - Pruebas de regresión
 - Pruebas de compatibilidad y UI/UX
 - Pruebas de concurrencia
 - Pruebas de rendimiento
 - Pruebas de seguridad
 - Pruebas de usabilidad
 - Pruebas de integración
 - Verificación final de defectos corregidos
- Los resultados de las pruebas serán entregados formalmente al desarrollador al cierre del proceso, acompañados del registro de defectos y observaciones.

5.17.3. Objetivos

- Identificar y documentar los riesgos potenciales que puedan impactar el proceso de pruebas.
- Definir estrategias para mitigar o eliminar dichos riesgos.
- Establecer planes de contingencia para reducir el impacto si los riesgos ocurren.
- Mantener la trazabilidad y control de la gestión de riesgos.

5.17.4. Equipo de trabajo y roles

Para garantizar una adecuada gestión de los riesgos durante las pruebas, se han definido roles entre los integrantes del equipo.

Cada miembro participa activamente en la ejecución de las pruebas, y además asumir una función orientada a la organización, consolidación y comunicación de los resultados obtenidos. A continuación, se detalla la asignación de responsabilidades dentro del equipo:

| Cargo | Nombre | Funciones principales |
|---|--------------------|---|
| Responsable de control y registro | Jennifer Cruz | <ul style="list-style-type: none">• Gestionar la planificación y desarrollo de las pruebas.• Apoyar en la ejecución de pruebas asignadas.• Preparar informe de resultados. |
| Responsable de comunicación de hallazgos | Nathaly Rodríguez | <ul style="list-style-type: none">• Comunicar los resultados y hallazgos al desarrollador.• Participar en la ejecución de pruebas asignadas.• Preparar informe de resultados. |
| Responsable de consolidación de resultados | Mauricio Hernández | <ul style="list-style-type: none">• Recopilar los hallazgos detectados• Participar en la ejecución de pruebas asignadas.• Preparar informe de resultados. |
| Desarrollador (usuario) | Dennis Orellana | <ul style="list-style-type: none">• Recibir el informe final con resultados del testing y aplicar correcciones necesarias. |

Tabla 12. Equipo responsable de actividades a desarrollar.

5.17.5. Criterios de evaluación de riesgos

- **Alto:** Riesgo que bloquea la ejecución de pruebas críticas o afecta la estabilidad del sistema.
- **Medio:** Riesgo que impacta parcialmente la ejecución o cobertura de pruebas.
- **Bajo:** Riesgo con impacto mínimo, no bloquea la ejecución de pruebas.

5.17.6. Plan de comunicación

- **Registro de hallazgos:** Cada tester documenta sus resultados usando un formato.
- **Consolidación:** El responsable designado consolida los hallazgos al final de las pruebas.
- **Entrega:** Informe final entregado al desarrollador con todos los riesgos y hallazgos detectados.

5.17.7. Identificación y evaluación de riesgos

Los siguientes riesgos están relacionados con el funcionamiento del sistema, la infraestructura y las herramientas utilizadas durante las pruebas. Incluyen posibles fallos técnicos, pérdidas de datos o incidencias operativas que podrían afectar la estabilidad o disponibilidad del entorno de pruebas.

| ID | Riesgo identificado | Nivel | Acciones de mitigación | Acciones de contingencia |
|----|--|---------|---|---|
| R1 | Caída del sistema durante pruebas de mantenimiento | Crítico | <ul style="list-style-type: none"> • Monitorear constantemente el sistema durante la ejecución de pruebas. • Ejecutar solo pruebas críticas primero para evitar sobrecargar el sistema. | <ul style="list-style-type: none"> • Documentar inmediatamente cualquier falla detectada. • Reprogramar pruebas no críticas hasta la recuperación del entorno principal. |
| R2 | Problemas de rendimiento en hardware limitado | Medio | <ul style="list-style-type: none"> • Limitar la cantidad de casos de prueba simultáneos en la máquina local. • Cerrar aplicaciones o procesos innecesarios en la máquina donde se ejecutan las pruebas. | <ul style="list-style-type: none"> • Documentar cualquier fallo o lentitud detectada durante las pruebas. • Dividir la ejecución de pruebas en bloques más pequeños para adaptarse a la capacidad del hardware. |
| R3 | Falta de cobertura en por tiempo limitado | Media | <ul style="list-style-type: none"> • Priorizar casos críticos. | <ul style="list-style-type: none"> • Registrar los casos no ejecutados y justificar en el informe final. |
| R4 | Comunicación insuficiente dentro del equipo | Bajo | <ul style="list-style-type: none"> • Establecer reuniones breves de coordinación antes de iniciar pruebas. | <ul style="list-style-type: none"> • Documentar cualquier discrepancia y unificar criterios antes de consolidar el informe. |
| R5 | Dificultad para reproducir errores detectados | Medio | <ul style="list-style-type: none"> • Mantener registros detallados. • Capturar evidencia (pantallazos, logs) de cada error. | <ul style="list-style-type: none"> • Registrar evidencia • Incluir observaciones en el informe final para referencia futura. |

Tabla 13. Identificación de riesgos, mitigación y contingencia (Parte I).

| ID | Riesgo identificado | Nivel | Acciones de mitigación | Acciones de contingencia |
|----|---|-------|--|---|
| R5 | Fallo en las herramientas de monitoreo o automatización | Medio | <ul style="list-style-type: none"> • Verificar que las herramientas estén correctamente instaladas y actualizadas. • Revisar dependencias y configuraciones necesarias • Realizar pruebas cortas previas (smoke tests) para confirmar su funcionamiento | <ul style="list-style-type: none"> • Documentar cualquier error de la herramienta y los casos afectados. • Continuar la ejecución de las pruebas críticas de forma manual mientras se resuelve el problema. |
| R7 | Bugs de alta prioridad bloquean la ejecución | Alto | <ul style="list-style-type: none"> • Priorizar primero la ejecución de casos que no dependan del módulo afectado. • Revisar los flujos críticos antes de iniciar la prueba para anticipar posibles bloqueos. | <ul style="list-style-type: none"> • Documentar el bug y los casos afectados. • Suspender temporalmente las pruebas en la sección afectada y continuar con otras áreas del sistema. • Incluir el hallazgo en el informe final. |
| R8 | Falta de claridad o ambigüedad en los requisitos funcionales. | Medio | <ul style="list-style-type: none"> • Preparar listas de preguntas o dudas para aclarar con el equipo antes de ejecutar los casos. | <ul style="list-style-type: none"> • Documentar cualquier ambigüedad detectada • Ejecutar los casos basándose en la interpretación más lógica • Incluir observaciones en el informe final. • Consultar al Dev. |

Tabla 14. Identificación de riesgos, mitigación y contingencia (Parte II).

5.17.8. Matriz de probabilidad e impacto

Se presenta una evaluación visual de los riesgos identificados en el sistema, categorizados según su nivel de impacto (Crítico, Alto, Medio, Bajo) y su probabilidad de ocurrencia (Alta, Media, Baja). Esta representación permite una rápida identificación de riesgos críticos, así como la documentación de riesgos con menor impacto, pero alta probabilidad, que podrían requerir mitigación preventiva.

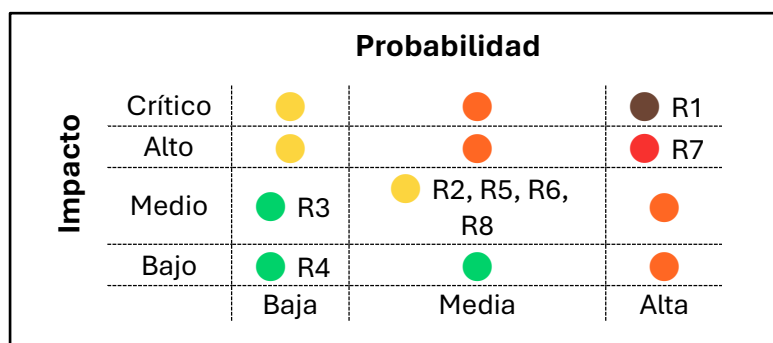


Ilustración 8. Matriz de evaluación de riesgos por impacto vs probabilidad.

5.18. Aprobaciones

E Plan de Pruebas sirve como el documento de compromiso formal para la ejecución de las actividades de testing y establece el alcance, los criterios de éxito, los recursos y el cronograma para la evaluación del sistema de comercio electrónico. La aprobación de este plan es un requisito indispensable para autorizar el inicio de la fase de ejecución de pruebas:

| Rol | Nombre | Fecha de Aprobación |
|-------------------|---|---------------------|
| Asesor | Ing. Sandra Romero | 11 de oct. de 25 |
| Líder de Proyecto | Nathaly Adriana Rodríguez Martínez | 11 de oct. de 25 |
| Miembro Técnico | Jennifer Tatiana Cruz Castro Nathaly Adriana Rodríguez Martínez Mauricio Daniel Hernández Jovel | 11 de oct. de 25 |

Tabla 15. Registro de aprobaciones.

CAPITULO III

CASO DE ESTUDIO

6. Caso de estudio

6.1. Antecedentes

El sistema de comercio electrónico analizado en esta investigación fue desarrollado por un único programador independiente y actualmente se encuentra en producción. Su propósito principal es facilitar la gestión de productos, órdenes, inventario, clientes y usuarios dentro de un entorno web que atiende tanto a consumidores minoristas como a vendedores externos. El sistema fue implementado utilizando el framework Laravel para el backend, MySQL como base de datos relacional y Blade con JavaScript para la capa de presentación. Estas tecnologías proporcionan una estructura funcional estable, aunque sin la integración de procesos formales de aseguramiento de calidad durante su desarrollo inicial.

En su estado previo al inicio del proyecto, el sistema operaba de manera efectiva desde el punto de vista funcional, pero no contaba con una estrategia documentada de pruebas ni con registros que permitieran evaluar su desempeño o la confiabilidad de sus componentes. Las pruebas realizadas eran de tipo exploratorio, ejecutadas de manera manual y sin seguimiento sistemático. No existían métricas de rendimiento, trazabilidad de defectos ni herramientas automatizadas que validaran la integridad del sistema ante cambios o actualizaciones. Esto provocaba que los posibles errores en producción fueran detectados tardíamente, lo que representaba un riesgo tanto para la estabilidad operativa como para la experiencia del usuario final.

Además, el sistema presentaba limitaciones asociadas a la falta de documentación técnica y a la dependencia de un único desarrollador para todas las tareas de mantenimiento y actualización. Esta situación dificultaba la identificación oportuna de fallos en módulos críticos, como el flujo de creación de órdenes o la administración del inventario. A nivel estructural, el proyecto tampoco contaba con un entorno de pruebas definido ni con procedimientos para validar la seguridad, la usabilidad o la eficiencia del sistema frente a diferentes condiciones de carga.

Ante estas deficiencias, se planteó la necesidad de diseñar e implementar un plan integral de pruebas de calidad, que permitiera establecer un proceso sistemático para verificar la funcionalidad y el rendimiento del sistema en cada uno de sus módulos principales. El objetivo fue crear una base técnica que asegurara la detección temprana de defectos, la documentación de resultados y la posibilidad de incorporar prácticas de automatización para futuras versiones del software.

6.2. Modelado de negocio

1. Descripción general del negocio

La empresa objeto de estudio es una pequeña distribuidora de productos farmacéuticos sin prescripción médica, productos de higiene personal y de belleza. Como parte de su proceso de transformación digital, la empresa decidió implementar una plataforma de comercio electrónico para modernizar sus operaciones. Esta iniciativa ha permitido reducir errores en la gestión de pedidos e inventario, mejorar la eficiencia operativa y ofrecer una experiencia de compra más profesional a sus clientes.

- **Identificación del sistema:** Para los efectos del presente proyecto, el sistema evaluado corresponde a un entorno replicado de una plataforma de comercio electrónico implementada por la empresa distribuidora “Importadora Guadalajara”, creado con fines de evaluación técnica, y con el propósito de analizar su comportamiento y aplicar diversas técnicas de testing. *Véase el detalle de su configuración en el Capítulo III: Entorno de prueba.*
- **Objetivos estratégicos del E-commerce:** La implementación de esta plataforma es parte de la transformación digital de la distribuidora, buscando principalmente:
 - Reducir errores en la gestión de pedidos e inventario.
 - Mejorar la eficiencia operativa general.
 - Ofrecer una experiencia de compra profesional y fluida al cliente.
 - Facilitar el control de stock y la generación de reportes comerciales para la toma de decisiones.

2. Actores del negocio

En un sistema e-commerce, los actores representan a todas las personas o roles que interactúan con la plataforma para realizar las diferentes operaciones del negocio. La siguiente tabla describe los actores que intervienen en el funcionamiento del e-commerce.

| Actor | Descripción / Rol |
|--------------------|--|
| Cliente (Customer) | Usuarios previamente registrados con acceso al sistema para consultar el catálogo de productos, gestionar su carrito, etc. |
| Vendedor (Seller) | Usuarios internos encargados de ingresar manualmente los pedidos al sistema. |
| Administrador | Supervisa todas las operaciones, gestiona usuarios, ordenes productos, reportes, proveedores, inventarios. |

Tabla 16. Actores principales del sistema e-commerce.

3. Procesos principales del negocio

La siguiente tabla presenta los módulos y submódulos del sistema, describiendo las funciones específicas que cada uno cumple. Esta información permite comprender cómo el sistema soporta los procesos principales del negocio, desde la gestión de productos, clientes y órdenes, hasta el control de inventarios y la administración de usuarios. *Véase el detalle visual del funcionamiento del sistema que complementan esta descripción en el Anexo 3: Diagramas y flujos del sistema*

| Módulo | Submódulo | Función |
|-----------------|------------------------|---|
| Products | Dashboard Product | Muestra tabla de los productos registrados, filtros y acciones para crear, editar, eliminar y restaurar producto. |
| | Add Product | Permite registrar nuevos productos en el sistema. |
| Orders | Dashboard Orders | Muestra tabla de órdenes creadas, filtros y acciones para editar, tipo de envío y pago, comprobante de la orden y exportar reporte. |
| | Orders | Registra las órdenes de productos realizadas por los clientes. |
| | Modified Orders | Permite visualizar órdenes y su detalle. Muestra el comprobante de la orden para su impresión |
| | Product Sales Report | Genera reportes de ventas por producto y usuario en un rango de fechas. |
| Inventory | Create Warehouse | Permite crear y registrar nuevos almacenes. Organización física de inventarios. |
| | Inventory by Warehouse | Visualiza el stock disponible por almacén y producto. Control logístico del inventario. Exportación a Excel. |
| | Kardex | Consultar los movimientos de inventario de un producto/s. |
| | Load Stock | Carga las cantidades de stock en los almacenes. |
| Users | Dashboard Users | Muestra tabla de usuarios creados y acciones de crear y editar. |
| | Register User | Permite registrar usuarios del sistema y editar su información |
| Customers | Dashboard Customer | Muestra tabla de clientes creados, acciones para crear, editar y eliminar |
| | Add Customer | Registrar crear y registrar nuevos clientes en el sistema. |
| Category | Dashboard Customer | Muestra tabla de categorías creadas, acciones para crear, editar y eliminar |
| | Add Category | Permite crear y registrar nuevas categorías de productos. Organización de productos en el sistema. |
| Purchase orders | Add New Order | Permite crear nuevas órdenes de compra. |
| Login | Login | Permite al usuario autenticarse en el sistema para acceder a funcionalidades personalizadas. Accesible para todos los roles. |

Tabla 17. Detalle de los módulos principales del sistema.

6.3. Reglas de negocio

Las reglas de negocio definen las condiciones y/o restricciones que el sistema debe cumplir para garantizar el correcto funcionamiento de los procesos y la integridad de la información. Estas reglas aseguran que solo los usuarios autorizados puedan realizar operaciones, que los pedidos sean válidos y que la información de clientes, productos y órdenes se gestione de manera segura y controlada.

A continuación, se definen las condiciones del sistema a cumplir:

- **Acceso restringido:** Solo los usuarios registrados (clientes B2C, revendedores B2B, vendedores y administradores) pueden acceder al sistema; el público general no tiene acceso.
- **Registro de clientes:** Los clientes deben estar previamente registrados y autorizados por un vendedor o administrador para poder realizar compras.
- **Gestión de inventario:** Los productos que no tengan stock disponible no pueden ser ordenados. Con cada venta registrada se actualiza inventario.
- **Validación de órdenes:** Cada orden debe contener al menos un producto. La orden debe estar asociada a un método de pago válido (efectivo o cheque). Los clientes regulares solo pueden visualizar sus propias órdenes.
- **Gestión de productos y categorías:** Solo los usuarios con permisos pueden crear, editar o eliminar productos y categorías. Los productos deben estar asociados a una categoría válida.
- **Roles y permisos:** Los vendedores pueden registrar, editar y visualizar órdenes, gestionar clientes y visualizar las categorías y productos existentes. El administrador cuenta con todos los permisos, a excepción de crear órdenes.
- **Facturación como comprobante:** Cada orden completada debe generar una factura, la cual sirve como respaldo legal y financiero de la transacción. La factura se emite al momento de que la orden ha sido creada
- **Seguridad de la información:** No se permite el acceso directo a funciones administrativas por parte de usuarios no autorizados.
- **Reportes:** Todas las transacciones y movimientos deben ser registradas para generar reportes confiables.

6.4. Necesidades del negocio

En el entorno digital actual, los sistemas de comercio electrónico constituyen una herramienta fundamental para la competitividad de los negocios, ya que permiten ampliar su alcance comercial, optimizar procesos y ofrecer experiencias personalizadas al cliente.

Sin embargo, cuando estos sistemas son desarrollados y mantenidos por un único profesional independiente, enfrentan desafíos significativos relacionados con la ausencia de procesos formales de aseguramiento de calidad, lo que incrementa el riesgo de fallos funcionales, pérdida de rendimiento y baja escalabilidad.

Para el caso del sistema en estudio, se ha identificado que la cobertura de pruebas es limitada, los procedimientos de verificación no están documentados y no existe automatización que permita validar los módulos críticos de forma continua. Esta situación genera vulnerabilidades en la estabilidad del sistema, afecta la experiencia del usuario final y representa un obstáculo para su crecimiento sostenible.

A partir del análisis FODA, se determinó que las principales necesidades del negocio son:

- Garantizar la estabilidad operativa del sistema, reduciendo el riesgo de defectos en producción que puedan afectar procesos clave como la gestión de órdenes, inventario y clientes.
- Implementar un proceso de aseguramiento de calidad estructurado, que permita documentar, automatizar y mantener pruebas de manera continua.
- Optimizar los tiempos de respuesta y la experiencia del usuario, asegurando que las operaciones del sistema se realicen de forma rápida, segura e intuitiva.
- Proveer métricas objetivas sobre la calidad del software, facilitando la toma de decisiones basada en datos y la priorización de mejoras.
- Fortalecer la reputación profesional del desarrollador independiente, mediante la adopción de buenas prácticas de testing que aumenten la confiabilidad del sistema y su valor competitivo en el mercado.

Estas necesidades reflejan la brecha existente entre el estado actual del sistema y el estado deseado, caracterizado por un entorno de pruebas integral que respalde la evolución del proyecto con evidencias verificables y resultados medibles.

| FORTALEZAS + | DEBILIDADES – |
|--|---|
| <ul style="list-style-type: none"> • Cobertura modular estratégica: enfoque en módulos críticos asegura alto impacto operativo. • Metodología ágil integrada: la iteraciones rápidas y adaptables que alinean desarrollo de pruebas con estándares de calidad. • Automatización y eficiencia: la inclusión de pruebas automatizadas optimiza tiempos y permite mantener la calidad en el futuro. • Toma de decisiones basada en datos: métricas cuantificables para priorizar mejoras. | <ul style="list-style-type: none"> • Recursos limitados: restricciones de tiempo/presupuesto del desarrollador pueden reducir profundidad de pruebas. • Cobertura manual insuficiente: riesgo de omitir escenarios complejos sin un equipo de QA dedicado. • Dependencia tecnológica: el framework requiere de un navegador real para pruebas E2E, incrementando la complejidad y reduciendo escalabilidad en suites de testing grandes. |
| OPORTUNIDADES + | AMENAZAS – |
| <ul style="list-style-type: none"> • Demanda creciente de calidad en ecommerce: el crecimiento del ecommerce requiere soluciones estables, seguras y rápidas. • Proyección internacional: la aplicación de un plan puede ayudar al desarrollador a posicionarse ante nuevos clientes. • Diferenciación profesional: aplicar buenas prácticas de calidad puede ser un valor agregado competitivo para un freelancer y pequeñas empresas. • Integración de herramientas open-source: existen muchas herramientas gratuitas para pruebas que facilitan la implementación del plan sin altos costos. | <ul style="list-style-type: none"> • Resistencia al cambio: desconocimiento del valor del testing por parte de stakeholders, llevando a recortes en el plan • Plazos ajustados: puede limitar el tiempo necesario para diseñar y ejecutar pruebas correctamente. • Mantenimiento de scripts: aunque ahorran tiempo, también requieren mantenimiento cuando cambia el sistema. • Competencia creciente: presión para mejorar calidad y velocidad debido a prácticas avanzadas de testing en competidores. • Falta de inversión adecuada: la falta de asignación de recursos suficientes al testing, podría aumentar riesgos en bugs en producción. |

Tabla 18. Análisis FODA

6.5. Levantamiento de requerimientos de pruebas

El levantamiento de requerimientos es una etapa fundamental en el proceso de aseguramiento de la calidad del software. ya que permite identificar y documentar los elementos que se evaluarán durante las pruebas funcionales y no funcionales. En esta fase se busca comprender a fondo el comportamiento esperado del sistema, los flujos de trabajo y las restricciones de cada módulo.

El objetivo es establecer una base sólida para diseñar los casos de prueba y garantizar que los requerimientos reflejen fielmente las necesidades del usuario final y los objetivos del proyecto. De esta manera, se definieron los criterios correspondientes tanto para los requerimientos funcionales, relacionados con las acciones y comportamientos del sistema, como los requerimientos no funcionales, que incluyen aspectos de rendimiento, seguridad, usabilidad y compatibilidad.

6.5.1. Criterios para el levantamiento de requerimientos

Mediante el proceso de recopilación, análisis y validación de información, se buscó garantizar que las pruebas reflejen con precisión las necesidades del usuario y los objetivos del proyecto. Para ello, se aplicaron criterios y técnicas específicas que se detallan a continuación.

1. Metodología empleada

Se aplicó un enfoque ágil, basado en reuniones de aclaración de dudas. Esta estrategia permitió recopilar los requerimientos de forma incremental y flexible, adaptándose a los cambios que surgieron durante la revisión de cada módulo del sistema.

El enfoque ágil permitió realizar el análisis funcional y verificación directas sobre el sistema con el fin de identificar comportamientos esperados y la validación preliminar de los módulos. Con dicha dinámica incremental, fue posible ajustar los requerimientos conforme se profundizaba en cada componente del sistema, asegurando condiciones verificables y trazables para el diseño posterior de los casos de prueba.

En resumen, la metodología empleada contribuyó a consolidar información precisa, reproducible y alineada con los criterios de calidad definidos y que sirviera como base sólida para la elaboración de los casos de prueba.

2. Instrumentos de levantamiento de información

Para obtener los requerimientos del sistema se emplearon diversos instrumentos que facilitaron la comunicación, la validación y el registro de la información, especialmente ante la ausencia de documentación técnica o manuales formales. Entre ellos destacan:

- **Reuniones y pláticas informales con el desarrollador:** Permitieron comprender la lógica interna del sistema, las dependencias entre módulos y las reglas de negocio aplicadas.
- **Pruebas manuales exploratorias:** Se ejecutaron sobre las funcionalidades disponibles, con el propósito de observar su comportamiento real y detectar posibles diferencias entre lo esperado y lo implementado.
- **Observación directa:** Durante la interacción con las distintas vistas y componentes del sistema se identificaron flujos alternos, validando el cumplimiento de los requisitos definidos, permitiendo ajustar los requerimientos.
- **Registro documental:** Se elaboraron descripciones detalladas de cada requerimiento, lo que facilitó su organización, seguimiento y trazabilidad durante las fases posteriores del proceso de pruebas.

3. Herramientas utilizadas

Durante el proceso de levantamiento de requerimientos se emplearon distintas herramientas de apoyo que facilitaron la documentación, organización y representación visual de la información obtenida:

- **Microsoft Word:** Utilizado para la redacción del borrador inicial y la descripción narrativa de los requerimientos identificados.
- **Microsoft Excel:** Empleado para estructurar los requerimientos en tablas, asignar identificadores y clasificarlos por módulo o tipo.
- **AzureDevops:** Incorporado como plataforma para la generación y gestión de reportes y resultados, además de registrar los casos de prueba, organizar el backlog y estructurar los sprints. Esta integración aseguró la trazabilidad entre requerimientos, pruebas y hallazgos en las etapas posteriores del ciclo de calidad.

6.6. Pruebas funcionales y no funcionales

Para la toma de requerimiento se tomó en cuenta los módulos en función de su importancia estratégica para el correcto funcionamiento y operación del negocio, tales como la gestión de productos, control de órdenes e inventarios y gestión de usuarios.

6.6.1. Requerimientos funcionales

Los requerimientos funcionales definen el comportamiento esperado del sistema desde el punto de vista del usuario y de los procesos que soportan la operación del negocio. Estos requerimientos establecen qué debe hacer el sistema y qué funcionalidades deben estar disponibles para cumplir con los objetivos comerciales y operativos definidos. A continuación, se detallan las funcionalidades específicas a cumplir cada módulo por perfil de usuario:

- **Requerimientos funcionales generales**

Los siguientes requerimientos establecen el comportamiento esperado de la interfaz de inicio de sesión, así como las validaciones y redirecciones específicas según el tipo de usuario.

| ID | Módulo | Descripción |
|-----|--------|--|
| L01 | Login | Formulario con campos “Email” y “Password”; ambos obligatorios; email válido |
| L02 | | Botón “Iniciar Sesión” habilitado solo cuando ambos campos estén completos |
| L03 | | Mensaje de error “Email o contraseña incorrectos” |
| L04 | | Redirección según rol: Admin → /login, Seller → /ecommerce/seller/login, Customer → /customer-ecommerce/seller/login |

Tabla 19. Requerimientos de autenticación de usuarios.

- **Funcionalidades por módulo - vista administrador**

A continuación, se detallan los requerimientos funcionales, no funcionales y las reglas de negocio que deben validarse en cada módulo correspondiente a la vista del administrador. Esta definición busca garantizar la calidad del sistema, la seguridad de los procesos y una experiencia de uso eficiente y confiable para los usuarios con rol administrativo.

| ID | Módulo | Sub módulo | Descripción | |
|------|----------|----------------------|---|---|
| P01 | Products | - | Visualización de productos en tabla con columnas: ID, Name, Category, Stock, Last update. Incluir paginación | |
| P02 | | | Botón “Add New Product” redirecciona a página de creación con campos: Name, Description, Category, Image. Validación de campos e imagen. | |
| P03 | | | Cada fila incluye acciones: Edit, Move to Trash, Manage Prices. Edit y Manage Prices redireccionan; Trash confirma eliminación. | |
| P04 | | | Barra de búsqueda con texto libre para filtrar productos. Búsqueda insensible a mayúsculas. | |
| P05 | | | Botón “Save” valida que todos los campos estén completos y correctos. Confirmación rápida tras guardar. | |
| P06 | | | “Move to Trash” mueve producto a pestaña “Trashed”. Producto desaparece de “Active” y aparece en “Trashed”. | |
| O01 | Orders | Orders | Permite filtros simultáneos: Customer, Seller, Date, Payment Status, Delivery Status, Origin, Product. | |
| O02 | | | Tabla dinámica con columnas: ID, Customer, Total, Origen, Fecha, Estado y Tracking. Incluye paginación. | |
| O03 | | | Acciones por fila: Detail, Edit Order, Edit/Ship, Update Order Status, Print, Tracking, Trash, Delete. Algunas abren modal o redireccionan. Impresión genera PDF. | |
| O04 | | | Botones globales: Filter, Clear, Export to Excel con datos | |
| MO01 | | Modified Orders | Permite filtros simultáneos: Customer, Seller, rango de fechas, Payment, Delivery Status, Origin y Product. Funcionalidad idéntica a Ordenes. | |
| MO02 | | | Acciones por fila: “Order Detail” y “Print Order”. Impresión genera PDF con formato y datos completos. | |
| SR01 | | Product Sales Report | Implementación de filtros: Select User, Start Date, End Date. Fecha en formato dd/mm/yyyy. | |
| SR02 | | | Mostrar tabla de ventas y el total de productos vendidos debe mostrarse correctamente. | |
| SR03 | | | Botones globales: Filter, Export to Excel. Exportación genera archivo .xlsx con datos visibles. | |
| I01 | | Inventory | Warehouse | Botón “Create Warehouse” con campos: Name, Address, State. Todos los campos son obligatorios. |
| I02 | | | | Tabla con listado de bodegas registradas. Muestra: ID, Name, Address, State. No se podrá editar la información. |
| I03 | | | | Botones globales: “Filter” y “Clear” en la vista. “Clear” limpia todos los filtros aplicados. |

Tabla 20. Requerimientos por módulo/submódulos – Vista administrador (Parte I).

| ID | Módulo | Sub módulo | Descripción |
|------|-----------|------------------------|---|
| I04 | Inventory | Inventory by Warehouse | Tabla de inventario por almacenes. Incluir paginación. |
| I05 | | | Botones: Filter, download Excel. Exportación genera archivo .xlsx con datos visibles. |
| I06 | | Kardex | Filtros: Product, Month, Year. Validación estricta: mes entre 1 y 12; año en formato YYYY. |
| I07 | | | Tabla de historial de entradas y salidas de inventario. Retorna datos precisos de movimientos y totales. |
| I08 | | Load Stock | Selección de warehouse y carga de archivo Excel. Incluir columnas: warehouse, product, cantidad y date. |
| I09 | | | Botones: Cargar, Descargar Plantilla Excel. Descarga genera archivo .xlsx con encabezados de productos cargados. |
| C01 | Customer | - | Registro de cliente con los campos: Name, Phone, Address, Email, Password, State. Todos los campos son requeridos. |
| C02 | | | Tabla con clientes del mismo estado. Acciones para Editar, cambiar contraseña, deshabilitar. |
| CO3 | | | Envío automático de credenciales por email tras el registro exitoso del cliente. |
| CO4 | | | Generación automática de contraseña segura al registrar un nuevo cliente. |
| CO5 | | | Filtro de clientes por Estado geográfico y búsqueda combinada por nombre + Estado. |
| U01 | Users | - | Botón “Crear User” con campos obligatorios: Name, Email, Password, Role. Todos los campos son requeridos. |
| U02 | | | Campo “Password” con funcionalidades: generación automática de contraseña y opción para ocultar caracteres. |
| U03 | | | Tabla de usuarios. Muestra: ID, Name, Email, Role. Incluye acción “Edit” por fila. |
| U04 | | | Campo “Role” con opciones limitadas. Solo permite seleccionar “Administrador”. |
| CA01 | Category | - | Botón “Crear Category” con campos obligatorios: Name, Description. Validar que sean obligatorios y únicos. |
| CA02 | | | Permitir subir imagen representativa de la categoría. Validar formato (PG, JPEG, PNG, GIF). |
| CA03 | | | Botón “Create Category” registra la nueva categoría. Muestra ID, Name, Email, Role. Incluye acción “Edit” por fila. |
| CA04 | | | Botón “Back to List”. Validar que al regresar no se pierda integridad de datos en la lista. |

Tabla 21. Requerimientos por módulo/submódulos – Vista administrador (Parte II).

- **Funcionalidades por módulo - vista seller**

Se describen los requerimientos funcionales correspondientes a los módulos disponibles en la vista del perfil Vendedor (Seller). Estos requerimientos definen el comportamiento esperado del sistema desde la perspectiva de este rol.

| ID | Módulo | Sub módulo | Descripción |
|------|----------|-----------------|---|
| O01 | Orders | Orders | Permite filtros simultáneos: Customer, Seller, Date, Payment Status, Delivery Status, Origin, Product. |
| O02 | | | Tabla dinámica con columnas: ID, Customer, Total, Origen, Fecha, Estado y Tracking. Incluye paginación. |
| O03 | | | Acciones por fila: Detail, Edit Order, Edit/Ship, Update Order Status, Print, Tracking, Trash, Delete. Algunas abren modal o redireccionan. Impresión genera PDF. |
| O04 | | | Botones globales: Filter, Clear, Export a Excel con datos |
| MO01 | | Modified Orders | Filtros combinables por Customer, Seller, rango de fechas, Payment, Delivery Status, Origin y Product. Funcionalidad idéntica a Orders. |
| MO02 | | | Acciones por fila: “Order Detail” y “Print Order”. Impresión genera PDF con formato y datos completos. |
| C01 | Customer | - | Registro de cliente con los campos: Name, Phone, Address, Email, Password, State. Todos los campos son requeridos. |
| C02 | | | Tabla con clientes del mismo estado. Acciones para Editar, cambiar contraseña, deshabilitar. |
| CO3 | | | Envío automático de credenciales por email tras el registro exitoso del cliente. |
| CO4 | | | Generación automática de contraseña segura al registrar un nuevo cliente. |
| CO5 | | | Filtro de clientes por Estado geográfico y búsqueda combinada por nombre + Estado. |
| P01 | Purchase | - | Al iniciar sesión, se muestra la pantalla de categorías. Validar que la vista inicial corresponda a categorías. |
| P02 | | | Al seleccionar una categoría, se muestran los productos asociados. |
| P03 | | | Se puede agregar o remover productos del carrito. Validar que el stock permita agregar y remover productos |
| P04 | | | Validar que los cálculos sean precisos. |
| P05 | | | Desde la vista del Seller, se selecciona el estado de pago. Validar que se actualice y sea consistente en facturación. |

Tabla 22. Requerimientos por módulo/submódulos – Vista seller.

- **Funcionalidades por módulo - vista customer**

Se describen los requerimientos funcionales correspondientes a los módulos disponibles en la vista del perfil Cliente (Customer).

| ID | Módulo | Sub módulo | Descripción |
|-----|----------|------------|---|
| O05 | Orders | - | Solo de lectura. Filtros y detalle de orden. |
| P01 | Purchase | - | Al iniciar sesión, se muestra la pantalla de categorías. Validar que la vista inicial corresponda a categorías. |
| P02 | | | Al seleccionar una categoría, se muestran los productos asociados. |
| P03 | | | Se puede agregar o remover productos del carrito. Validar que el stock permita agregar y remover productos |
| P04 | | | Validar que los cálculos sean precisos. |

Tabla 23. Requerimientos por módulo/submódulos – Vista customer.

6.6.2. Requerimientos no funcionales

Los requerimientos no funcionales definen atributos de calidad que complementan el comportamiento funcional del sistema. A continuación, se presenta una tabla que agrupa estos requerimientos. Estos elementos son esenciales para garantizar la estabilidad, confiabilidad y usabilidad del sistema en condiciones reales de operación.

| ID | Categoría | Requerimiento |
|------|----------------|--|
| NF01 | Seguridad | Manejo seguro de sesiones |
| NF02 | Compatibilidad | Compatibilidad con navegadores |
| NF03 | Seguridad | Cierre de sesión tras inactividad |
| NF04 | Seguridad | Headers HTTP seguros |
| NF05 | Rendimiento | Soporte con alto volumen de usuarios concurrentes |
| NF06 | Seguridad | Rechazo de inyección SQL |
| NF07 | Seguridad | Análisis de código seguro |
| NF08 | Seguridad | Verificación de certificados SSL/TLS del sitio web |
| NF09 | Seguridad | Seguridad de contraseñas |
| NF10 | Seguridad | Privacidad de información de usuarios |
| NF11 | UI/UX | Validación visual y resiliencia |
| NF12 | Seguridad | Autenticación de multifactor |

Tabla 24. Requerimientos no funcionales del sistema.

6.7. Casos a automatizar

La siguiente tabla presenta una selección de casos de prueba cuya automatización ha sido priorizada por su alto impacto en la estabilidad, seguridad y trazabilidad del sistema, donde se destaca el beneficio de automatizar el escenario:

| ID | Módulo | Caso de prueba | Justificación |
|------|----------|--|---|
| CP01 | Login | Acceso exitoso | Flujo crítico y repetitivo que valida el acceso principal, asegurando disponibilidad del sistema |
| CP02 | Login | Manejo de credenciales incorrectas | Permite probar múltiples combinaciones de credenciales inválidas para verificar que el sistema rechace el acceso. |
| CP04 | Login | Bloqueo por intentos fallidos | Simula escenarios de ataques automatizados, fortaleciendo la seguridad mediante pruebas consistentes y repetibles |
| CP05 | Login | Redirección por rol | Valida los permisos de usuario, previniendo errores de acceso que podrían comprometer datos sensibles o funcionalidades críticas. |
| CP06 | Login | Redirección por logueo | Asegura de forma consistente que un usuario autenticado no pueda regresar al login o páginas de acceso, validando la gestión de la sesión. |
| CP07 | Products | Crear producto válido | Su automatización permite crear datos de prueba rápidamente para otras pruebas que dependen de la existencia de productos. |
| CP09 | Products | Editar producto existente | La automatización valida que los cambios en la UI se reflejen correctamente en la base de datos sin errores. |
| CP10 | Products | Búsqueda de productos | Ejecuta búsquedas de manera rápida, optimizando el tiempo de ejecución. |
| CP11 | Products | Envío de producto a estado "Trashed" | Valida que el estado lógico del producto cambie en la base de datos y que no sea visible en la interfaz principal, asegurando la integridad de los datos. |
| CP13 | Products | Restauración de producto individual | Caso de uso común que asegura el proceso de recuperación funcione sin errores mediante pruebas repetible. |
| CP15 | Products | Guardar con campos obligatorios vacíos | Permite verificar que las validaciones se disparan correctamente, los mensajes son visibles y que no se guarda ningún registro inválido, sin omitir combinaciones |

Tabla 25. Casos de prueba con automatización completa (Parte I).

| ID | Módulo | Caso de prueba | Justificación |
|------|-----------|--|---|
| CP16 | Products | Manejo de valores numéricos negativos | Previene errores en cálculos y de inventario mediante la validación proactiva de escenarios. |
| CP17 | Orders | Filtrar órdenes por rango de fechas | Facilita probar múltiples rangos de fechas, reduciendo errores humanos en cálculos manuales |
| CP19 | Orders | Actualización de envío y número de seguimiento | Automatiza la verificación de un flujo de complejo, asegurando que la edición y el envío de la orden funcionen sin errores. |
| CP20 | Orders | Modificación de detalles de orden | Valida que los cambios en precios y descuentos se apliquen de forma consistente, lo que es crítico para el negocio. |
| CP21 | Orders | Actualización de estado de pago | Automatiza la verificación de un cambio de estado crítico para el negocio, asegurando que la interfaz y la base de datos se sincronicen correctamente. |
| CP22 | Orders | Exportación de órdenes a Excel | Permite validar de forma automatizada que el archivo exportado contenga los datos correctos. |
| CP24 | Orders | Generación de reporte de ventas por producto | Automatiza la validación de cálculos y filtros complejos en un reporte, lo que sería muy lento y propenso a errores de forma manual. |
| CP25 | Orders | Eliminación de orden | Asegura la consistencia de operaciones de limpieza de datos mediante validación automatizada |
| CP26 | Inventory | Registro exitoso de bodega | Flujo estándar que valida el registro correcto de bodegas, clave para la trazabilidad del inventario. |
| CP27 | Inventory | Filtro por bodega en inventario | Ejecuta búsquedas de manera rápida, optimizando el tiempo de ejecución. |
| CP28 | Inventory | Exportación de inventario a Excel | Permite validar que el archivo se genera correctamente, con estructura y datos esperados, sin depender de revisión manual. Clave para asegurar trazabilidad y consistencia en reportes. |
| CP32 | Customer | Registrar nuevo cliente válido | Es un caso de uso común. Su automatización permite crear datos de prueba rápidamente, optimizando la ejecución de pruebas E2E. |
| CP31 | Inventory | Carga masiva de stock desde Excel | Permite automatizar la validación de un proceso complejo, asegurando que los datos se importen correctamente sin fallos. |
| CP33 | Customer | Editar cliente existente | Valida que los cambios de datos en el cliente se reflejen correctamente en la base de datos y la interfaz, evitando errores de datos. |

Tabla 26. Casos de prueba con automatización completa (Parte II).

| ID | Módulo | Caso de prueba | Justificación |
|------|-------------|--|--|
| CP34 | Customer | Desactivar (eliminar) cliente | Automatiza la verificación de que el cliente cambie de estado en la base de datos, asegurando que no se pueda acceder a él. |
| CP35 | Customer | Búsqueda de cliente por nombre | Ejecuta búsquedas de manera rápida, optimizando el tiempo de ejecución. |
| CP37 | Customer | Generación automática de contraseña | Automatiza la validación de contraseñas para detectar credenciales débiles o predeterminadas, reduciendo el riesgo de accesos no autorizados. |
| CP38 | Customer | Búsqueda de cliente según Estado y combinada | Permite validar de forma automatizada que los datos exportados sean correctos y tengan el formato adecuado. |
| CP39 | Users | Creación de usuario Administrador o Seller | Es un flujo de trabajo estándar que se repite. La automatización ayuda a prevenir errores en la asignación de roles. |
| CP40 | Users | Validación de email único | Elimina problemas de datos duplicados que afectan user experience |
| CP41 | Users | Generación de contraseña de usuario | Automatiza la validación de contraseñas para detectar credenciales débiles o predeterminadas, reduciendo el riesgo de accesos no autorizados. |
| CP42 | Users | Edición de rol de usuario | Protege contra errores de configuración de mediante verificación de cambios de permisos que podrían resultar en asignación de privilegios no autorizada |
| CP43 | Users | Validación de opciones del campo Rol | Automatiza las restricciones del dropdown, asegurando que solo se puedan seleccionar roles válidos. |
| CP44 | Users | Validación de Campos Vacíos | Entrega máxima cobertura de validaciones básicas con mínimo esfuerzo de mantenimiento, previniendo registros corruptos |
| CP45 | Seguridad | Inyección código SQL | Detecta vulnerabilidades críticas de seguridad con pruebas automatizadas que simulan ataques, previniendo brechas de datos. |
| CP47 | Seguridad | Headers HTTP | Asegura el cumplimiento de mejores prácticas de seguridad mediante validación de configuraciones que protegen contra ataques comunes como clickjacking. |
| CP48 | Rendimiento | 100 usuarios concurrentes | Identifica cuellos de botella y límites de escalabilidad con la simulación de carga que es imposible de replicar manualmente, previniendo caídas en momentos críticos. |

Tabla 27. Casos de prueba con automatización completa (Parte III).

| ID | Módulo | Caso de prueba | Justificación |
|------|-------------|---|---|
| CP49 | Rendimiento | Saturación del sistema | Permite identificar los cuellos de botella del sistema antes de la producción, lo cual es imposible de hacer manualmente. |
| CP51 | Seguridad | Gestión de sesión – time-out | Se automatiza el registro y repetición del flujo, la observación del comportamiento visual requiere validación manual. |
| CP52 | Seguridad | Restricción de acceso por rol | Permite validar de forma rápida y repetible los permisos de acceso de cada rol, reduciendo errores humanos y asegurando la consistencia en múltiples módulos |
| CP53 | UI/UX | Compatibilidad con navegadores | Permite probar la aplicación en múltiples navegadores y versiones de forma automática, lo que sería muy lento de forma manual. |
| CP54 | Seguridad | Revisión de código seguro | Se automatiza el escaneo de código, pero la detección de vulnerabilidades complejas requiere análisis humano. |
| CP55 | Seguridad | Validación de certificado SSL/TLS | Monitorea proactivamente la vigencia y configuración de certificados mediante verificaciones periódicas que previenen interrupciones. |
| CP56 | Seguridad | Transporte seguro de credenciales | Garantiza la confidencialidad de datos sensibles mediante validación continua de cifrado que previene interceptación de credenciales en tránsito |
| CP57 | Seguridad | Protección contra fuerza bruta | Valida mecanismos de defensa contra ataques automatizados mediante pruebas que aseguran la efectividad de controles de seguridad en escenarios realistas |
| CP58 | Seguridad | Detección de credenciales predeterminadas/débiles | Automatiza la validación de contraseñas para detectar credenciales débiles o predeterminadas, reduciendo el riesgo de accesos no autorizados. |
| CP59 | Seguridad | Enumeración de cuentas | Previene la filtración de información sensible mediante validación automatizada que asegura que los mensajes de error no revelen información útil para atacantes. |
| CP60 | Seguridad | Validación de “Remember Me” | La automatización porque permite validar de forma consistente y trazable el flujo de sesión y la gestión de cookies. |

Tabla 28. Casos de prueba con automatización completa (Parte IV).

| ID | Módulo | Caso de prueba | Justificación |
|------|-----------|--|---|
| CP61 | Seguridad | Gestión segura de sesiones | Asegura mecanismos de sesión contra vulnerabilidades comunes mediante validación automatizada de atributos de seguridad. |
| CP62 | Seguridad | Bypass de autenticación y MFA | Protege puntos críticos de autenticación mediante pruebas ofensivas automatizadas que validan la efectividad de controles de seguridad. |
| CP63 | Seguridad | Respuestas ante errores y mensajes | Mantiene la confidencialidad del sistema mediante validación automatizada que asegura que los mensajes de error no revelen información útil para atacantes. |
| CP64 | UI/UX | Pruebas de UI/UX y resiliencia | Se automatiza la validación de enlaces y CAPTCHA, pero la experiencia visual y de diseño requiere inspección humana. |
| CP65 | Purchase | Vista inicial de categorías | Redirección post-login es rápida de automatizar y crítica para validar el flujo inicial. Requiere bajo esfuerzo y aporta alto valor en regresiones. |
| CP66 | Purchase | Visualización de productos por categoría | Asegura la integridad de la navegación mediante validación automatizada que previene errores en la asociación producto-categoría |
| CP67 | Purchase | Gestión de productos en carrito | Agregar y remover productos implica múltiples interacciones, pero es un flujo esencial. Requiere esfuerzo medio y permite validar lógica de stock y carrito |
| CP68 | Purchase | Cálculo de totales | Cálculo de totales involucra lógica de negocio. Requiere más tiempo por validaciones dinámicas, pero es clave para evitar errores financieros |
| CP69 | Purchase | Actualización de estado de pago (Seller) | Se automatiza el cambio de estado en la UI, pero validar su impacto en facturación y base de datos requiere integración. |
| CP70 | Category | Validación de campos obligatorios | Validar campos obligatorios es simple y rápida de implementar. Aporta valor inmediato al evitar registros inválidos |
| CP71 | Category | Validación de carga de imagen | Se automatiza la subida y validación técnica del archivo, pero la calidad visual en la interfaz requiere revisión manual. |

Tabla 29. Casos de prueba con automatización completa (Parte V).

| ID | Módulo | Caso de prueba | Justificación |
|------|----------|--|---|
| CP72 | Category | Registro exitoso de categoría | Permite verificar de forma repetible y confiable que el registro de categorías funcione correctamente, asegurando la estabilidad del formulario y la respuesta del sistema. |
| CP74 | Category | Eliminar categoría | Previene la inconsistencia de datos mediante validación automatizada que asegura la limpieza completa y mantiene la integridad. |
| CP75 | Category | Editar una categoría | Garantizar que los cambios se reflejan correctamente en la UI y en el backend, sin depender de pruebas manuales que podrían omitir validaciones clave. |
| CP76 | Orders | Agregar un producto a la orden | Permitir simular múltiples escenarios asegurando que el sistema responde sin errores y los sean cálculos correctos. |
| CP77 | Orders | Remover producto /productos de una orden | Permitir validar que no quedan inconsistencias, que el sistema responde correctamente y que la interfaz refleja los cambios sin errores visuales o lógicos. |

Tabla 30. Casos de prueba con automatización completa (Parte VI).

6.8. Casos que no se van automatizar.

Se detallan los casos de prueba excluidos de automatización por razones de eficiencia o necesidad de validación manual especializada. La decisión responde a dos principios clave:

- 1. Eficiencia técnica:** En ciertos escenarios, el esfuerzo de diseñar, programar y mantener scripts de automatización supera el beneficio obtenido. Casos simples o de baja frecuencia se validan más rápido de forma manual que con automatización.
- 2. Criterio humano indispensable:** Existen validaciones que requieren inspección visual, análisis contextual o juicio experto. La automatización no puede sustituir la percepción humana en aspectos como calidad visual, presentación de datos, interpretación contable o análisis de seguridad.

De esta manera, se optimiza el uso de la automatización en los casos que realmente aportan valor, mientras que los casos manuales aseguran precisión y confiabilidad en áreas críticas

| ID | Módulo | Caso de prueba | Justificación |
|------|--------------|--|--|
| CP03 | Login | Campos obligatorios vacíos del login | Validación manual más rápida que el desarrollo y mantenimiento de automatización para este caso. |
| CP08 | Products | Agregar Imagen en formato en diferentes formatos | La validación de calidad visual y presentación requiere criterio humano que la automatización no puede replicar |
| CP12 | Products | Eliminación en cascada | Necesita control manual para validar relaciones complejas y mantener trazabilidad de auditoría |
| CP14 | Products | Restauración masiva de productos | Requiere selección manual y validaciones contextuales que la automatización no puede proporcionar |
| CP18 | Orders | Visualización de detalle de orden | Requiere una inspección visual y la validación de la presentación de los datos. La automatización no puede reemplazar la inspección humana en este caso. |
| CP23 | Orders | Impresión de detalle de orden | Validación de formato PDF, paginación y calidad de impresión requiere inspección visual manual |
| CP29 | Inventory | Consulta de historial Kardex mensual | Verificación de fórmulas contables y cálculos financieros necesita análisis experto manual. |
| CP30 | Inventory | Verificación del saldo por movimiento | Previene errores mediante validación automática de cálculos de inventario que impactan directamente en estados financieros. |
| CP36 | Customer | Envío de credenciales por email | La prueba depende de un servicio externo y requiere la validación manual del correo electrónico recibido. |
| CP46 | Seguridad | Acceder a vista Admin | La validación de la respuesta HTTP y los logs de seguridad requiere un análisis humano. |
| CP50 | Concurrencia | Edición del mismo producto | La validación se ejecuta de forma manual debido a la complejidad de reproducir escenarios de edición simultánea en pruebas automatizadas. |
| CP73 | Category | Retorno a lista desde Create Category | Navegación simple que no justifica el esfuerzo de automatización. |

Tabla 31. Casos de prueba no automatizables.

6.9. Cobertura de pruebas

El aseguramiento de la calidad de un sistema requiere la aplicación de distintos enfoques, cada uno con objetivos específicos. Los tipos de pruebas se enfocan en aspectos particulares del sistema, como funcionalidad, seguridad, rendimiento y usabilidad.

Se priorizo la trazabilidad y la necesidad de asegurar la integridad funcional, transaccional y operativa. A continuación, se presentan la selección de los tipos de prueba considerados durante el proceso de validación del sistema.

| Tipo de prueba | Estado | Justificación |
|--------------------------|----------|---|
| Funcional | Incluida | Fundamental para asegurar que cada requisito y flujo de negocio (Login, Productos, Órdenes, etc.) funcione correctamente con los requisitos del negocio |
| Integración | Incluida | Requerida para verificar la correcta interacción entre módulos (ej. Login/Redirección por rol, Modificación de Órdenes, etc). Clave para garantizar que los cambios en un módulo no rompan las interfaces con otros. |
| Seguridad | Incluida | Crítica en un sistema que maneja credenciales y datos sensibles, así como la evaluación de codificación segura. Busca prevenir vulnerabilidades antes de que sean explotadas. |
| Rendimiento | Incluida | Esencial para verificar la estabilidad del sistema bajo condiciones reales. Garantiza que el sistema mantenga una experiencia de usuario aceptable bajo la carga operativa esperada. |
| Exploratoria | Incluida | Una exploración manual inicial del sistema para familiarizarse con los flujos y validar la estabilidad general del entorno replicado. |
| Concurrencia | Incluida | Específicamente seleccionada para evitar pérdida de datos en procesos críticos (ej. Edición de mismo producto). Asegura la integridad transaccional en un ambiente multiusuario. |
| Compatibilidad | Incluida | Necesaria para asegurar que la experiencia de usuario sea consistente en los navegadores y dispositivos utilizados por el cliente, lo cual es vital para un sistema en producción. |
| Usabilidad/UI-UX | Incluida | Incluida en casos donde el resultado esperado se enfoca en la claridad y el formato para el usuario final (ej. PDF). |
| Pruebas de Estrés | Incluida | Estas pruebas se aseguran de identificar los límites máximos del sistema y verificar su comportamiento bajo condiciones extremas de carga, asegurando que pueda recuperarse adecuadamente después de superar su capacidad normal. |
| Pruebas Alfa/Beta | Excluida | El sistema ya fue lanzado y está siendo utilizado por los usuarios finales. |
| Regresión | Excluida | No se consideran necesarias en esta fase, ya que no se han realizado cambios sobre funcionalidades previamente validadas, por lo que no existe riesgo de regresión en dicha fase. El sistema se encuentra en funcionamiento por los usuarios finales, y no se han introducido modificaciones. |
| Pruebas de humo | Excluida | No se han realizado cambios estructurales ni despliegues recientes que requieran validación superficial. |

Tabla 32. Resumen de tipos de pruebas aplicados.

6.10. Alcance de pruebas

El plan de pruebas abarca la validación integral de los módulos principales del sistema, con el propósito de garantizar su correcto funcionamiento, estabilidad, seguridad y rendimiento operativo. El alcance se centra en los flujos de negocio críticos que sustentan las operaciones principales del sistema, tales como la autenticación de usuarios, gestión de productos, órdenes, inventario, clientes, usuarios y categorías.

El plan incluye tanto pruebas funcionales como no funcionales, aplicando un enfoque mixto de pruebas manuales y automatizadas, según la criticidad, complejidad y frecuencia de uso de cada módulo.

En esta fase, se ejecutarán los siguientes tipos de prueba:

- **Pruebas funcionales:** Validan el cumplimiento de los requerimientos definidos por módulo, asegurando que cada componente ejecute correctamente las operaciones previstas.
- **Pruebas de estrés:** Evalúan el comportamiento del sistema bajo condiciones extremas de carga o uso, superando los niveles normales de operación. Estas pruebas permiten identificar los límites de capacidad, y verificar la estabilidad del sistema cuando se enfrenta a escenarios de saturación.
- **Pruebas de rendimiento:** Determinan la capacidad del sistema para operar bajo distintas cargas de usuarios concurrentes, midiendo tiempos de respuesta, estabilidad y tasa de error.
- **Pruebas de usabilidad y UX/UI:** Analizan la experiencia del usuario al interactuar con la interfaz del sistema. Estas pruebas permiten identificar barreras de uso, inconsistencias en el diseño y oportunidades de mejora que impactan directamente en la satisfacción y productividad del usuario final
- **Pruebas de seguridad:** Analizan la protección de los datos, autenticación, manejo de sesiones, cifrado y resistencia ante ataques comunes como inyección SQL o fuerza bruta.
- **Pruebas de concurrencia y compatibilidad:** Aseguran la correcta operación en escenarios de múltiples usuarios simultáneos y en distintos navegadores.

El proceso de aseguramiento de calidad se ejecutará sobre los módulos críticos definidos en la siguiente tabla:

| Módulo | Función principal |
|------------------|---|
| Login | Autenticación y gestión de sesión de usuarios. |
| Products | Administración del catálogo de productos y sus características. |
| Orders | Registro, edición y seguimiento de órdenes de compra. |
| Inventory | Control logístico, gestión de bodegas y existencias. |
| Customers | Gestión de clientes y vinculación con órdenes. |
| Users | Administración de usuarios del sistema y roles de acceso. |
| Category | Organización y clasificación de productos. |
| Purchase | Registro de compras a proveedores para abastecimiento. |

Tabla 33. Definición de módulos críticos.

Los módulos complementarios como Seller, States, Suppliers, Coupons y Purchase Orders, quedan fuera del alcance por no impactar directamente en los procesos críticos definidos para esta etapa del plan de pruebas.

Finalmente, el alcance contempla la documentación de resultados y generación de métricas de calidad, tales como tiempos de respuesta, tasa de defectos y cobertura de pruebas, las cuales servirán para medir la mejora continua y brindar recomendaciones al desarrollador del sistema para futuras iteraciones.

6.11. Estrategia de pruebas

La estrategia de pruebas establecida para el sistema, busca garantizar la calidad y confiabilidad del software mediante un enfoque estructurado que combine actividades manuales, automatizadas y parciales, orientadas a validar la funcionalidad, el rendimiento y la seguridad de los módulos críticos del sistema.

El proceso se desarrolló bajo una metodología ágil basada en Scrum, adaptada al contexto del proyecto. A lo largo de tres sprints se planificaron y ejecutaron las pruebas de manera incremental, permitiendo una revisión de los resultados y una retroalimentación directa con el usuario (Dev). Este enfoque facilitó la detección temprana de defectos, la priorización de correcciones y la mejora progresiva del sistema.

La estrategia se centra en la aplicación práctica de los tipos de prueba definidos previamente; funcionales, de integración, seguridad, rendimiento y compatibilidad, priorizando aquellos escenarios que representan mayor impacto en la estabilidad y operatividad del sistema.

Las pruebas automatizadas se enfocaron en flujos repetitivos, críticos y técnicos, mientras que las pruebas manuales se aplicaron en validaciones visuales, de interfaz o dependientes de juicio humano.

Para su ejecución se emplearon herramientas integradas al entorno de desarrollo del sistema, seleccionadas por su compatibilidad con Laravel y su capacidad para cubrir diferentes necesidades de validación. Estas herramientas permitieron ejecutar pruebas unitarias, funcionales, de rendimiento y seguridad de forma eficiente, asegurando trazabilidad y reproducibilidad de los resultados.

- **PHPUnit y Laravel Dusk:** para pruebas unitarias y funcionales automatizadas end-to-end.
- **Apache JMeter:** para la ejecución de pruebas de carga y rendimiento.
- **OWASP ZAP y Qualys SSL Labs:** para la detección de vulnerabilidades y verificación de seguridad en la comunicación.
- **SonarQube:** para el análisis estático del código fuente y detección temprana de errores, malas prácticas y vulnerabilidades.

El proceso de gestión y documentación de las pruebas se llevó a cabo en Azure DevOps, plataforma que permite registrar requerimientos, casos de prueba, defectos y métricas de avance por sprint. Este control garantiza visibilidad del progreso y facilita la toma de decisiones respecto a la cobertura alcanzada y la priorización de nuevos escenarios de prueba.

Como parte de la estrategia, se establecieron criterios de cobertura y seguimiento que permiten medir la efectividad del proceso, garantizando que las pruebas abordaran los escenarios de mayor relevancia operativa. Los resultados obtenidos servirán de referencia para fortalecer la madurez del proceso de aseguramiento de la calidad y optimizar el ciclo de pruebas en futuras versiones del sistema.

6.11.1. Enfoque de ejecución: pruebas manuales vs automatizadas

La estrategia de pruebas adoptada contempló una ejecución secuencial que permitió contrastar directamente los resultados obtenidos mediante pruebas manuales para la obtención de requerimiento, tal como se expone en el apartado 5.5.1 y pruebas automatizadas. Esta comparación fue clave para evidenciar las ventajas concretas de incorporar automatización en el flujo de validación.

La ejecución manual, implicó una alta inversión de tiempo, esfuerzo humano y riesgo de variabilidad en los resultados debido a la intervención directa del ejecutor. En contraste, la ejecución automatizada se centró en los casos previamente definidos y desarrollados con herramientas específicas.

Esta etapa demostró beneficios significativos en términos de eficiencia, al reducir los tiempos de ejecución y permitir la repetición constante de pruebas sin pérdida de precisión. Además, se logró trazabilidad completa de los resultados, y disminución de errores humanos. La automatización también facilitó la generación de evidencias reproducibles, fortaleciendo la calidad del proceso y su defensa ante auditorías o revisiones académicas.

La comparación entre ambos enfoques permitió identificar qué casos son más adecuados para automatización, cuáles requieren validación manual por su complejidad o variabilidad, y cómo optimizar el esfuerzo del equipo en futuras iteraciones. En resumen, la automatización no reemplaza la observación humana, pero sí potencia la capacidad de validación, mejora la consistencia de los resultados y libera recursos para tareas de mayor análisis.

6.12. Análisis de riesgos

El análisis de riesgos tiene como propósito identificar los eventos o condiciones que podrían afectar la ejecución y los resultados del plan integral de pruebas aplicado al sistema de comercio electrónico. Este ejercicio se realizó de forma preventiva, considerando escenarios técnicos, operativos y organizacionales que podrían presentarse durante la validación, según lo establecido en el plan de riesgos y contingencias.

Durante la etapa de planificación, se identificaron riesgos que, en caso de materializarse, podrían comprometer la calidad y estabilidad del sistema. Estos riesgos fueron clasificados en categorías técnicas, operativas y de seguridad, evaluando su nivel de impacto (crítico, alto, medio o bajo) y su probabilidad de ocurrencia (alta, media o baja). Esta clasificación permitió priorizar su atención y definir estrategias de mitigación adecuadas.

Aunque no todos estos riesgos se materializaron durante la ejecución, se documentaron como parte del análisis para establecer medidas preventivas en caso de que llegaran a ocurrir. Entre los escenarios contemplados se incluyen posibles interrupciones del sistema, problemas de rendimiento, limitaciones de cobertura por tiempo, fallos en herramientas de automatización, ambigüedad en los requisitos funcionales y dificultades en el equipo de trabajo como la comunicación interna.

Para cada uno de estos riesgos se definieron medidas de mitigación y contingencia específicas, orientadas a preservar la trazabilidad, reproducibilidad y calidad del proceso de validación. Por ejemplo, se consideró la posibilidad de una cobertura de pruebas limitada debido a restricciones de tiempo o recursos, así como problemas de rendimiento en hardware limitado, errores en la red durante pruebas remotas, y dificultades para reproducir errores detectados. También se contemplaron riesgos organizacionales como la comunicación insuficiente dentro del equipo, y funcionales como la ambigüedad en los requisitos o la duplicidad de órdenes.

El siguiente resumen compila los principales riesgos identificados y las medidas propuestas para su control. Todos estos riesgos, junto con sus respectivas estrategias de mitigación y contingencia, se encuentran detallados en la “Tabla 12 y 13 sobre la Identificación de riesgos, mitigación y contingencia. Este análisis permitió anticipar los posibles incidentes que podrían impactar el cumplimiento del plan de pruebas, garantizando la continuidad de las actividades mediante acciones preventivas y correctivas.

6.13. Roles y responsabilidades

El equipo está conformado por tres integrantes, quienes asumieron funciones técnicas y de gestión de manera colaborativa, manteniendo una comunicación continua y una documentación viva del proceso. La ejecución del plan de pruebas requirió una estructura organizativa que combinara los principios de la metodología ágil Scrum con una distribución operativa específica. Para ello, se definieron roles formales, que permitieron coordinar las etapas de planificación, ejecución y validación técnica del sistema de comercio electrónico.

Además de estos roles metodológicos, se asignaron funciones operativas a cada, con el objetivo de garantizar el registro de los hallazgos, la comunicación efectiva el equipo y el análisis conjunto de los resultados. Esta asignación permitió mantener una documentación viva, una ejecución colaborativa y una gestión de riesgos alineada con los objetivos del proyecto. La siguiente tabla presenta de forma consolidada los roles y responsabilidades asignados, así como sus funciones desempeñadas:

| Rol | Nombre | Responsabilidad | Funciones principales |
|-------------------------------------|--------------------|--|---|
| Scrum Master & Equipo QA | Jennifer Cruz | Responsable de control y registro | <ul style="list-style-type: none"> • Coordina el ciclo Scrum, gestiona riesgos • Diseña y ejecuta pruebas. • Gestiona planificación, ejecuta casos asignados y prepara el informe de resultados. |
| Equipo QA | Nathaly Rodríguez | Responsable de comunicación de hallazgos | <ul style="list-style-type: none"> • Ejecuta pruebas funcionales. • Comunica hallazgos al desarrollador y colabora en la elaboración del informe. |
| Equipo QA | Mauricio Hernández | Responsable de consolidación de resultados | <ul style="list-style-type: none"> • Ejecuta pruebas funcionales y recopila hallazgos detectados y colabora en la elaboración del informe. |
| Desarrollador del sistema | Dennis Orellana | Receptor técnico | <ul style="list-style-type: none"> • Recibe el informe final, aplica correcciones necesarias. |
| Product Owner | Ing. Sandra Romero | Asesor Académico | <ul style="list-style-type: none"> • Supervisora y de acompañamiento metodológico |

Tabla 34. Roles, responsabilidades y funciones del equipo de trabajo.

6.14. Calendario de pruebas

Se muestra a continuación, la planificación de pruebas técnicas y de calidad, con tareas, fechas y equipos asignados. Lo cual permite visualizar el avance, ajustes y evaluaciones en cada fase, facilitando el seguimiento y la coordinación en el equipo.

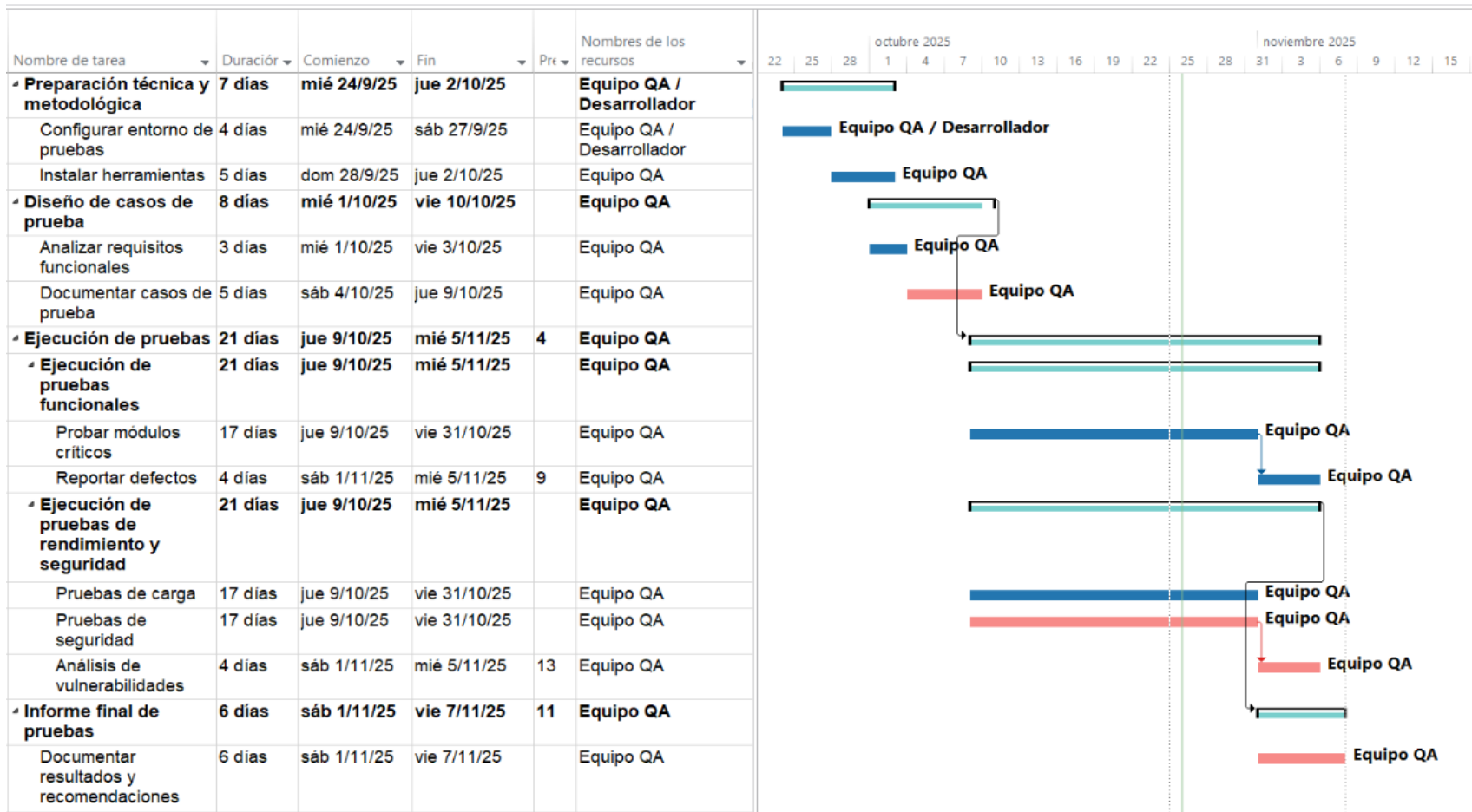


Ilustración 9. Calendario de pruebas

6.15. Ejecución CI/CD

La integración continua y el despliegue continuo (CI/CD) aplicada a pruebas automatizadas con Selenium en Python constituye un componente esencial dentro de los procesos modernos de aseguramiento de la calidad del software. En el contexto planteado, el repositorio principal del proyecto se gestiona en GitHub, el entorno de ejecución del sistema se encuentra alojado en un servidor de Hostinger y la automatización de las pruebas se lleva a cabo mediante GitHub Actions.

Esta arquitectura permite establecer un flujo de trabajo en el cual cada modificación realizada sobre la rama principal del repositorio desencadena de manera automática la verificación del estado funcional de la aplicación mediante la ejecución de pruebas en un entorno controlado.

Asimismo, la práctica de CI/CD fortalece la trazabilidad de los procesos, ya que cada ejecución queda documentada en el historial del repositorio y puede ser revisada en auditorías técnicas o académicas. Esto permite generar métricas de desempeño, identificar patrones de fallos recurrentes y establecer planes de mejora continua que incrementan la confiabilidad del sistema y la eficiencia operativa del equipo de desarrollo.



Ilustración 10. Arquitectura CI/CD.

Cuando se produce un cambio en el repositorio ya sea a través de un push directo o mediante la aprobación de una solicitud de integración

```

# ejemplo_selenium_fill_fields.py
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service as ChromeService
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
#from webdriver_manager.chrome import ChromeDriverManager # opcional, facilita el driver
from dotenv import load_dotenv
import os
import time
from mailjet_rest import Client

# --- CONFIG: obtener credenciales (evitar hardcodear en producción) ---
load_dotenv()
EMAIL = os.getenv("MY_APP_EMAIL_ADMIN", "your_email@example.com")
PASSWORD = os.getenv("MY_APP_PASSWORD_ADMIN", "your_password")
URL = os.getenv("URL_ADMIN_LOGIN", "https://cuscatec.cuscatec.com/login")

api_key = os.environ['MJ_APIKEY_PUBLIC']
api_secret = os.environ['MJ_APIKEY_PRIVATE']
SENDER_EMAIL = os.environ.get('SENDER_EMAIL', 'hj15001@ues.edu.sv')
RECIPIENT_EMAIL = os.environ.get('RECIPIENT_EMAIL', 'mauricioricaldone14@gmail.com')
mailjet = Client(auth=(api_key, api_secret), version='v3.1')

# --- INICIALIZAR DRIVER (Chrome) ---
#options = webdriver.ChromeOptions()
options = Options()
options.add_argument("--headless=new")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")
# options.add_argument("--headless=new") # descomenta si quieres modo headless
driver = webdriver.Chrome(options=options)
#driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), options=options)

try:
    driver.get(URL) # cambia por la URL real

    wait = WebDriverWait(driver, 15)

    # --- Buscar input por name "email" y por id "username" ---
    # Espero que cualquiera de las dos coincidencias esté presente (primero por id, luego por name)
    try:
        username_input = wait.until(EC.presence_of_element_located((By.ID, "username")))
    except:
        username_input = wait.until(EC.presence_of_element_located((By.NAME, "email")))

```

Ilustración 11. Modificación de repositorio

GitHub Actions activa un workflow previamente definido. Dicho workflow especifica los recursos necesarios para la ejecución de las pruebas, incluyendo la instalación del intérprete de Python, los paquetes de Selenium y el controlador de navegación correspondiente, el cual opera en modo headless para permitir la automatización en un ambiente sin interfaz gráfica.

A partir de estas configuraciones, el sistema ejecuta los casos de prueba automatizados, que interactúan con la instancia real de la aplicación alojada en Hostinger, lo que garantiza una evaluación basada en el comportamiento del sistema en producción y no únicamente en entornos locales o simulados.

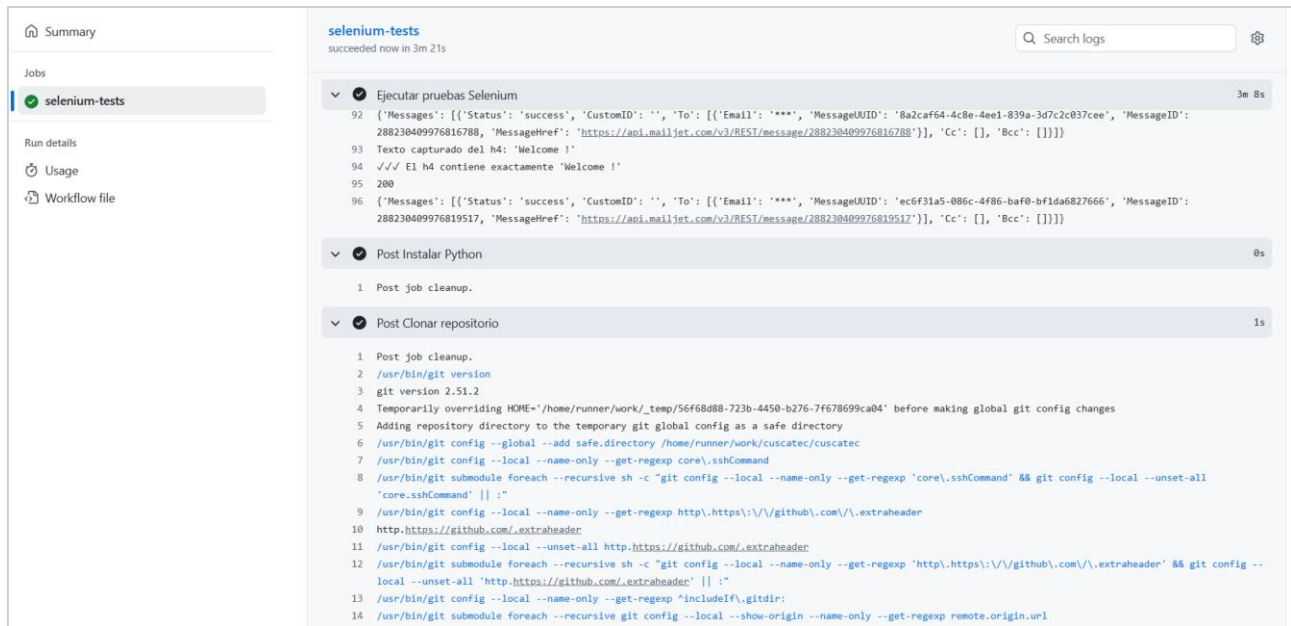


Ilustración 12. Ejecución de pruebas desde Github Actions.

La ejecución automatizada produce un conjunto de resultados que recoge información relevante sobre el estado del software, como la correcta funcionalidad de los flujos críticos, la presencia de anomalías o fallas, y el rendimiento de las interacciones controladas por Selenium. Estos resultados se transforman en un reporte que forma parte del registro histórico del pipeline de CI/CD, contribuyendo así a la trazabilidad del proceso de verificación.

Posteriormente, el mismo workflow incorpora un mecanismo de notificación mediante correo electrónico que remite los resultados obtenidos a los destinatarios designados. Para ello, GitHub Actions emplea un servicio SMTP configurado dentro del flujo, o un script que automatiza el envío del informe, garantizando que los involucrados en el desarrollo reciban una comunicación oportuna sobre el estado de las pruebas.

Gracias a este encadenamiento de procesos, la integración CI/CD asegura que cualquier modificación en el código sea sometida automáticamente a un proceso de verificación objetivo y reproducible. Esto reduce de manera significativa la probabilidad de introducir errores en el entorno de producción y fortalece la confiabilidad del sistema durante su ciclo de vida.

De este modo, la combinación de GitHub, GitHub Actions, Selenium y el servidor de Hostinger constituye una infraestructura coherente para el aseguramiento continuo de la calidad, alineada con los principios de desarrollo moderno orientado a la automatización, la trazabilidad y la eficiencia operativa.

Se muestra el registro automatizado del envío de resultados correspondiente a cada una de las pruebas ejecutadas durante el proceso de validación. Estos correos, generados por el sistema tras cada ejecución, permiten documentar de forma estructurada los hallazgos obtenidos, incluyendo tanto los casos exitosos como los fallos detectados.

Este mecanismo de notificación contribuye a la trazabilidad del proceso de aseguramiento de calidad, facilitando la revisión posterior por parte de auditores, desarrolladores o responsables técnicos.

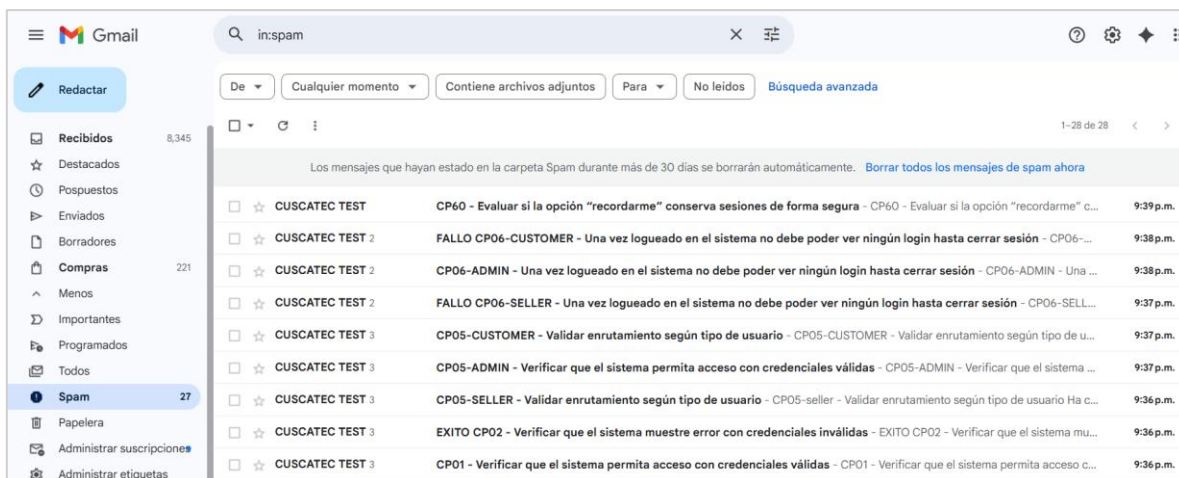


Ilustración 13. Envío de Resultados de cada una de las pruebas.

La implementación del flujo de CI/CD se ha centrado exclusivamente en las pruebas automatizadas desarrolladas con Selenium, dado que ofrecen una integración más directa con GitHub Actions y permiten validar escenarios críticos de interacción en navegadores reales. Esta decisión responde a la necesidad de contar con un pipeline inicial estable y reproducible, que garantice trazabilidad y resultados auditables en cada ejecución.

Las pruebas realizadas con Laravel Dusk y PHPUnit continúan siendo parte del ecosistema de validación del proyecto, pero fuera del pipeline principal, en busca a una futura incorporación progresiva.

6.16. Entorno de prueba

Para garantizar la calidad y confiabilidad del sistema, así como la seguridad de los datos y la operación en producción, se estableció un entorno de pruebas independiente que replica de manera controlada las condiciones del sistema productivo. Esta decisión responde a buenas prácticas en aseguramiento de calidad, evitando cualquier riesgo de afectar directamente a usuarios finales, procesos activos o información sensible.

La configuración del entorno permite validar funcionalidades críticas en condiciones equivalentes, manteniendo un aislamiento completo del sistema en operación. Esta estrategia asegura que los resultados obtenidos durante las pruebas sean confiables, reproducibles y trazables, sin comprometer la estabilidad del entorno productivo ni generar riesgos operativos innecesarios.

1. Requerimientos de versiones

- Framework Laravel (PHP) V^8.2.2
- Composer V^2.8.9
- Gestor de base de datos MySQL V^8.0
- Windows 11

2. Requisitos previos

Antes de configurar el entorno de pruebas, se deben cumplir los siguientes requerimientos:

- Cuenta de hosting en Hostinger con acceso administrativo al servidor.
- Servidor web LiteSpeed disponible para desplegar la aplicación.
- Acceso a la base de datos MySQL.
- Archivos del sistema (back-end en Laravel, front-end Blade/JS/HTML)
- Datos de prueba de la base de datos, con información sensible debidamente ofuscada para preservar la confidencialidad.

3. Pasos de configuración

- **Preparar el servidor en Hostinger**
 - Crear un dominio o subdominio exclusivo para pruebas.
- **Desplegar la aplicación**
 - Clonar los archivos del proyecto (back-end y front-end) al entorno de pruebas.
 - Configurar las rutas, controladores y vistas para que apunten a la base de datos
 - Instalar dependencias de Laravel mediante Composer.
- **Configurar la base de datos**
 - Crear un esquema independiente en MySQL.
 - Importar los datos replicados desde producción, asegurando ofuscación de información sensible.
 - Configurar usuarios y permisos de la base de datos

4. Ajustes del entorno de Laravel

- Configurar el archivo .env para apuntar a la base de datos.

5. Validar el entorno de pruebas

- Verificar que el front-end cargue correctamente y que todas las funcionalidades del back-end respondan como en producción.
- Probar conexión a la base de datos y endpoints críticos.
- Ejecutar un conjunto básico de pruebas para asegurar que el entorno está listo para pruebas funcionales y automatizadas.

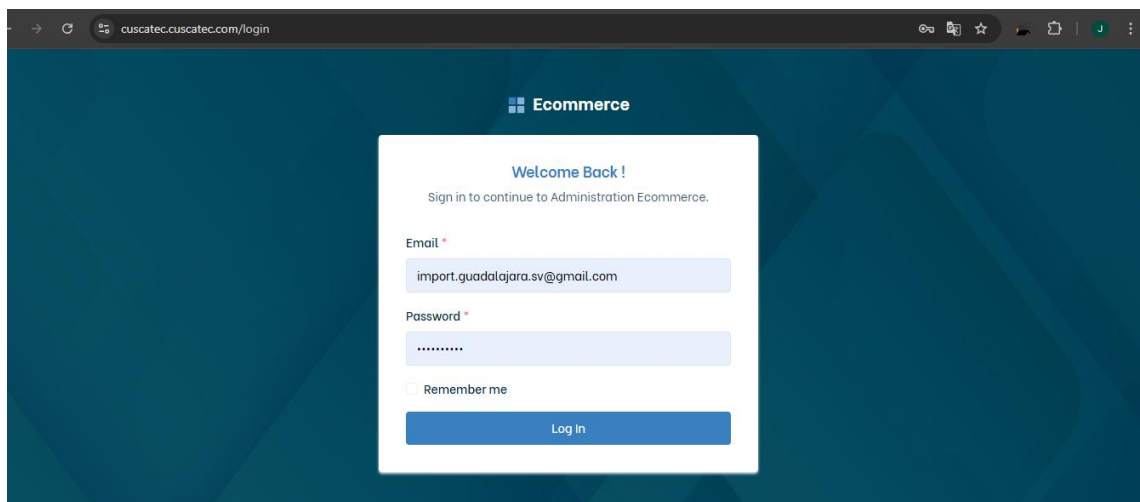


Ilustración 14. Interfaz de autenticación del sistema: Importaciones Guadalajara.

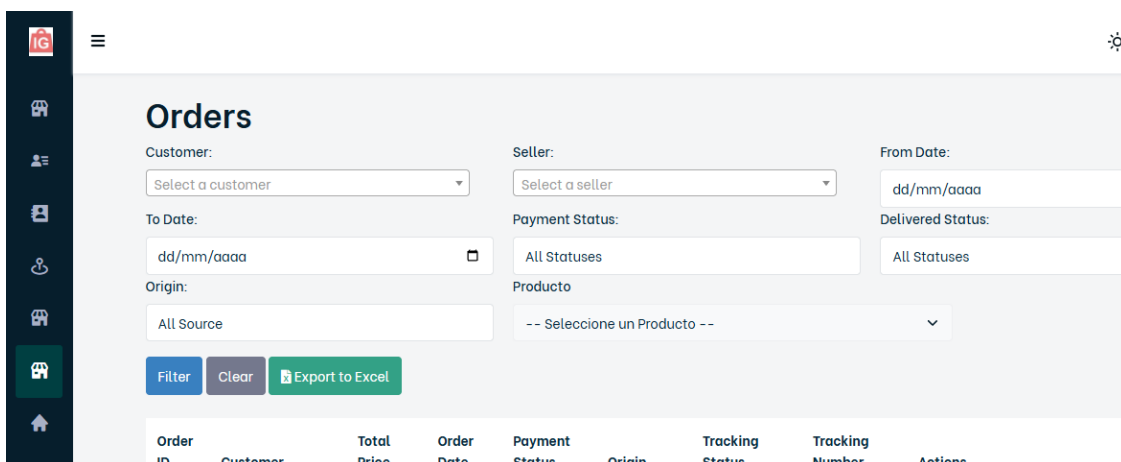


Ilustración 15. Interfaz de ordenes: Importaciones Guadalajara.

6.17. Registro de ejecución de pruebas

El registro de ejecución de pruebas documenta todos los casos de prueba ejecutados durante el proceso de aseguramiento de calidad. Este registro permite mantener trazabilidad completa de los requerimientos, evidenciar la cobertura de pruebas y facilita tanto la identificación de errores como la mejora continua del sistema.

Cada caso de prueba se documenta con una plantilla estandarizada que incluye información sobre el caso, su ejecución, resultados y evidencia. Esto garantiza que sea entendible el estado de las pruebas y los resultados obtenidos en cualquier momento.

- **Campos de registro**

- a) Identificador del caso (ID):** Código único asignado a cada caso de prueba para facilitar su seguimiento y referencia.
- b) Ejecutado por:** Nombre del QA que realiza la prueba.
- c) Nombre del caso:** Descripción breve y clara del caso de prueba.
- d) Prioridad:** Nivel de importancia de la prueba (Alta, Media, Baja).
- e) Sprint:** Iteración del proyecto en la que se ejecuta la prueba.
- f) Estado:** Resultado de la ejecución (Aprobado, Rechazado, Pendiente, Con hallazgos).
- g) Propósito del caso:** Objetivo de la prueba y funcionalidad que se valida.
- h) Módulo/Componente:** Área del sistema donde se realiza la prueba.
- i) Condiciones de prueba:** Requisitos previos para ejecutar la prueba correctamente, incluyendo usuarios, datos y configuraciones necesarias.
- j) Procedimiento de prueba:** Pasos detallados que se deben seguir durante la ejecución del caso.
- k) Resultados esperados:** Comportamiento esperado si la prueba se ejecuta correctamente.
- l) Resultado obtenido:** Resultado real observado tras la ejecución de la prueba.
- m) Observaciones/Recomendaciones:** Comentarios adicionales sobre la prueba, incidencias detectadas o aclaraciones.
- n) Evidencia:** Capturas de pantalla, registros, logs o cualquier soporte que valide la ejecución de la prueba.

6.18. Entregables de pruebas automatizada y no automatizadas

Una vez concluida la ejecución de las pruebas y recopilados los resultados correspondientes, se establecen los entregables que documentan las actividades realizadas y permiten evaluar la efectividad del proceso. Estos productos permiten garantizar la trazabilidad del proceso, la transparencia en la ejecución y la evaluación objetiva de la calidad alcanzada. En el presente proyecto, los entregables se diferencian según la naturaleza de las pruebas ejecutadas, ya sean automatizadas o no automatizadas.

a) Pruebas no automatizadas

Los entregables correspondientes a las pruebas manuales están orientados a documentar de forma estructurada los resultados obtenidos durante la ejecución de los casos de prueba definidos, entre los principales se incluyen:

- Plan de pruebas documentado, con el alcance, objetivos y criterios de aceptación.
- Casos de prueba manuales, detallando los pasos, datos de entrada y resultados esperados.
- Bitácoras o registros de ejecución con los resultados reales observados.
- Evidencias gráficas (capturas de pantalla, reportes de comportamiento del sistema).
- Registro de incidencias detectadas.
- Informe de resultados de pruebas manuales, consolidando hallazgos, métricas y conclusiones.

b) Pruebas automatizadas

En las pruebas automatizadas, los entregables son generados a partir de las herramientas empleadas para la ejecución de scripts y la obtención de resultados automáticos. Estos entregables permiten la repetición sistemática de los casos de prueba y la comparación objetiva entre distintas ejecuciones. Se consideran los siguientes:

- Scripts o suites de pruebas desarrolladas en herramientas de automatización.
- Archivos de configuración y datos de entrada asociados a los scripts.
- Reportes de ejecución generados automáticamente.
- Evidencias de ejecución (capturas de pantalla, registros de logs o reportes de fallos).
- Informe consolidado de resultados de pruebas automatizadas, con métricas de cobertura, tiempo de ejecución y defectos encontrados.

Ambos tipos de entregables constituyen la evidencia documental y técnica que respalda el cumplimiento de los objetivos de calidad establecidos en el plan de pruebas.

6.19. Matriz de pruebas

Los casos de prueba son elementos importantes en el proceso de aseguramiento de calidad. Cada uno está diseñado para validar el correcto funcionamiento de una funcionalidad específica, asegurando que el sistema cumpla con los requisitos establecidos. Esta sección presenta una matriz con los casos de prueba que cubren los módulos del sistema a evaluar.

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|----------|--|--|---|-----------|
| CP01 | Login | Acceso exitoso | Verificar que el sistema permita acceso con credenciales válidas | Redirección al dashboard correspondiente | Alta |
| CP02 | Login | Manejo de credenciales incorrectas | Verificar que el sistema muestre error con credenciales inválidas | Mensaje de error "Email o contraseña incorrectos" visible | Alta |
| CP03 | Login | Campos obligatorios vacíos del login | Validar comportamiento con campos vacíos | Validaciones frontend bloquean el envío | Alta |
| CP04 | Login | Bloqueo por intentos fallidos | Verifica que la cuenta sea bloqueada luego de varios intentos fallidos | Cuenta bloqueada con mensaje visible | Alta |
| CP05 | Login | Redirección por rol | Validar enrutamiento según tipo de usuario | Redirección al URL según rol | Media |
| CP06 | Login | Redirección por logueo | Una vez logueado en el sistema no debe poder ver ningún login hasta cerrar sesión | Redirección a inicio según rol | Media |
| CP07 | Products | Crear producto válido | Validar que el sistema permita registrar un producto con todos los campos obligatorios | Producto guardado y aparece en tabla | Alta |
| CP08 | Products | Agregar Imagen en formato JPG, PNG o ninguna | Validar formatos permitidos en carga de imagen | Imagen aceptada o sin imagen, con formato valido | Media |
| CP09 | Products | Editar producto existente | Validar edición de producto previamente registrado | Cambios reflejados en tabla | Alta |

Tabla 35. Diseño de casos de prueba (Parte I).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|----------|--|--|---|-----------|
| CP10 | Products | Búsqueda de productos | Evaluar búsqueda con texto parcial o incompleto | Resultados relacionados visibles | Media |
| CP11 | Products | Envío de producto a estado "Trashed" | Verificar que al seleccionar "Move to Trash" el producto se desactive lógicamente | El producto seleccionado es movido a estado "Trashed". El mensaje de confirmación de acción se muestra y la operación se completa sin errores. | Media |
| CP12 | Products | Eliminación en cascada | Permitir desactivar varios productos a la vez | Todos los productos reflejan el cambio en estado | Media |
| CP13 | Products | Restauración de producto individual | Verificar que producto "trashed" se restaure individualmente | Producto reaparece en listado activo | Baja |
| CP14 | Products | Restauración masiva de productos | Validar restauración masiva desde pestaña "Trashed" | Todos los productos seleccionados reaparecen activos | Baja |
| CP15 | Products | Guardar con campos obligatorios vacíos | Validar que el formulario no envíe datos incompletos | Formulario bloqueado para enviar | Alta |
| CP16 | Products | Manejo de valores numéricos negativos | Validar que el sistema rechace valores numéricos negativos en campos como Stock o Precio | El sistema debe mostrar un mensaje de error indicando que los valores deben ser mayores a cero. | Alta |
| CP17 | Orders | Filtrar órdenes por rango de fechas | Validar que filtros por fechas funcionen correctamente | Resultados filtrados correctamente | Media |
| CP18 | Orders | Visualización de detalle de orden | Verificar que detalle de orden muestre información completa | La página de detalle de la orden se carga correctamente. Se visualiza el nombre del cliente y el vendedor. El monto total de la orden coincide con lo esperado. El estado de seguimiento es visible | Alta |

Tabla 36. Diseño de casos de prueba (Parte II).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|--------|--|---|--|-----------|
| CP19 | Orders | Actualización de envío y número de seguimiento | Validar modificación el estado de envío de una orden y el número de envío | El sistema edito correctamente la orden existente El total calculado correctamente actualizado. Número y estado de envío actualizado en la orden | Alta |
| CP20 | Orders | Modificación de detalles de orden | Editar cantidad, precios y descuentos | El sistema edito correctamente la orden existente El total calculado correctamente actualizado. Número y estado de envío actualizado en la orden | Alta |
| CP21 | Orders | Actualización de estado de pago | Cambiar tipo de pago asociado a una orden | El estado del método de pago se actualiza correctamente en la base de datos y se refleja en el sistema | Alta |
| CP22 | Orders | Exportación de órdenes a Excel | Validar generación de archivo Excel con datos filtrados | Excel descargado con encabezados. Los registros deben coincidir con los filtros aplicados. | Media |
| CP23 | Orders | Impresión de detalle de orden | Verificar generación correcta de PDF con detalle de orden | El PDF debe mostrar el detalle completo de la orden, con formato correcto, productos listados, totales calculados y sin errores de visualización. | Media |
| CP24 | Orders | Generación de reporte de ventas por producto | Generar reporte con productos más vendidos filtrados por fecha | El sistema debe aplicar los filtros correctamente y generar un archivo Excel con las órdenes correspondientes. | Media |

Tabla 37. Diseño de casos de prueba (Parte III).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|-----------|---------------------------------------|--|--|-----------|
| CP25 | Orders | Eliminación de orden | Verificar eliminación lógica de orden | La orden del cliente fue eliminada correctamente. Eliminación confirmada en el sistema. | Alta |
| CP26 | Inventory | Registro exitoso de bodega | Registrar nueva bodega con datos completos | La bodega se crea correctamente en la base de datos, el registro contiene: nombre, dirección, estado. | Alta |
| CP27 | Inventory | Filtro por bodega en inventario | Mostrar productos según bodega seleccionada | Productos mostrados corresponden a bodega seleccionada | Media |
| CP28 | Inventory | Exportación de inventario a Excel | Generar archivo Excel con reporte visible | Archivo Excel generado con datos correctos | Baja |
| CP29 | Inventory | Consulta de historial Kardex mensual | Mostrar historial de movimientos filtrado por producto y fecha | Historial correcto y cálculos precisos | Media |
| CP30 | Inventory | Verificación del saldo por movimiento | Validar cálculo de saldo en Kardex | Los saldos se reflejan correctamente | Alta |
| CP31 | Inventory | Carga masiva de stock desde Excel | Validar carga correcta desde archivo Excel | El archivo se carga correctamente. Mostrar mensaje de confirmación. | Alta |
| CP32 | Customer | Registrar nuevo cliente válido | Crear cliente con todos los campos requeridos | El sistema registrara un nuevo cliente al sistema de forma exitosa | Alta |
| CP33 | Customer | Editar cliente existente | Modificar información de contacto del cliente | Campos phone, address, email, y password actualizados. Correo enviado a nuevo email | Alta |
| CP34 | Customer | Desactivar (eliminar) cliente | Cambiar estado de cliente a "Inactivo" | El cliente seleccionado debe ser eliminado/ desactivado y ya no debe aparecer en la lista de clientes. | Media |

Tabla 38. Diseño de casos de prueba (Parte IV).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|----------|--|--|---|-----------|
| CP35 | Customer | Búsqueda de cliente por nombre | Filtrar clientes por nombre | Cada cliente buscado debe aparecer en los resultados, y el tiempo de respuesta debe ser menor a 2 segundos. | Media |
| CP36 | Customer | Envío de credenciales por email | Validar envío automático de credenciales tras registro | El correo debe enviarse exitosamente, contener los datos correctos, mostrarse con diseño responsivo. | Alta |
| CP37 | Customer | Generación automática de contraseña | Generar contraseña segura y automática | El campo password se autocompleta con una contraseña de al menos 6 caracteres, incluyendo letras, números y símbolos. | Alta |
| CP38 | Customer | Búsqueda de cliente según Estado y combinada | Filtrar clientes según su Estado geográfica en el país y búsqueda combinada (cliente + Estado) | El sistema debe mostrar clientes por Estado. Las combinaciones válidas de nombre + Estado deben devolver resultados. Las búsquedas sin datos deben excluirse éticamente sin error | Baja |
| CP39 | Users | Creación de usuario Administrador o Seller | Registrar usuario con rol válido | Usuario creado con rol asignado | Alta |
| CP40 | Users | Validación de email único | No permitir crear usuario con email duplicado | Error de validación y no creación de usuario | Alta |
| CP41 | Users | Generación de contraseña de usuario | Validar complejidad y seguridad de contraseña autogenerada | Campo de contraseña inicialmente vacío. Contraseña generada correctamente al hacer clic en el botón y que cumpla con los criterios mínimos de complejidad | Alta |

Tabla 39. Diseño de casos de prueba (Parte V).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|-----------|--------------------------------------|---|--|-----------|
| CP42 | Users | Edición de rol de usuario | Cambiar rol de usuario solo por Admin | Rol actualizado solo si es Admin | Alta |
| CP43 | Users | Validación de opciones del campo Rol | El dropdown solo debe mostrar roles válidos | Opciones: solo "Administrador" | Alta |
| CP44 | Users | Validación de Campos Vacíos | Verificar que el sistema no permita crear un usuario si los campos obligatorios están vacíos. | El sistema debe mostrar un mensaje de error para cada campo vacío (ej. "El nombre es obligatorio"). El formulario debe permanecer en la misma página, sin redirigir ni crear un usuario | Alta |
| CP45 | Seguridad | Inyección código SQL | Validar que el sistema rechaza intentos de inyección SQL en el formulario de login | El sistema debe rechazar entradas maliciosas. No debe ejecutar consultas SQL no autorizadas. Debe mostrar mensajes de error genéricos o validar el formato antes de enviar. No debe reflejar contenido sin codificación. | Alta |
| CP46 | Seguridad | Acceder a vista Admin | Bloquear acceso a endpoints Admin para usuarios básicos | Acceso denegado y log de seguridad generado | Alta |
| CP47 | Seguridad | Headers HTTP | Verificar seguridad de headers HTTP | Headers definidos desde el código deben estar presentes y configurados correctamente. Headers inseguros o ausentes deben ser identificados para corrección. | Alta |

Tabla 40. Diseño de casos de prueba (Parte VI).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|--------------|-----------------------------------|--|---|-----------|
| CP48 | Rendimiento | 100 usuarios concurrentes | Simular 100 usuarios haciendo búsquedas simultáneas | Respuesta estable sin caídas | Alta |
| CP49 | Rendimiento | Saturación del sistema | Simular saturación con 500+ usuarios | Sistema se recupera automáticamente | Alta |
| CP50 | Concurrencia | Edición del mismo producto | Validar edición simultánea sin conflictos | Transacciones atómicas, sin conflictos | Alta |
| CP51 | Seguridad | Gestión de sesión – time-out | Validar que la sesión se cierre tras detectar inactividad | Sesión expira automáticamente y se redirige al login | Media |
| CP52 | Seguridad | Restricción de acceso por rol | Verificar que usuarios no autorizados no puedan acceder a funciones restringidas | Acceso denegado y mensaje adecuado, sin exposición de información | Alta |
| CP53 | UI/UX | Compatibilidad con navegadores | Validar que el sistema funciona correctamente en los diferentes navegadores en sus versiones recientes | Login y dashboard cargado. Formulario abierto y cargado sin errores. Recursos (CSS, JS, imágenes) cargados. | Media |
| CP54 | Seguridad | Revisión de código seguro | Confirmar que se aplican prácticas seguras de programación, minimizando vulnerabilidades comunes | El reporte final no muestra vulnerabilidades críticas y el código cumple con los estándares de seguridad establecidos | Alta |
| CP55 | Seguridad | Validación de certificado SSL/TLS | Verificar que el certificado SSL/TLS del sitio web sea válido, esté correctamente configurado y utilice protocolos seguros | El certificado es válido, está vigente, la conexión utiliza protocolos y cifrados seguros sin advertencias | Alta |
| CP56 | Seguridad | Transporte seguro de credenciales | Verificar que el envío de usuario y contraseña se haga bajo canal cifrado (HTTPS/TLS) | Tráfico cifrado, sin envío en texto plano | Alta |

Tabla 41. Diseño de casos de prueba (Parte VII).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|-----------|---|---|---|-----------|
| CP57 | Seguridad | Protección contra fuerza bruta | Verificar bloqueo temporal y CAPTCHA tras múltiples intentos fallidos. | El sistema debe activar mecanismos de defensa ante múltiples intentos fallidos: bloqueo temporal, CAPTCHA, rate limit, etc. | Alta |
| CP58 | Seguridad | Detección de credenciales predeterminadas/débiles | Intentar iniciar sesión con credenciales comunes o vacías. | Acceso denegado, mensaje genérico | Alta |
| CP59 | Seguridad | Enumeración de cuentas | Verificar si los mensajes de error revelan si el usuario existe. | Mensaje genérico ante credenciales inválidas que no indique existencia del usuario. Códigos HTTP consistentes para fallos de autenticación No diferencias observables en headers/cookies que permitan inferir existencia. | Alta |
| CP60 | Seguridad | Validación de “Remember Me” | Evaluar si la opción “recordarme” conserva sesiones de forma segura | Sesión activa, cookies seguras | Media |
| CP61 | Seguridad | Gestión segura de sesiones | Verificar atributos de cookies, expiración y control de sesiones concurrentes | Cookies seguras, sesión expirada, control de concurrencia | Alta |
| CP62 | Seguridad | Bypass de autenticación y MFA | Intentar saltarse el login o manipular parámetros. Activar MFA si disponible | MFA solicitado correctamente, login no evadido | Alta |
| CP63 | Seguridad | Respuestas ante errores y mensajes | Evaluar si los mensajes de error revelan información sensible. | 1. Provocar errores (login, formularios). 2. Analizar mensajes | Media |
| CP64 | UI/UX | Pruebas de UI/UX y resiliencia | Verificar funcionamiento de CAPTCHA, enlaces, validaciones, y responsive. | UI correcta, mensajes claros, enlaces funcionales, responsive | Media |

Tabla 42. Diseño de casos de prueba (Parte VIII).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|----------|--|---|--|-----------|
| CP65 | Purchase | Vista inicial de categorías | Validar que al iniciar sesión el usuario sea redirigido a la pantalla de categorías | Se muestra pantalla de categorías | Alta |
| CP66 | Purchase | Visualización de productos por categoría | Verificar que al seleccionar una categoría se muestren solo los productos asociados | Productos listados de esa categoría | Alta |
| CP67 | Purchase | Gestión de productos en carrito | Validar agregar y remover productos desde la categoría seleccionada | Producto agregado o removido del carrito | Alta |
| CP68 | Purchase | Cálculo de totales | Validar que se muestren los cálculos correctos (subtotal, descuentos, shipping, total final) | Totales mostrados correctamente | Alta |
| CP69 | Purchase | Actualización de estado de pago (Seller) | Validar que desde la vista Seller se pueda cambiar estado de pago y se refleje en facturación | Estado actualizado en BD y visible en facturación | Alta |
| CP70 | Category | Validación de campos obligatorios | Verificar que el sistema no permita crear categoría sin completar Name y Description | Si los campos obligatorios están vacíos, se muestran mensajes de error y el formulario permanece abierto. Si los campos obligatorios se completan correctamente, el registro se guarda y el sistema redirige a la lista de categorías. | Alta |
| CP71 | Category | Validación de carga de imagen | Verificar que solo se puedan subir imágenes válidas (JPG, PNG, tamaño permitido) | Categoría guardada correctamente junto con su imagen. Formatos válidos: mensaje de éxito. Formatos no válidos: mensaje de error. | Media |

Tabla 43. Diseño de casos de prueba (Parte IX).

| ID | Módulo | Título del caso | Descripción | Resultado esperado | Prioridad |
|------|----------|--|--|--|-----------|
| CP72 | Category | Registro exitoso de categoría | Validar que al crear categoría se registre correctamente y aparezca en la lista | La categoría se guarda en la base de datos. La categoría aparece en la lista de categorías. El campo name, description e imagen se muestran y guardan correctamente. | Alta |
| CP73 | Category | Retorno a lista desde Create Category | Verificar que al presionar "Back to List" se regrese a la vista lista sin pérdida de datos | El sistema regresa a la lista de categorías. Los datos existentes en la lista se muestran correctamente. No se presentan mensajes de error. | Baja |
| CP74 | Category | Eliminar categoría | Remover una categoría de la lista | La categoría seleccionada se elimina correctamente. La lista de categorías se actualiza y no muestra la categoría eliminada. | Media |
| CP75 | Category | Editar una categoría | Validar la edición de una categoría existente | La categoría se actualiza correctamente y se muestran los cambios en la lista, incluyendo la imagen. | Alta |
| CP76 | Orders | Agregar un producto a la orden | Agregar un producto a una orden existente | Todos los pasos ejecutados correctamente. Producto agregado, totales recalculados. El sistema no permite agregar producto, si la orden está vacía. | Alta |
| CP77 | Orders | Remover producto /productos de una orden | Remover uno o más productos a una orden existente | Productos seleccionados son eliminados. El total se actualiza correctamente. No se generan errores y la orden permanece activa y editable. | Media |

Tabla 44. Diseño de casos de prueba (Parte X).

6.20. Test Readiness Review (TRR)

El Test Readiness Review es una revisión formal previa al inicio de la ejecución de pruebas, con el objetivo principal de asegurar que todos los elementos técnicos, metodológicos y organizativos estén debidamente preparados. Esta evaluación permite verificar que el entorno de pruebas se encuentre operativo, que las herramientas y datos requeridos estén disponibles, y que los casos de prueba hayan sido correctamente documentados, revisados y validados.

En el contexto del presente proyecto, la TRR se realizó antes de iniciar las pruebas automatizadas, con el fin de garantizar que no existan bloqueos, inconsistencias o dependencias pendientes que puedan comprometer la ejecución o afectar la calidad de los resultados. Esta revisión actúa como un punto de control que valida la estabilidad del entorno y la preparación del equipo, asegurando que las condiciones mínimas necesarias para una ejecución efectiva estén cumplidas.

La ejecución de esta revisión no solo permite reducir riesgos operativos, sino que también fortalece la trazabilidad del proceso de validación. Al documentar el cumplimiento de cada criterio, se establece una base sólida para la ejecución controlada de las pruebas, contribuyendo a la calidad del software y a la defensa técnica del proyecto en auditorías o revisiones académicas.

La siguiente tabla presenta el formato propuesto para la revisión de preparación de pruebas. Este documento servirá como instrumento de verificación y control, permitiendo registrar el cumplimiento de los criterios establecidos antes de dar inicio formal al proceso de validación

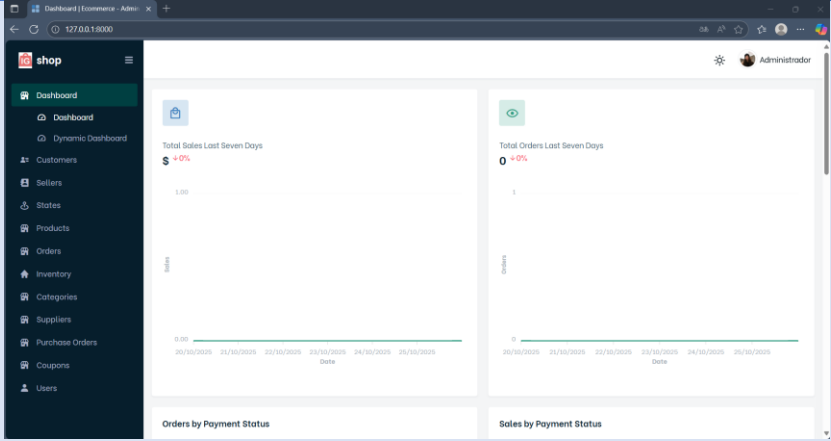
| Elemento | Descripción | Estado | Comentario |
|---|---|--------|---|
| Entorno de prueba instalado y funcional | Servidor local configurado, base de datos conectada y módulos accesibles. | Listo |  |
| Herramientas disponibles | PHPUnit, Laravel Dusk, JMeter, Selenium, entre otras, | Listo | Se validó su ejecución en el entorno. |
| Casos de prueba documentados y validados | Casos revisados y aprobados. | Listo | La matriz de prueba fue aprobada por el desarrollador del sistema. |
| Datos de prueba creados. | Datos cargados en la base. | Listo | |
| Roles definidos | Equipo de QA establecido. | Listo | Para este proyecto los 3 integrantes del equipo desempeñan rol de QA. |
| Criterios de entrada cumplidos | Se cumplen los requisitos de inicio de pruebas. | Listo | |

Tabla 45. Test Readiness Review

6.21. Estimaciones

A continuación, se estima el esfuerzo (horas/hombre), duración calendarizada y recursos necesarios para la ejecución del plan de pruebas (manuales y automatizadas), utilizando un enfoque probabilístico (método PERT) para promediar incertidumbres en las estimaciones.

El método PERT (Program Evaluation and Review Technique) se usa para producir estimaciones más realistas cuando existe variabilidad en la duración de tareas. Para cada actividad se define:

- Tiempo optimista (O)
- Más probable (M)
- Pesimista (P)

La duración esperada se calcula con la siguiente fórmula:

$$E = \frac{O + 4M + P}{6}$$

Ecuación 1. Método de PERT.

Este enfoque pondera el escenario más probable, pero incorpora los extremos, reduciendo el sesgo optimista y proporcionando una base cuantitativa para reservar contingencia. En función de ello, se establecieron los datos base del proyecto, los cuales permiten aplicar el modelo de estimación de manera consistente y verificable.

- **Número de casos:**
 - a) 64 casos automatizables
 - b) 13 casos no automatizables
- **Herramientas y entorno:** PHPUnit, Laravel Dusk, Selenium, JMeter, OWASP ZAP, SonarQube; gestión en Azure DevOps y evidencias en OneDrive/Excel.
- **Equipo:** 3 integrantes con roles combinados, participando todos como QA.

6.21.1. Supuestos de estimación (por caso) – valores PERT

1. Caso manual sencillo (validaciones simples):

O = 0.5 h

M = 0.75 h

P = 1.0 h

Lo que nos da E = 0.75h.

2. Caso manual complejo (validaciones que requieren integración, verificaciones en BD o revisiones intensas):

O = 1.0 h

M = 1.5 h

P = 2.0 h

Lo que nos da E = 1.50 h.

3. Script de automatización:

O = 2.0 h

M = 3.0 h

P = 5.0 h

Lo que nos da E \approx 3.17 h.

4. Mantenimiento o ajuste de script existente:

O = 0.25 h

M = 0.75 h

P = 1.5 h

Lo que nos da E \approx 0.79 h.

5. Registro de evidencia por caso:

O = 0.15 h

M = 0.25 h

P = 0.5 h

Lo que nos da E \approx 0.28 h.

Estos supuestos se definieron considerando la complejidad habitual presente en aplicaciones web desarrolladas con frameworks MVC como Laravel, así como en proyectos de automatización de pruebas funcionales para plataformas tipo e-commerce.

Diversos estándares de la industria establecen que las estimaciones deben basarse en la naturaleza del sistema, la complejidad de los flujos y el nivel de automatización previsto (IEEE 29119-2, 2013).

6.21.2. Cálculo aplicado a la matriz de pruebas

1. Distribución de casos:

- No automatizables: 13 casos
- Automatizables: 64 casos, supuesto: 55 requieren desarrollo de script desde cero y 14 se consideran como reutilizables.

2. Cálculo por tipo

- Manual (13 casos x 0.75 h) = 9.75 h.
- Automatización (desde 0) (50 casos × 3.17 h) = 158.50 h.
- Automatización (reutilización) (14 casos × 0.75 h) = 10.50 h.
- Registro y consolidación de evidencia (77 casos × 0.28 h) = 21.56 h.
- Total, de horas estimadas = 200.31 horas.
- Reserva de contingencia: se recomienda 15% por incertidumbres técnicas, correcciones y retests, por lo que $193.46 \times 0.15 = 30.05$ h.
- Total, con contingencia: = 230.36 h.

6.21.3. Duración calendarizada

Tomando en cuenta que se tomaran 4 horas al día por tester, siendo el equipo de 3 testers.

- Horas disponibles por día: 3 testers x 4 h/día = 12 h/día.
- Días necesarios = Total de contingencia / 12 horas por día
- $222.48 / 12 = 18.54$ días laborales.

Plan práctico: 19 días laborales (aprox. 4 semanas si consideras días no continuos, revisiones con el desarrollador y sesiones de triaje). Esto incluye desarrollo de scripts, ejecución inicial, re-ejecuciones tras correcciones y consolidación de evidencia.

6.21.4. Entregables asociados a las estimaciones

- Tabla detallada de casos vs horas.
- Cronograma por sprint de pruebas.
- Backup de evidencia (OneDrive/Sharepoint).
- Registro de horas ejecutadas y desviaciones.

6.22. Métricas de pruebas

Se presenta a continuación las métricas de prueba que permitan evaluar de manera objetiva el avance y la calidad del proceso de verificación de un sistema con el propósito es ofrecer información clara y cuantificable

a) Cobertura de pruebas

La cobertura de pruebas mide el grado de avance en la ejecución de los casos planificados. Se calcula dividiendo el número de casos de prueba efectivamente ejecutados entre el total de casos de prueba planificados.

$$\text{Cobertura de pruebas} = \frac{\text{Total de casos de prueba ejecutados}}{\text{Total de casos de prueba planificados}} \times 100$$

Ecuación 2. Porcentaje de casos de pruebas ejecutados.

b) Porcentaje de casos de prueba aprobados

Esta métrica muestra la proporción de casos de prueba que han sido ejecutados con resultado exitoso (aprobado). Un valor alto indica que el sistema cumple con lo esperado y funciona correctamente; mientras que un valor bajo señala la existencia de fallos que deben ser corregidos antes de avanzar.

$$\text{Casos de casos de prueba aprobados} = \frac{\text{Total de casos aprobados}}{\text{Total de casos ejecutados}} \times 100$$

Ecuación 3. Porcentaje de casos de pruebas aprobados.

c) Porcentaje de casos de prueba fallidos

Esta métrica refleja la proporción de casos de prueba ejecutados que no cumplen con los criterios esperados. Un valor elevado indica que el producto presenta defectos significativos y requiere correcciones urgentes; mientras que un valor bajo sugiere que el sistema es más estable y confiable.

$$\text{Casos de casos de prueba fallidos} = \frac{\text{Total de casos de prueba fallidos}}{\text{Total de casos de prueba ejecutados}} \times 100$$

Ecuación 4. Porcentaje de casos de pruebas fallidos.

d) Porcentaje de casos de prueba con hallazgos

Esta métrica refleja la proporción de casos ejecutados que generaron algún tipo de hallazgo o estado intermedio. Un valor alto puede indicar que el sistema presenta situaciones

no concluyentes que requieren análisis adicional, mientras que un valor bajo sugiere que la mayoría de los casos se clasifican claramente como aprobados o fallidos.

$$\text{Casos de casos de prueba con hallazgos} = \frac{\text{Total de casos de prueba con hallazgos}}{\text{Total de casos ejecutados}} \times 100$$

Ecuación 5. Porcentaje de casos de pruebas fallidos.

e) Tasa de defectos por módulo

Esta métrica permite identificar qué módulos del sistema concentran mayor cantidad de defectos en relación con el total encontrado. Un valor alto indica que el módulo es crítico y requiere mayor atención en pruebas y correcciones; mientras que un valor bajo sugiere que el módulo es más estable o menos problemático.

$$\text{Total de defectos por módulo} = \frac{\text{Número de defectos en módulo}}{\text{Total de defectos}} \times 100$$

Ecuación 6. Porcentaje de defectos encontrados por módulo.

6.23. Criterios de entrada y salida

Los criterios de entrada y salida se establecieron para garantizar que el proceso de pruebas se desarrollara bajo condiciones controladas y concluyera con resultados verificables

- **Criterios de entrada:** Los criterios de entrada establecen las condiciones mínimas necesarias para iniciar las pruebas, con el objetivo de que se inicien de manera ordenada y confiable.

| Criterio | Validación realizada |
|---------------------------|--|
| Entorno preparado | Se comprobó que el entorno estuviera correctamente configurado y en funcionamiento, asegurando estabilidad durante la ejecución. |
| Casos de prueba listos | Se verificó que los casos estuvieran diseñados y revisados, evitando inconsistencias en la validación. |
| Herramientas operativas | Se confirmó que las herramientas requeridas estuvieran instaladas y funcionando, previniendo retrasos o fallos por falta de recursos técnicos. |
| Accesos y compatibilidad | Se validó que el equipo tuviera acceso al entorno y que los navegadores definidos para compatibilidad estuvieran disponibles y probados, evitando riesgos de incompatibilidad. |
| Roles y responsabilidades | Se definieron y comunicaron roles y responsabilidades dentro del equipo de pruebas, garantizando claridad en la ejecución y trazabilidad. |

Tabla 46. Criterios de entrada

- **Criterios de salida:** Los criterios de salida definen cuándo las pruebas pueden considerarse concluidas, lo que implica haber ejecutado los casos planificados, documentado los defectos y alcanzado la cobertura definida para esta fase

| Categoría | Criterio de salida | Descripción |
|------------------------|--------------------------------|---|
| Ejecución de pruebas | Cobertura total de ejecución | Deben ejecutarse el 100% de los casos de prueba diseñados. |
| Resultados | Validación de resultados | Los resultados obtenidos deben coincidir con los resultados esperados sin desviaciones. |
| | Registro de fallos | Todo caso fallido debe tener un defecto documentado con evidencia adjunta. |
| Casos de prueba | Casos de prueba alto | Los casos de alta prioridad se ejecutaron y validaron con evidencia registrada. |
| | Casos de prueba medios y bajos | Los casos de prioridad media y baja se ejecutaron y documentaron. |
| Pruebas no funcionales | Rendimiento | Tiempos de respuesta cortas |
| | Seguridad | No deben existir vulnerabilidades críticas o altas detectadas por OWASP ZAP. |
| | Compatibilidad | Funcionamiento correcto en los navegadores definidos. |
| Automatización | Ejecución estable | Las suites automatizadas deben correr sin errores de script. |
| | Cobertura automatizada | Deben ejecutarse los casos definidos como automatizables en la matriz. |
| | Reportes | Se generó la evidencia disponible según las capacidades de cada herramienta utilizada |
| Documentación | Evidencia completa | Se debe entregar evidencia de todos los casos ejecutados. |
| | Reportes técnicos | Presentación de los resultados obtenidos. |
| | Matriz de trazabilidad | Debe estar actualizada con resultados finales. |
| Aprobación formal | Validación del equipo | El equipo de QA verifica el cumplimiento de todos los criterios. |
| | Aprobación del asesor | El docente asesor aprueba formalmente el cierre del ciclo de pruebas. |

Tabla 47. Criterios de salida

6.24. Procedimiento para la ejecución de prueba

El procedimiento para la ejecución de pruebas describe la secuencia de actividades, herramientas, responsables y artefactos necesarios para llevar a cabo el plan de pruebas. Este proceso asegura que las pruebas se realicen de forma controlada, reproducible y alineada con el plan de pruebas definido.

El procedimiento aplicado en este proyecto se estructura en las fases siguientes:

1. Preparación del entorno de pruebas

Antes de iniciar la ejecución, se debe verificar que todos los elementos técnicos estén listos y en condiciones estables, para lo cual:

- a) Se valida el cumplimiento del Test Readiness Review (TRR), asegurando que el entorno, las herramientas, los datos de prueba y los accesos estén disponibles y funcionando correctamente.
- b) Se configura el entorno de pruebas según las especificaciones del proyecto (servidor local, base de datos, navegador, versión del sistema).
- c) Se cargan o restauran los datos de prueba necesarios (usuarios, inventarios, movimientos, entre otros).
- d) Se confirman las versiones y el funcionamiento de las herramientas utilizadas: PHPUnit, Laravel Dusk, Selenium WebDriver, JMeter y OWASP ZAP.

2. Asignación y planificación diaria

El equipo de pruebas revisa el plan general y distribuye los casos según el módulo y la prioridad:

- a) Revisión diaria del avance y asignación de casos por tester.
- b) Verificación de dependencias entre casos y módulos.
- c) Identificación de posibles bloqueos técnicos antes de ejecutar.

3. Ejecución de pruebas automatizadas

La ejecución inicia con las pruebas automatizadas, ya que permiten identificar rápidamente errores de regresión:

- a) Ejecución de los scripts desarrollados
- b) Captura de resultados y evidencias generadas por cada suite.
- c) Registro del estado de cada ejecución (Aprobado/Fallido o ejecutado con hallazgos).
- d) En caso de fallas, se documentan las incidencias con:
 - Descripción del error
 - Pasos para reproducir
 - Evidencia como capturas de pantalla o logs de consola.

4. Ejecución de pruebas manuales

Se ejecutan los casos que no pueden ser automatizados o requieren validación visual:

- a) Cada caso de prueba se ejecuta siguiendo la plantilla establecida (ID, precondiciones, pasos, resultado esperado).
- b) Se registra el resultado obtenido y se compara con el resultado esperado.
- c) Se capturan evidencias visuales (capturas de pantalla o videos).
- d) Si existe un fallo, se genera un defecto en la herramienta de seguimiento.
- e) Se actualiza la matriz de pruebas con el estado correspondiente.

5. Ejecución de pruebas no funcionales

De acuerdo con los criterios definidos en el apartado de pruebas no funcionales:

- a) Pruebas de rendimiento
 - Ejecución de los escenarios definidos: carga concurrente, picos y volumen.
 - Registro de métricas: tiempo promedio de respuesta, throughput, tasa de error.
 - Generación del reporte automático de JMeter para adjuntar como evidencia.
- b) Pruebas de seguridad
 - Escaneo activo y pasivo del sitio.
 - Identificación de vulnerabilidades según niveles (Alto, Medio, Bajo).
 - Registro y documentación de hallazgos, especialmente los de severidad alta.
- c) Pruebas de calidad de código
 - Evaluación del código en busca de bugs, vulnerabilidades y code smells.
 - Registro de métricas generadas por la herramienta.

6. Consolidación de resultados

Al finalizar el ciclo de ejecución:

- a) Se genera un resumen de métricas (casos ejecutados, aprobados, fallidos, defectos por severidad).
- b) Se actualiza la matriz de trazabilidad entre requerimientos y casos.
- c) Se consolidan todos los reportes.
- d) Se prepara el Informe de Defectos Encontrados y el Informe de Cierre de Pruebas.

7. Cierre formal

El ciclo de pruebas finaliza cuando:

- a) Se cumplen los criterios de salida establecidos.
- b) El equipo de QA verifica que la documentación esté completa.
- c) El asesor o responsable del proyecto aprueba formalmente el cierre.

DISEÑO E IMPLEMENTACIÓN DE UN PLAN INTEGRAL DE PRUEBAS MANUALES Y AUTOMATIZADAS PARA ASEGURAR LA CALIDAD DE UN SISTEMA DE COMERCIO ELECTRÓNICO EN PRODUCCIÓN



 Ecommerce

REGISTRO DE EJECUCION DE PRUEBAS ECOMMERCE - IMPORTACIONES GUADALAJARA

Test Execution

| Versión | Fecha | Equipo | Descripción |
|---------|------------|---|---|
| V1.0 | 21/11/2025 | Jennifer Tatiana Cruz Nathaly Adriana Rodríguez Mauricio Daniel Hernández | Se documenta la ejecución de los casos de prueba realizados. Se incluye identificador, propósito, condiciones, procedimiento, resultados esperados y obtenidos, observaciones y evidencias. El objetivo es dejar constancia clara y trazable de las pruebas realizadas, facilitando la revisión y corrección por parte del desarrollador responsable. |

6.25. Métricas de pruebas Resultados de ejecución de las pruebas

La ejecución de pruebas representa un momento importante dentro del proceso de aseguramiento de la calidad, ya que permite validar de manera práctica el cumplimiento de los requisitos establecidos y detectar posibles desviaciones en el comportamiento del sistema.

A continuación, se documenta la ejecución de los casos de prueba definidos para el sistema, detallando las condiciones iniciales, procedimientos realizados, resultados esperados, resultados obtenidos y observaciones relevantes. Cada caso de prueba se presenta con su identificador único, propósito y evidencia correspondiente, con el fin de garantizar su seguimiento y transparencia de registro.

Asimismo, se detallan todas las incidencias, comportamientos atípicos y observaciones relevantes encontradas durante el proceso, garantizando la trazabilidad completa entre los requisitos, los casos de prueba y los resultados finales. Las evidencias recopiladas (capturas de pantalla) respaldan la verificación realizada y permiten asegurar la auditabilidad del proceso de aseguramiento de la calidad.

La ejecución de los casos de prueba se realizó empleando herramientas de apoyo para registro, captura de evidencias y seguimiento, tales como hojas de control, sistemas de gestión de pruebas o herramientas de documentación (Azure DevOps).

El presente apartado proporciona una visión clara y estructurada del desempeño del sistema bajo las diferentes condiciones evaluadas, así como el estado final de aprobación o rechazo de cada caso de prueba ejecutado.

Clasificación de resultados

- **Casos “Aprobados”:** Cumplen con lo esperado.
- **Casos “Fallidos”:** No cumplen con lo esperado.
- **Casos “Ejecutados con hallazgos”:** Cumplen parcialmente o presentan desviaciones menores.

Tipificación de hallazgos

- **Altos:** Impactan la funcionalidad principal.
- **Medios:** Afectan experiencia de usuario.
- **Bajos:** Observaciones menores o mejoras recomendadas.

6.26. Resultados de ejecución de las pruebas

| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP01 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Acceso exitoso | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Verificar que el sistema permita acceso con credenciales válidas | | |
| Módulo/Componente | Login | | |
| Condiciones de prueba | El usuario ejecutor debe colocar en el login sus credenciales. Este usuario debe existir en la base de datos para poder lograr el login exitoso. | | |
| Procedimiento de prueba | 1.Colocar el email del usuario en la caja de texto "Email" 2.Colocar la contraseña del usuario en la caja de texto "Password" 3.Dar Click en el botón "Log in" | | |
| Resultados esperados | El sistema debe permitir el acceso únicamente cuando se ingresen credenciales válidas registradas en la base de datos. | | |
| Resultado obtenido | El sistema permitió el ingreso exitoso del usuario con las credenciales válidas registradas en la base de datos | | |
| Observaciones | Se debería de tener un login independientemente del rol del usuario | | |
| Evidencia |  <p>The terminal window shows the following output:</p> <pre> 121 data = { 122 "Messages": [123 { 124 "From": { 125 "Email": "SENDER_EMAIL", 126 "Name": "CUSCATEC TEST" 127 }, 128 "To": { 129 "Email": "RECIPIENT_EMAIL", 130 "Name": "You" 131 } 132 } 133] 134 } 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 </pre> <p>The email screenshot shows the following content:</p> <p>CP01 - Verificar que el sistema permita acceso con credenciales válidas Recibidos x</p> <p>CUSCATEC TEST a través de bnc3.mailjet.com 10:55 p.m. (hace 0 minutos) ☆</p> <p>para mi ▾</p> <p> Parece que este mensaje está en inglés ×</p> <p>Traducir al español</p> <p>CP01 - Verificar que el sistema permita acceso con credenciales válidas Ha concluido exitosamente!</p> <p>Se coloco usuario en su casilla correspondiente correctamente</p> <p>Se coloco la contraseña en el campo respectivo</p> <p>✓✓✓ El texto capturado contiene exactamente 'Welcome !'</p> | | |

Tabla 48. Caso de prueba 01.

| | | | |
|--------------------------------|---|----------------------|--------------------|
| Identificador del caso | CP02 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Manejo de credenciales incorrectas | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Verificar que el sistema muestre error con credenciales inválidas | | |
| Módulo/Componente | Login | | |
| Condiciones de prueba | El usuario ejecutor debe colocar en el login credenciales que no estén en los registros de la base de datos. Este usuario no debe existir en la base de datos para que muestre mensaje de error. | | |
| Procedimiento de prueba | 1.Colocar un email del usuario incorrecto en la caja de texto “Email” 2.Colocar la contraseña del usuario en la caja de texto “Password” 3.Dar Click en el botón “Log in” | | |
| Resultados esperados | El sistema debe rechazar el intento de acceso, mostrando un mensaje de error que indique credenciales inválidas. El usuario no debe ingresar al sistema ni acceder a la interfaz principal. | | |
| Resultado obtenido | Se muestra un mensaje de que las credenciales no tienen un match en los registros y no ingresa al sistema | | |
| Observaciones | N/A | | |

Evidencia

The image shows two parts of evidence. The top part is a screenshot of Selenium test code in Python. The code sets up a login form with an email field, a password field, and a submit button. It uses Selenium WebDriver to interact with the form, including clearing fields, sending keys, and clicking the submit button. The code includes comments in Spanish explaining the steps.

```

username_input = wait.until(EC.presence_of_element_located((By.NAME, "email")))
# Limpia y escribe
username_input.clear()
username_input.send_keys(DMAIL)
print("Se colocó el correo no registrado en la casilla user correctamente")
# Buscar input por name "password" y por la "userpassword"
try:
    password_input = wait.until(EC.presence_of_element_located((By.ID, "userpassword")))
except:
    password_input = wait.until(EC.presence_of_element_located((By.NAME, "password")))
password_input.clear()
password_input.send_keys(PASSWORD)
print("Se colocó la contraseña en el campo respectivo")
# (Optional) hacer click en el botón de login
try:
    login_btn = wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, "button[type='submit']"), input[type='submit'])))
    login_btn.click()
except:
    # Si no hay botón submit, hacer CTRL desde el campo password
    from selenium.webdriver.common.keys import Keys
    password_input.send_keys(Keys.RETURN)
print("Se utilizó correctamente el botón de Ingresar al sistema")
# Espera corta para ver resultado (en prueba)
time.sleep(10)
# Esperar 10 segundos después del intento de login

```

The bottom part is a screenshot of a Gmail email. The email is titled "EXITO CP02 - Verificar que el sistema muestre error con credenciales inválidas" and is from "CUSCATEC TEST a través de tnc3@mailjet.com". The email content states: "Se colocó el usuario no registrado en la base en la casilla user correctamente. Se colocó la contraseña en el campo respectivo. Se utilizó correctamente el botón de Ingresar al sistema. Mensaje encontrado dentro de 0.000000s. (These credentials do not match our records.) 200 Credenciales incorrectas detectadas." The email also includes a status message: "[Status]: 'success', 'ContentID': '', 'to': [{'Email': 'mauricio.raldon1@gmail.com', 'MessageUID': '1622467-426-436-921-5647263956', 'MessageID': '115292153024242060', 'MessageRef': 'https://api.mailjet.com/v3/REST/message/115292153024242060'}], 'cc': [], 'bcc': []]".

Tabla 49. Caso de prueba 02.

| | | | |
|--------------------------------|--|----------------------|-------------------------|
| Identificador del caso | CP03 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Campos obligatorios vacíos del login | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Ejecutada con hallazgos |
| Propósito del caso | Validar comportamiento con campos vacíos | | |
| Módulo/Componente | Login | | |
| Condiciones de prueba | El usuario ejecutor no debe colocar en el login credenciales. Estos campos deben estar vacíos para la ejecución de esta prueba. | | |
| Procedimiento de prueba | 1. Limpiar los campos usuario y contraseña de ser necesario 2. Dar Click en el botón “Log in” | | |
| Resultados esperados | Validaciones frontend bloquean el envío | | |
| Resultado obtenido | El formulario se envía y carga nuevamente la vista de login con un mensaje de error de campos requeridos | | |
| Observaciones | El frontend no valida que los campos se encuentren debidamente llenos antes del envío, por lo que la request de campos vacíos se envía al servidor y luego regresa la request y muestra el mensaje de error. | | |

Evidencia

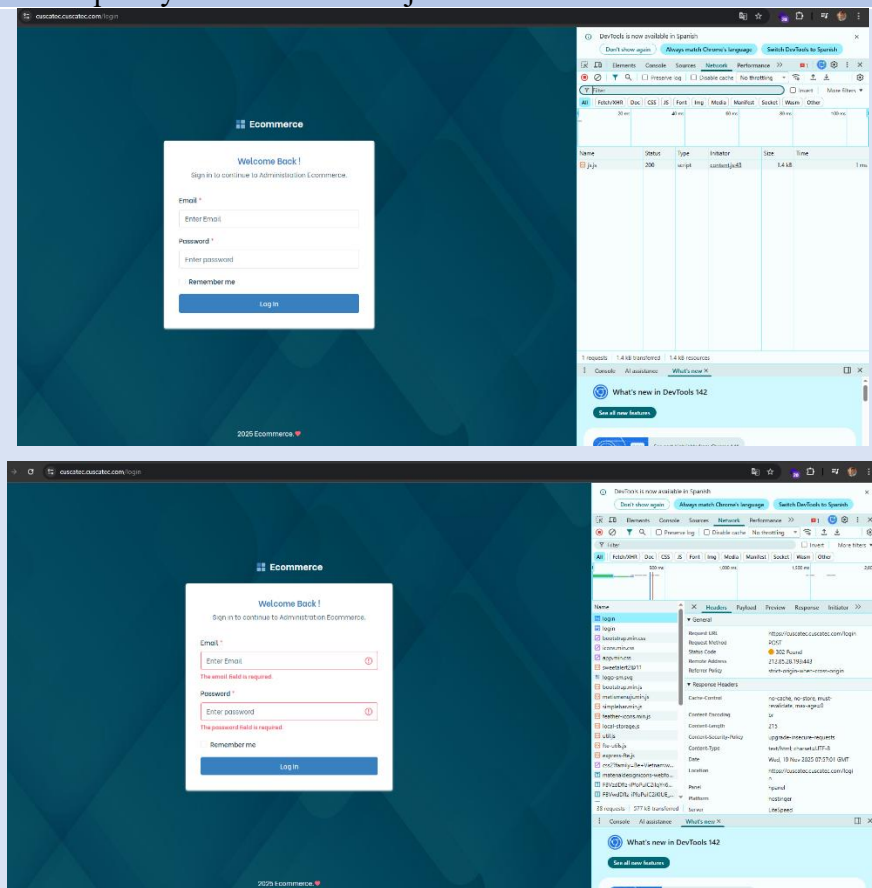


Tabla 50. Caso de prueba 03.

| | | | |
|--------------------------------|---|----------------------|--------------------|
| Identificador del caso | CP04 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Bloqueo por intentos fallidos | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Fallido |
| Propósito del caso | Verifica que la cuenta sea bloqueada luego de varios intentos fallidos | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | El usuario ejecutor hace 10 intentos de ingresar con un usuario registrado en la base de datos que tiene incorrecta la contraseña. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. El usuario ejecutor coloca en la caja de texto un usuario registrado en el sistema 2. El usuario ejecutor coloca una contraseña incorrecta la caja de texto "Password" 3. Dar Click en el botón "Log in" 4. El sistema indica que las credenciales no hacen match con los registros que se tienen 5. Repetir paso del 1 al 4 cinco veces. | | |
| Resultados esperados | El sistema debe bloquear la cuenta del usuario después de varios intentos fallidos consecutivos, impidiendo nuevos accesos | | |
| Resultado obtenido | La cuenta no se bloquea por más intentos fallidos de inicio de sesión con un usuario valido | | |
| Observaciones | Crear un espacio donde se registren los intentos fallidos con un usuario registrado en la base de datos y al pasar de 3 bloquee la cuenta | | |
| Evidencia |  <p>The evidence consists of two parts. The top part is a terminal screenshot showing Selenium test execution. The output indicates that the test script (cp04.py) successfully performed 10 login attempts with incorrect passwords. The final log entry shows a message: "Texto encontrado dentro de ; These credentials do not match our records." This suggests the system is not blocking the account after the specified number of failed attempts. The bottom part is a screenshot of an email notification titled "FALLO CP04 - Verifica que la cuenta sea bloqueada luego de varios intentos fallidos". The email content repeats the test steps and concludes with "Nunca se bloqueo el usuario", which contradicts the expected outcome of account blocking.</p> | | |

Tabla 51. Caso de prueba 04.

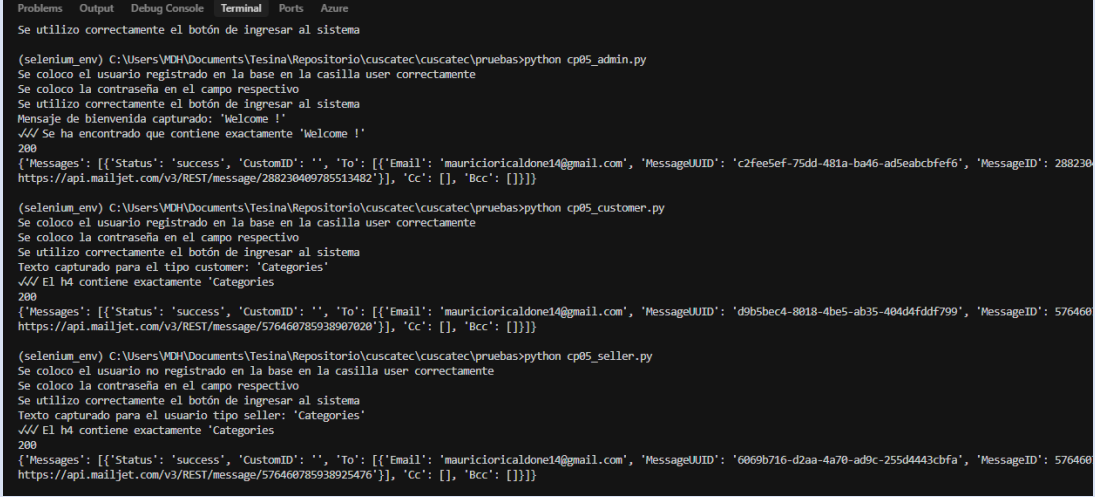
| | | | |
|--------------------------------|---|----------------------|--------------------|
| Identificador del caso | CP05 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Redirección por Rol | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Validar enrutamiento según tipo de usuario | | |
| Módulo/Componente | Login | | |
| Condiciones de prueba | El usuario ejecutor debe colocar en el login sus credenciales. Este usuario debe existir en la base de datos para poder lograr el login exitoso y debe ser redirigido a la vista según su tipo de usuario. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Colocar un email del usuario correcto en la caja de texto “Email” 2. Colocar la contraseña del usuario en la caja de texto “Password” 3. Dar Click en el botón “Log in” | | |
| Resultados esperados | El sistema debe redirigir al usuario autenticado hacia la URL o vista definida para su rol específico, garantizando que cada tipo de usuario acceda únicamente a las funcionalidades que le corresponden. | | |
| Resultado obtenido | El sistema redirigió correctamente al usuario hacia la página esperada para su rol, confirmando que el enrutamiento por rol funciona de manera adecuada y consistente. No se presentaron errores ni accesos indebidos a vistas de otros roles | | |
| Observaciones | Se debería tener un solo login no importando el tipo de usuario que ingrese. | | |
| Evidencia |  <pre> Problems Output Debug Console Terminal Ports Azure Se utilizo correctamente el botón de Ingresar al sistema (selenium_env) C:\Users\VDH\Documents\Tesina\Repositorio\cuscatec\cuscatec\pruebas>python cp05_admin.py Se coloco el usuario registrado en la base en la casilla user correctamente Se coloco la contraseña en el campo respectivo Se utilizo correctamente el botón de ingresar al sistema Mensaje de bienvenida capturado: 'Welcome !' /// Se ha encontrado que contiene exactamente 'Welcome !' 200 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': 'c2fee5ef-75dd-481a-ba46-ad5eabcfbef6', 'MessageID': 288230409785513482}], 'Cc': [], 'Bcc': []}]} https://api.mailjet.com/v3/REST/message/288230409785513482'} (selenium_env) C:\Users\VDH\Documents\Tesina\Repositorio\cuscatec\cuscatec\pruebas>python cp05_customer.py Se coloco el usuario registrado en la base en la casilla user correctamente Se coloco la contraseña en el campo respectivo Se utilizo correctamente el botón de ingresar al sistema Texto capturado para el tipo customer: 'Categories' /// El H4 contiene exactamente 'Categories' 200 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': 'd9b5bec4-8818-4be5-ab35-404d4fddf799', 'MessageID': 576460785938907020}], 'Cc': [], 'Bcc': []}]} https://api.mailjet.com/v3/REST/message/576460785938907020'} (selenium_env) C:\Users\VDH\Documents\Tesina\Repositorio\cuscatec\cuscatec\pruebas>python cp05_seller.py Se coloco el usuario no registrado en la base en la casilla user correctamente Se coloco la contraseña en el campo respectivo Se utilizo correctamente el botón de ingresar al sistema Texto capturado para el usuario tipo seller: 'Categories' /// El H4 contiene exactamente 'Categories' 200 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': '6069b716-d2aa-4a70-ad9c-255d4443cbfa', 'MessageID': 576460785938925476}], 'Cc': [], 'Bcc': []}]} https://api.mailjet.com/v3/REST/message/576460785938925476'} </pre> | | |

Tabla 52. Caso de prueba 05.

| | | | |
|--------------------------------|--|----------------------|-------------------------|
| Identificador del caso | CP06 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Redirección por logueo | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Una vez logueado en el sistema no debe poder ver ningún login hasta cerrar sesión | | |
| Módulo/Componente | Login | | |
| Condiciones de prueba | El usuario ejecutor debe estar ingresado en el sistema, luego este ingresa intenta ingresar a la vista de Login sin haber cerrado sesión, | | |
| Procedimiento de prueba | 1. Estar ingresado en el sistema 2. Ingresar a la URL de Login. | | |
| Resultados esperados | El sistema debe redirigir automáticamente al usuario hacia la página de inicio correspondiente a su rol, impidiendo el acceso a la vista de login mientras la sesión esté activa | | |
| Resultado obtenido | El usuario administrador si cumple con esta condición, sin embargo, los usuarios tipo seller y customer si pueden regresar al login sin haber cerrado su sesión. | | |
| Observaciones | Se debería tener un solo login no importando el tipo de usuario que ingrese. | | |
| Evidencia | <pre>(selenium_env) C:\Users\VDH\Documents\Tesina\Repositorio\cuscatec\cuscatec\pruebas\python_cp06_admin.py Se coloco usuario en su casilla correspondiente correctamente Se coloco la contraseña en el campo respectivo Se utilizo correctamente el botón de ingresar al sistema Se ingreso exitosamente al sistema Se intenta ingresar a la URL del login No es posible ver el login, se ha redirigido a la página principal del tipo de usuario admin 200 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': 'e6c86e13-1139-4980-8609-d56e834e1ab0', 'MessageID': 28823040978715}]}]} https://api.mailjet.com/v3/REST/message/288230409787158188', 'Cc': [], 'Bcc': []]} (selenium_env) C:\Users\VDH\Documents\Tesina\Repositorio\cuscatec\cuscatec\pruebas\python_cp06_seller.py Se coloco usuario en su casilla correspondiente correctamente Se coloco la contraseña en el campo respectivo Se utilizo correctamente el botón de ingresar al sistema Se ingreso exitosamente al sistema Se intenta ingresar a la URL del login Se puede visualizar el login 200 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': 'a9b6c5ec-d440-4e56-8f74-4bfb6e8c8069', 'MessageID': 28823040978722}]}]} https://api.mailjet.com/v3/REST/message/288230409787226058', 'Cc': [], 'Bcc': []]} (selenium_env) C:\Users\VDH\Documents\Tesina\Repositorio\cuscatec\cuscatec\pruebas\python_cp06_customer.py Se coloco usuario en su casilla correspondiente correctamente Se coloco la contraseña en el campo respectivo Se utilizo correctamente el botón de ingresar al sistema Se ingreso exitosamente al sistema Se intenta ingresar a la URL del login Se puede visualizar el login para el tipo de usuario customer 200 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': 'abebe190-7e36-495d-b931-1a1648eac613', 'MessageID': 576460785940607655}]}]} https://api.mailjet.com/v3/REST/message/576460785940607655', 'Cc': [], 'Bcc': []]} </pre> | | |

Tabla 53. Caso de prueba 06.

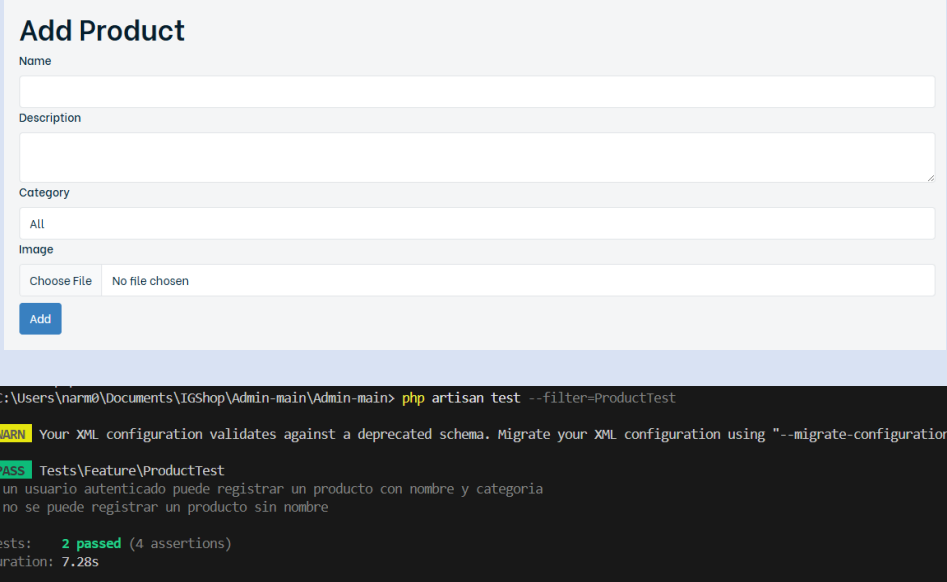
| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP07 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Crear producto válido | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Validar que el sistema permita registrar un producto con todos los campos obligatorios. | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | El usuario debe estar logueado como admin. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder a formulario de creación 2. Completar todos los campos 3. Guardar producto | | |
| Resultados esperados | El sistema debe registrar correctamente un producto cuando todos los campos obligatorios han sido completados, mostrando un mensaje de confirmación. | | |
| Resultado obtenido | El sistema permitió guardar el producto de manera satisfactoria, desplegando el mensaje de confirmación y reflejando el registro en la tabla de productos. | | |
| Observaciones | N/A | | |
| Evidencia |  <p>The evidence section contains two parts. The top part is a screenshot of a web form titled 'Add Product'. It has fields for Name, Description, Category (with a dropdown menu showing 'All'), and Image (with a 'Choose File' button and 'No file chosen' text). There is an 'Add' button at the bottom. The bottom part is a terminal screenshot showing the command <code>php artisan test --filter=ProductTest</code> and its output. The output includes a warning about XML configuration, a pass message for 'Tests\Feature\ProductTest', and two assertions: '✓ un usuario autenticado puede registrar un producto con nombre y categoria' and '✓ no se puede registrar un producto sin nombre'. The final summary shows 'Tests: 2 passed (4 assertions)' and 'Duration: 7.28s'.</p> | | |

Tabla 54. Caso de prueba 07.

| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP08 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Agregar Imagen en formato JPG, PNG o ninguna | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Agregar Imagen en formato JPG, PNG o ninguna. | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | Formulario de creación o edición de producto abierto. | | |
| Procedimiento de prueba | 1. Seleccionar imagen .jpg / .png / no seleccionar imagen 2. Guardar | | |
| Resultados esperados | Imagen aceptada o sin imagen, con formato valido | | |
| Resultado obtenido | Se aceptan archivos únicamente en formato de imagen, de lo contrario, arroja mensaje de error. | | |
| Observaciones | N/A | | |
| Evidencia |  <p>The screenshot shows the 'Add Product' form with the following fields: Name (Prueba imagen), Description (empty), Category (All), and Image (video.mp4). An 'Add' button is visible at the bottom.</p> | | |
| |  <p>The screenshot shows the 'Add Product' form with the following fields: Name (Prueba imagen), Description (empty), Category (All), and Image (No file chosen). A red error message is displayed: 'The image url must be an image.' An 'Add' button is visible at the bottom.</p> | | |

Tabla 55. Caso de prueba 08.

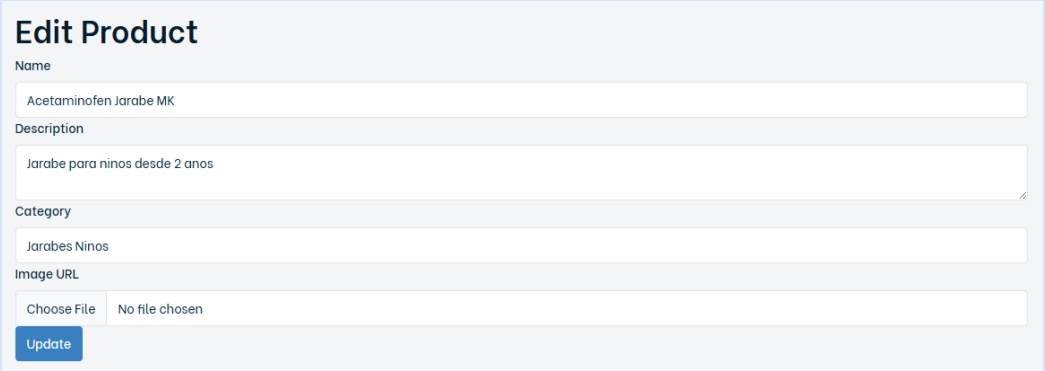
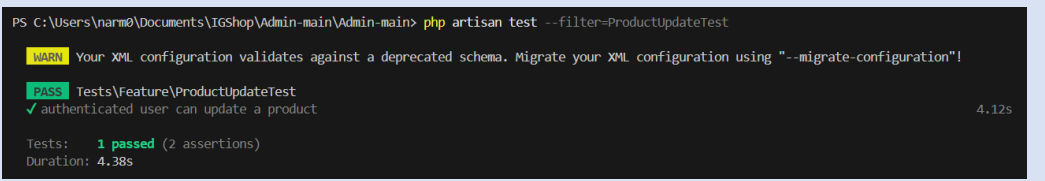
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP09 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Editar producto existente | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Validar que el sistema permita registrar un producto con todos los campos obligatorios. | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | El usuario debe estar logueado como admin. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder a formulario de creación 2. Completar todos los campos 3. Guardar producto | | |
| Resultados esperados | El sistema debe permitir la edición de un producto existente, validando que todos los campos obligatorios se completen antes de guardar. Al finalizar, el producto debe actualizarse en la base de datos y reflejarse en tabla. | | |
| Resultado obtenido | El sistema procesó la edición del producto de manera satisfactoria, y actualizando la información en la lista de productos. | | |
| Observaciones | N/A | | |
| Evidencia |  | | |
| |  | | |

Tabla 56. Caso de prueba 09.

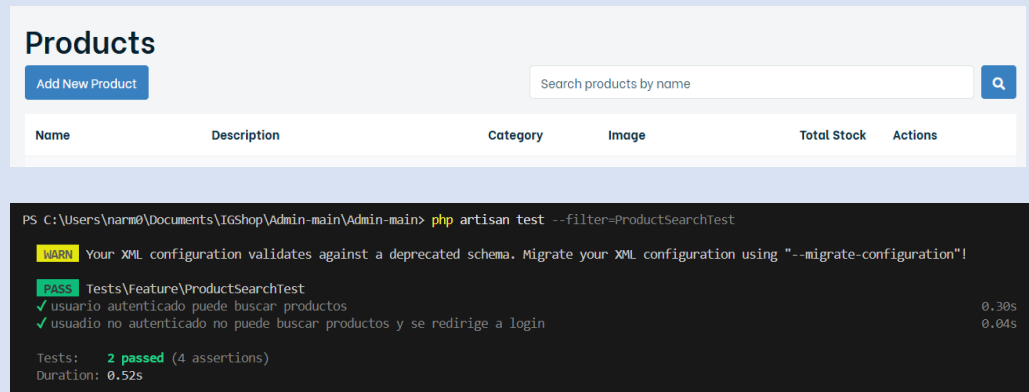
| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP10 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Búsqueda de productos | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Evaluar búsqueda con texto parcial o incompleto | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | Lista de productos visible | | |
| Procedimiento de prueba | 1. Ingresar texto parcial en barra de búsqueda | | |
| Resultados esperados | El sistema debe mostrar en la lista los productos relacionados con el texto parcial o incompleto ingresado en la barra de búsqueda. La funcionalidad debe responder de manera ágil y sin errores. | | |
| Resultado obtenido | Se muestran los productos que incluyen los caracteres ingresados en la búsqueda. No se presentaron fallos ni inconsistencias durante la ejecución. | | |
| Observaciones | N/A | | |
| Evidencia |  <p>The screenshot shows a web application interface with a search bar and a table of products. Below it, a terminal window displays the following output:</p> <pre> PS C:\Users\namr\Documents\IGShop\Admin-main\Admin-main> php artisan test --filter=ProductSearchTest [WARN] Your XML configuration validates against a deprecated schema. Migrate your XML configuration using "--migrate-configuration"! [PASS] Tests\Feature\ProductSearchTest ✓ usuario autenticado puede buscar productos 0.30s ✓ usuario no autenticado no puede buscar productos y se redirige a login 0.04s Tests: 2 passed (4 assertions) Duration: 0.52s </pre> | | |

Tabla 57. Caso de prueba 10.

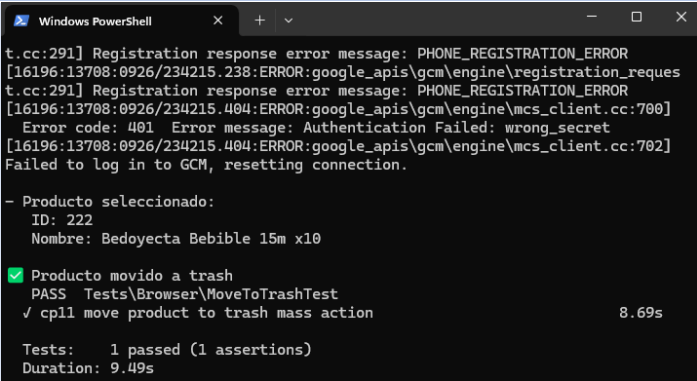
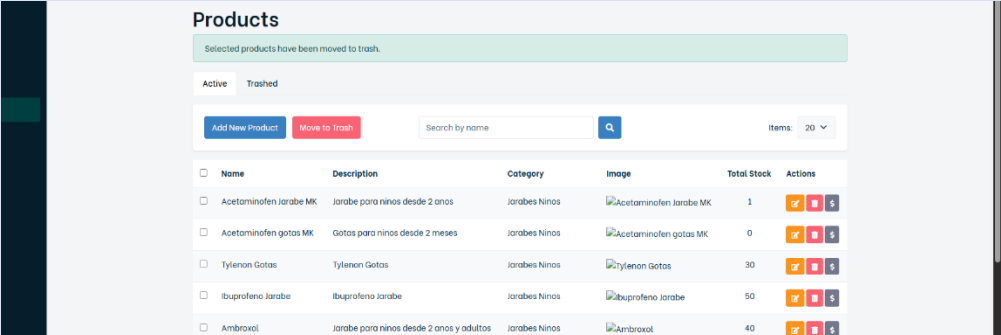
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP11 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Envío de producto a estado “Trashed” | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Validar que un producto activo pueda ser movido al estado Trashed mediante desde la pestaña “Active” | | |
| Módulo/Componente | Gestión de productos/Tabla de productos activos | | |
| Condiciones de prueba | Usuario autenticado como administrador Tener al menos un producto activo en la lista. La pestaña Active debe estar seleccionada. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Abrir la página de productos. 2. Asegurarse de estar en la pestaña Active. 3. Marcar el checkbox del producto a mover. 4. Confirmar el mensaje de alerta de confirmación. 5. Verificar que el producto desaparece de la lista de activos | | |
| Resultados esperados | El producto seleccionado es movido a estado “Trashed”. El mensaje de confirmación de acción se muestra y la operación se completa sin errores. | | |
| Resultado obtenido | Producto movido a trash confirmado en base de datos. Acción ejecutada correctamente. | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 58. Caso de prueba 11.

| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP12 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Eliminación en cascada | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Validar que se desactivan varios productos a la vez | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | Productos activos | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Seleccionar múltiples productos 2. Aplicar acción de desactivar 3 Verificar si se movio a la pestaña correspondiente | | |
| Resultados esperados | El sistema debe permitir la desactivación simultánea de múltiples productos, reflejando el cambio de estado en todos ellos. Los productos seleccionados deben desaparecer de la lista activa y mostrarse en la pestaña correspondiente. | | |
| Resultado obtenido | Los productos seleccionados se mueven a trash y quedan visibles en la papelera, manteniendo los registros en la base de datos. | | |
| Observaciones | Se sugiere renombrar la ruta massDestroy a bulk-delete para mejorar la claridad semántica y alinearse con convenciones RESTful. | | |

Evidencia

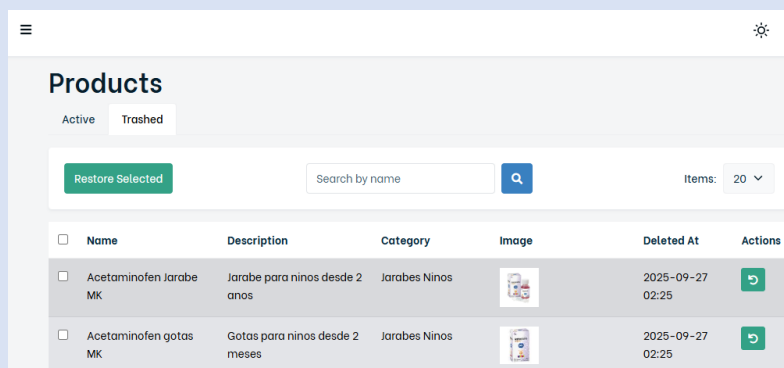
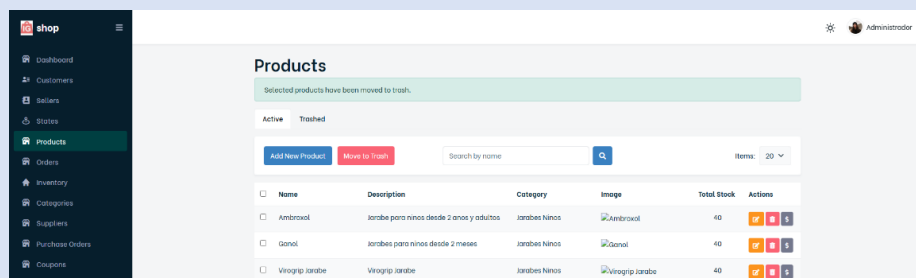


Tabla 59. Caso de prueba 12.

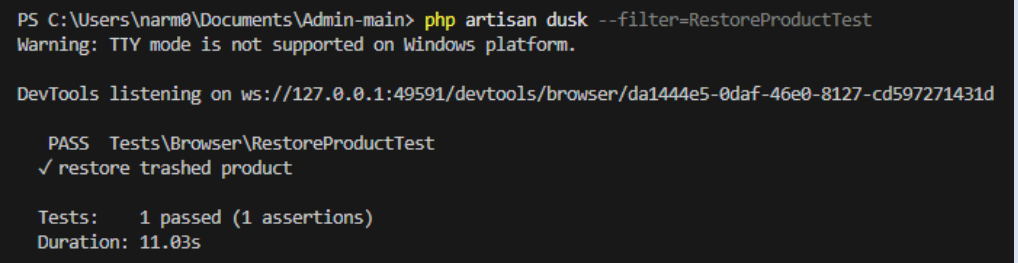
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP13 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Restauración de producto individual | Prioridad | Baja |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Verificar que producto “trashed” se restaure individualmente. | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | Producto en estado “Trashed” | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder al módulo de productos. 2. Seleccionar producto existente 3. Click en el botón “Trashed” 4. Ir a la tab para restaurar el producto 5. Clic en “Restore” | | |
| Resultados esperados | El sistema debe permitir la restauración individual de un producto previamente enviado a la papelera, mostrando nuevamente el registro en la tabla de productos activos. | | |
| Resultado obtenido | El sistema restauró correctamente el producto seleccionado desde la papelera, mostrándolo nuevamente en la tabla de productos activos. Además, desplegó un mensaje de confirmación de éxito. | | |
| Observaciones | N/A | | |
| Evidencia |  <pre> PS C:\Users\narm0\Documents\Admin-main> php artisan dusk --filter=RestoreProductTest Warning: TTY mode is not supported on Windows platform. DevTools listening on ws://127.0.0.1:49591/devtools/browser/da1444e5-0daf-46e0-8127-cd597271431d PASS Tests\Browser\RestoreProductTest ✓ restore trashed product Tests: 1 passed (1 assertions) Duration: 11.03s </pre> | | |

Tabla 60. Caso de prueba 13.

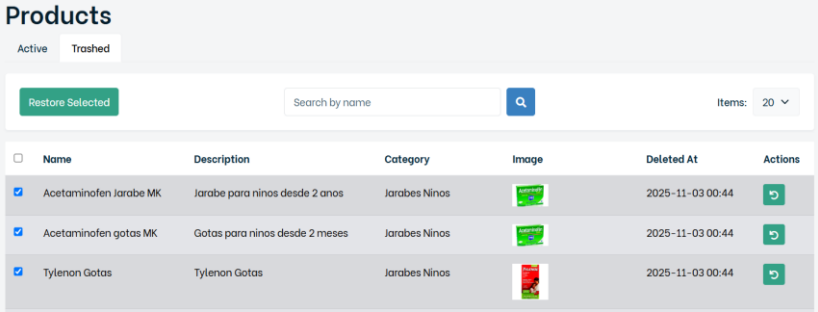
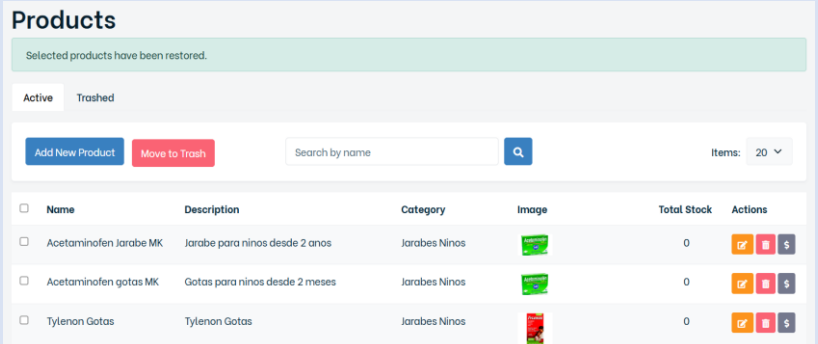
| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP14 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Restauración masiva de productos | Prioridad | Baja |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Validar restauración masiva desde pestaña “Trashed” | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | Lista de productos en “Trashed” | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder al módulo de productos. 2. Seleccionar más de un producto 3. Click en el botón “Trashed” 4. Ir a la tab para restaurar el producto 5. Clic en “Restore” | | |
| Resultados esperados | El sistema debe permitir la restauración masiva de productos previamente enviados a la papelera, mostrando nuevamente todos los registros seleccionados en la lista de productos activos. | | |
| Resultado obtenido | El sistema restauró correctamente los productos seleccionados desde la pestaña “Trashed”, mostrándolos nuevamente en la tabla de productos activos. La funcionalidad se ejecutó de manera consistente | | |
| Observaciones | N/A | | |
| Evidencia | Productos seleccionados:  | | |
| | Productos restaurados:  | | |

Tabla 61. Caso de prueba 14.

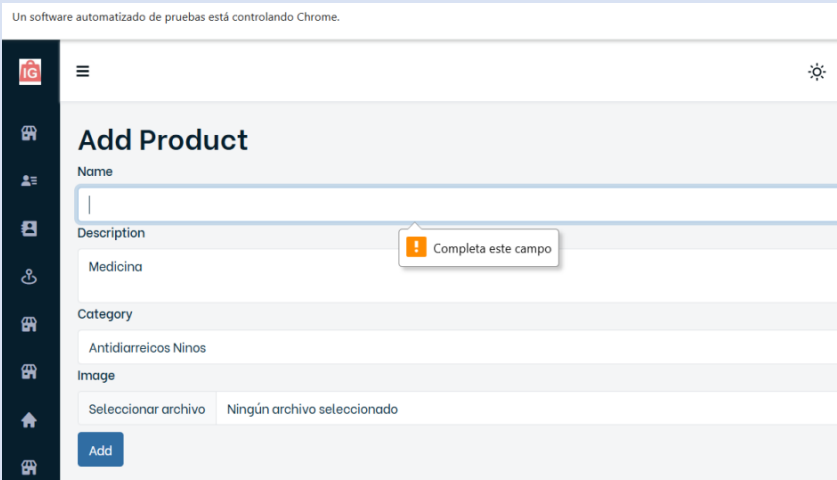
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP15 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Guardar con campos obligatorios vacíos | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Validar restauración masiva desde pestaña “Trashed” | | |
| Módulo/Componente | El usuario debe estar logueado como admin. | | |
| Condiciones de prueba | Lista de productos en “Trashed” | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder al módulo de productos. 2. Click en el botón para crear producto 4. Enviar formulario vacío o parcialmente vacío 5. Validar mensaje de aviso. | | |
| Resultados esperados | El sistema debe impedir el envío del formulario cuando existan campos obligatorios vacíos, mostrando un mensaje de validación claro al usuario. | | |
| Resultado obtenido | No permite ingresar un producto sin nombre ya que ese campo es obligatorio y muestra un mensaje de advertencia que el campo debe ser llenado. | | |
| Observaciones | N/A | | |
| Evidencia | <pre>C:\Users\narm0\Desktop\Tesina\selenium-tests>php tests/ProductNameRequiredTest.php TEST PASSED: La validación HTML5 impidió enviar el formulario (Name requerido).</pre> | | |
| |  | | |

Tabla 62. Caso de prueba 15.

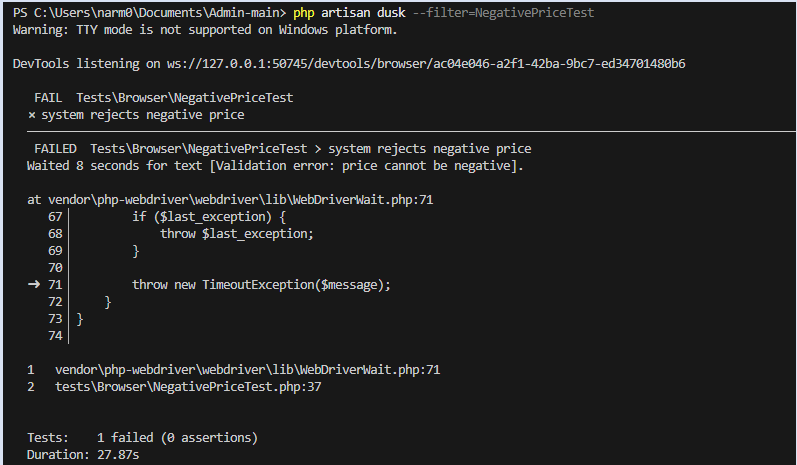
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP16 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Manejo de valores numéricos negativos | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Fallida |
| Propósito del caso | Manejo de valores numéricos negativos | | |
| Módulo/Componente | Gestión de productos | | |
| Condiciones de prueba | Acceso al formulario de actualización de precios. | | |
| Procedimiento de prueba | 1. Ingresar un valor negativo en el campo Precio (ej. -100). 2. Hacer clic en "Guardar". | | |
| Resultados esperados | El sistema debe mostrar un mensaje de error indicando que los valores deben ser mayores a cero. | | |
| Resultado obtenido | El sistema permite guardar el precio negativo sin mostrar ningún mensaje de error y finaliza la operación normalmente, redirigiendo a la lista de productos. | | |
| Observaciones | El error indica que el mensaje de validación nunca apareció, lo que confirma que no existe la restricción en la aplicación. Se recomienda implementar validación en backend (Request o Model) y opcionalmente validar también en frontend para mejorar la experiencia de usuario. | | |
| Evidencia |  <pre> PS C:\Users\narm0\Documents\Admin-main> php artisan dusk --filter=NegativePriceTest Warning: TTY mode is not supported on Windows platform. DevTools listening on ws://127.0.0.1:50745/devtools/browser/ac04e046-a2f1-42ba-9bc7-ed34701480b6 FAIL Tests\Browser\NegativePriceTest x system rejects negative price FAILED Tests\Browser\NegativePriceTest > system rejects negative price Waited 8 seconds for text [Validation error: price cannot be negative]. at vendor\php-webdriver\webdriver\lib\WebDriverWait.php:71 67 if (\$last_exception) { 68 throw \$last_exception; 69 } 70 → 71 throw new TimeoutException(\$message); 72 } 73 } 74 1 vendor\php-webdriver\webdriver\lib\WebDriverWait.php:71 2 tests\Browser\NegativePriceTest.php:37 Tests: 1 failed (0 assertions) Duration: 27.87s </pre> | | |

Tabla 63. Caso de prueba 16.

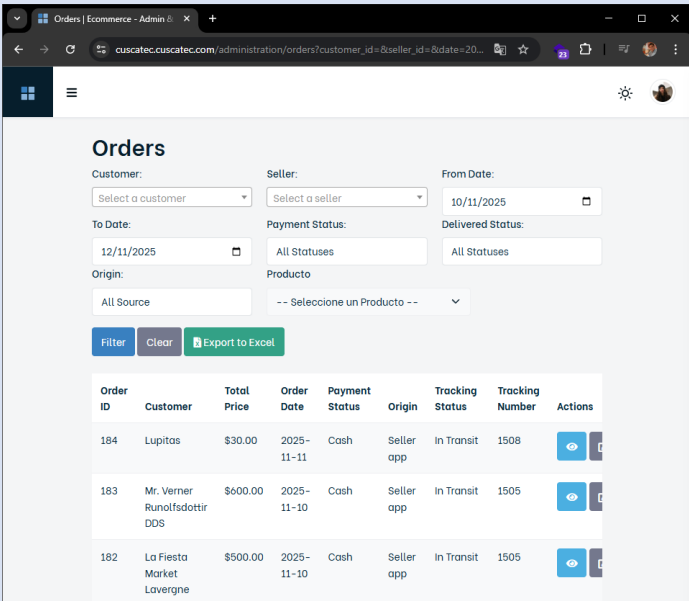
| | | | |
|--------------------------------|---|----------------------|--------------------|
| Identificador del caso | CP17 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Filtrar órdenes por rango de fechas | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Validar que filtros por fechas funcionen correctamente | | |
| Módulo/Componente | Gestión de orders/filtros | | |
| Condiciones de prueba | El usuario ejecutor debe estar ingresado en el sistema Se debe tener ordenes ingresadas en el sistema Las fechas colocadas en el rango de búsqueda deben tener ordenes ya ingresadas previamente. | | |
| Procedimiento de prueba | 1. Estar ingresado en el sistema 2. Ingresar a la vista orders 3. Colocar la fecha 10/11/2025 en el input con name "date" 4. Colocar la fecha 12/11/2025 en el input con name "date_to" 5. Presionar el botón de "Filter" | | |
| Resultados esperados | El sistema debe mostrar únicamente las órdenes registradas dentro del rango de fechas ingresado, excluyendo aquellas que no cumplen con el criterio. | | |
| Resultado obtenido | Resultados filtrados correctamente | | |
| Observaciones | N/A | | |
| Evidencia |  | | |

Tabla 64. Caso de prueba 17.

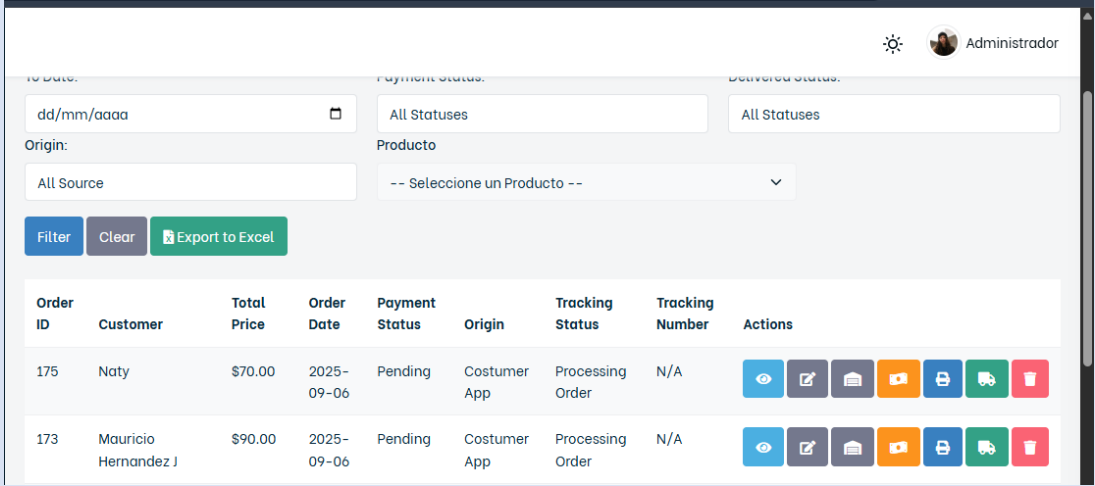
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP18 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Visualización de detalle de orden | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Verificar que el detalle de orden muestre información completa y clara para revisión. | | |
| Módulo/Componente | Orders / Vista de detalle | | |
| Condiciones de prueba | El usuario debe estar autenticado con un rol que tenga acceso al módulo de "Orders". Debe existir al menos una orden creada en el sistema | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder al módulo de órdenes. 2. Filtrar/ubicar una orden existente. 3. Abrir el detalle. 4. Verificar visualmente los campos: cliente, total, fecha, pago, origen, tracking | | |
| Resultados esperados | <p>La página de detalle de la orden se carga correctamente.</p> <p>Se visualiza el nombre del cliente y el vendedor.</p> <p>El monto total de la orden coincide con el esperado</p> <p>La fecha de la orden es visible</p> <p>El estado de seguimiento es visible</p> | | |
| Resultado obtenido | <p>La página de detalle de la orden se carga correctamente.</p> <p>La información del vendedor, cliente, monto, fecha y estado de seguimiento es visible.</p> | | |
| Observaciones | N/A | | |
| Evidencia |  | | |

Tabla 65. Caso de prueba 18.

| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP19 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Actualización de envío de orden y numero de seguimiento | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Validar que una orden existente pueda ser modificada correctamente en cuanto a tipo de estado de envío, así como su número de seguimiento y bodega. | | |
| Módulo/Componente | Órdenes/Formulario de edición | | |
| Condiciones de prueba | Usuario admin existente con correo. Orden existente para actualización de tracking y orden | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticar como usuario admin 2. Editar orden: tracking status y tracking number 3. Validar Warehouse seleccionada 3. Validar que los cambios se encuentren reflejados | | |
| Resultados esperados | El sistema edito correctamente la orden existente El total se mantiene sin ser afectado Número y estado de envío actualizado en la orden, así como la Warehouse. | | |
| Resultado obtenido | Ambos tests pasaron. Los datos de la orden fueron actualizados exitosamente. | | |
| Observaciones | Mantener la lógica de reutilización de tracking number para evitar errores por campos obligatorios. | | |
| Evidencia |   | | |

Tabla 66. Caso de prueba 19.

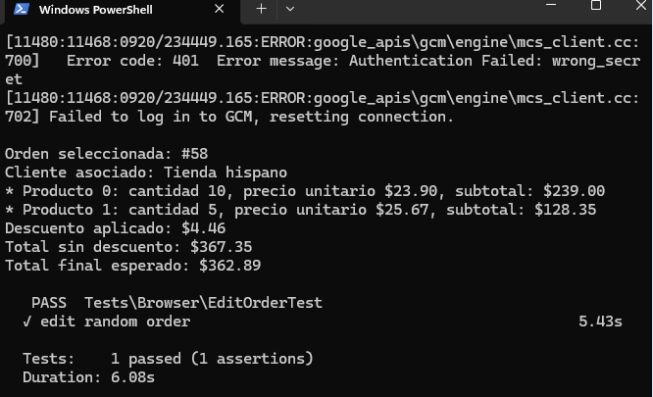
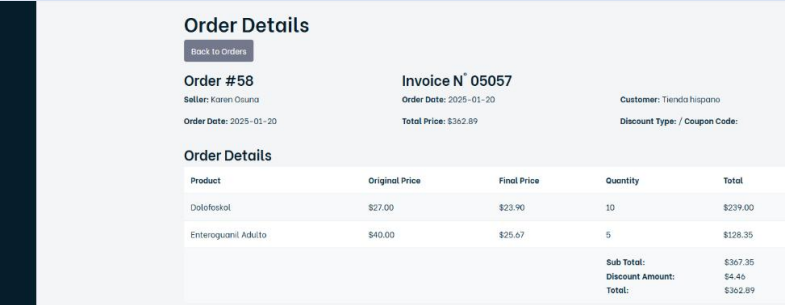
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP20 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Modificación de detalles de orden | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Validar que una orden existente pueda ser modificada correctamente en cuanto a cantidad de productos, precios unitarios, descuentos y cliente asociado, asegurando que los cambios se reflejen en el total calculado. | | |
| Módulo/Componente | Órdenes/Formulario de edición | | |
| Condiciones de prueba | Usuario admin existente y autenticado Orden existente para actualización de tracking y orden | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticar con usuario con permisos 2. Acceder a una orden existente 3. Modificar cantidad, precio unitario y descuento 4. Cambiar cliente asociado 5. Validar que el total se actualice correctamente | | |
| Resultados esperados | El sistema edito correctamente la orden existente El total calculado correctamente actualizado. Número y estado de envío actualizado en la orden | | |
| Resultado obtenido | Los cambios en cantidad, precio, descuento y cliente se reflejan correctamente en la orden y en el total calculado | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 67. Caso de prueba 20.

| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP21 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Actualización de estado de pago | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Validar que el estado del método de pago de una orden pueda actualizarse correctamente | | |
| Módulo/Componente | Administración de ordenes | | |
| Condiciones de prueba | Usuario administrador autenticado Orden existente Estado inicial distinto al estado nuevo | | |
| Procedimiento de prueba | 1. Autenticarse como usuario administrador 2. Seleccionar orden 3. Validar que el cambio se realizó exitosamente. | | |
| Resultados esperados | El estado del método de pago se actualiza correctamente en la base de datos y se refleja en el sistema | | |
| Resultado obtenido | El método de pago fue actualizado de manera exitosa. El estado fue actualizado de C a D (Cheque a Efectivo) | | |
| Observaciones | N/A | | |

Evidencia

```

Windows PowerShell
PS C:\xampp\htdocs\Admin-main> php artisan test tests/Feature/OrderPaymentTest.php

Your XML configuration validates against a deprecated schema. Migrate your XML configuration using "--migrate-configuration"!

Estado de pago actualizado para orden ID: 167

Estado anterior: P → Estado nuevo: C

Cliente asociado: Linas tienda Mexicana.
Estados de pago: P → Pending C → Payment by check D → Cash payment
PASS Tests\Feature\OrderPaymentTest
✓ cp21 actualizar estado de pago de orden existente 0.41s

Tests: 1 passed (4 assertions)
Duration: 0.63s

```

Origin:

All Source

Filter Clear

| Order ID | Customer | Total Price | Order Date | Payment Status | Origin | Tracking Status | Tracking Number | Actions |
|----------|------------------------|-------------|------------|----------------|--------------|------------------|-----------------|---------|
| 167 | Linas tienda Mexicana. | \$15.00 | 2025-04-27 | Check | Costumer App | Processing Order | N/A | |

Tabla 68. Caso de prueba 21.

| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP22 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Exportación de órdenes a Excel | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Verificar que la funcionalidad de exportación de Orders funciona correctamente y se descargue un archivo Excel válido. | | |
| Módulo/Componente | Orders / Exportación | | |
| Condiciones de prueba | Usuario logueado y dashboard cargado, menú Orders accesible | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticar con usuario con permisos 2. Desplegar menú lateral y acceder a Orders/Orders. 3. Hacer click en botón de exportación. 4. Verificar que el archivo se descargue y leer su contenido. | | |
| Resultados esperados | Excel descargado con encabezados. Los registros deben coincidir con los filtros aplicados. | | |
| Resultado obtenido | Excel descargado correctamente con datos y sin errores. | | |
| Observaciones | N/A | | |

Evidencia

```

Windows PowerShell
(selenium_env) PS C:\Users\jenc\OneDrive\Escritorio\Selenium_TestCases> python OrderReport.py
[✓] Login enviado
[✓] Dashboard cargado
[✓] Menú lateral desplegado
[✓] Menú padre 'Orders' activado
[✓] Submenú 'Orders' clickeado
[✓] Click en botón de exportación enviado
[✓] Archivo descargado: C:\Users\jenc\Downloads\orders_2025-09-25_01-58-26.xlsx
[✓] Contenido del Excel:
('Order ID', 'Customer', 'Seller', 'Total Price', 'Order Date', 'Payment Status', 'Origin', 'Tracking Status', 'Tracking Number')
(176, 'Mr. Verner Runolfsson DDS', 'Administrador', 55.0, '2025-09-24', 'Pending', 'Customer App', 'Processing Order', 'N/A')
(175, 'Naty', 'Administrador', 70.0, '2025-09-06', 'Pending', 'Customer App', 'Processing Order', 'N/A')
(175, 'Naty', 'Administrador', 70.0, '2025-09-06', 'Pending', 'Customer App', 'Processing Order', 'N/A')
(173, 'Mauricio Hernandez J', 'Administrador', 370.0, '2025-09-06', 'Pending', 'Customer App', 'In Transit', '365-987654321')

```

| Order ID | Customer | Seller | Total Price | Order Date | Payment Status | Origin | Tracking Status | Tracking Number |
|----------|----------------------------------|---------------|-------------|------------|----------------|--------------|--------------------|-----------------|
| 176 | Mr. Verner Runolfsson DDS | Administrador | 55.0 | 2025-09-24 | Pending | Customer App | Processing Order | N/A |
| 175 | Naty | Administrador | 70.0 | 2025-09-06 | Pending | Customer App | Processing Order | N/A |
| 173 | Mauricio Hernandez J | Administrador | 370.0 | 2025-09-06 | Pending | Customer App | In Transit | 365-987654321 |
| 171 | La mexicana | Jennifer Cruz | 35.0 | 2025-08-18 | Cash | Seller App | Immediate Delivery | N/A |
| 170 | Tienda Y Taqueria las Americas 2 | Jennifer Cruz | 105.0 | 2025-08-18 | Check | Seller App | Processing Order | N/A |
| 169 | Tienda Little Mexico | Jennifer Cruz | 95.0 | 2025-07-20 | Cash | Seller App | Immediate Delivery | N/A |
| 168 | Linax tienda Mexicana. | Karen Osuna | 35.0 | 2025-04-28 | Pending | Seller App | Immediate Delivery | N/A |
| 167 | Linax tienda Mexicana. | Administrador | 15.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 164 | Smyrna Supermercado | Administrador | 70.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 163 | Smyrna Supermercado | Administrador | 105.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 162 | Smyrna Supermercado | Administrador | 17.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 161 | Smyrna Supermercado | Administrador | 210.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 160 | Smyrna Supermercado | Administrador | 90.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 159 | Los Paisanos 2 Nashville | Administrador | 241.0 | 2025-04-27 | Pending | Customer App | Processing Order | N/A |
| 158 | Tienda La Centramericana | Diana Cruz | 511.0 | 2025-02-28 | Check | Seller App | Immediate Delivery | N/A |
| 157 | Supermercado Wow | Diana Cruz | 465.0 | 2025-02-28 | Check | Seller App | Immediate Delivery | N/A |
| 156 | Xiomara wow | Diana Cruz | 480.0 | 2025-02-28 | Cash | Seller App | Immediate Delivery | N/A |

Tabla 69. Caso de prueba 22.

| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP24 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Generación de reporte de ventas por producto | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Verificar que el sistema permita exportar correctamente las órdenes filtradas a un archivo Excel y que los datos sean consistentes. | | |
| Módulo/Componente | Orders / Product Sales Report | | |
| Condiciones de prueba | Usuario válido con permisos para acceder al módulo Orders; navegador configurado para permitir descargas; existencia de órdenes. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al sistema con usuario válido. 2. Navegar a Orders/Product Sales Report. 3. Seleccionar un usuario y definir el rango de fechas 4. Aplicar filtros. 5. Hacer clic en “Export to Excel”. 6. Abrir el archivo descargado y verificar el contenido. | | |
| Resultados esperados | El sistema debe aplicar los filtros correctamente y generar un archivo Excel con las órdenes correspondientes. | | |
| Resultado obtenido | Se aplicaron los filtros correctamente, se descargó el archivo Excel y los datos mostrados corresponden. | | |
| Observaciones | N/A | | |

Evidencia

```

Windows PowerShell
( selenium_env ) PS C:\Users\jenc\OneDrive\Escritorio\Selenium_TestCases> python OrdersReport.py

[✓] Login enviado
[✓] Dashboard cargado
[✓] Menú lateral desplegado
[✓] Submenú Orders desplegado
[✓] Navegando a Product Sales Report
[✓] Filtros aplicados
[✓] Exportación iniciada
[✓] Archivo descargado: C:\Users\jenc\Downloads\product_sales_report.xlsx
[ ] Contenido del Excel:
('Product ID', 'Product Name', 'Total Products Sold')
(1, 'Acetaminofen Jarabe MK', 38)
(2, 'Acetaminofen gotas MK', 5)
(3, 'Tylenon Gotas', 15)
(4, 'Ibuprofeno Jarabe', 43)
(5, 'Ambroxol', 65)
(6, 'Pulmogrip', 23)
(7, 'Ganol', 14)
  
```

| Product ID | Product Name | Total Products Sold |
|------------|------------------------|---------------------|
| 1 | Acetaminofen Jarabe MK | 38 |
| 2 | Acetaminofen gotas MK | 5 |
| 3 | Tylenon Gotas | 15 |
| 4 | Ibuprofeno Jarabe | 43 |
| 5 | Ambroxol | 65 |
| 6 | Pulmogrip | 23 |
| 7 | Ganol | 14 |

Tabla 71. Caso de prueba 24.

| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP25 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Eliminación de orden | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobado |
| Propósito del caso | Verificar que una orden existente pueda eliminarse correctamente desde el sistema y que el registro se elimine físicamente de la base de datos. | | |
| Módulo/Componente | Orders/Administración | | |
| Condiciones de prueba | Orden existente disponible Usuario autenticado con permisos de administrador | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticarse como usuario administrador 2. Seleccionar orden aleatoria 3. Visitar listado de órdenes 4. Ejecutar acción de eliminación 5. Validar que la orden ya no exista | | |
| Resultados esperados | La orden del cliente fue eliminada correctamente. Eliminación confirmada en el sistema y base de datos. | | |
| Resultado obtenido | Orden fue eliminada correctamente Confirmación en base de datos | | |
| Observaciones | N/A | | |

Evidencia

```

Windows PowerShell
PS C:\xampp\htdocs\Admin-main> php artisan dusk --filter=DeleteOrderTest
Warning: TTY mode is not supported on Windows platform.

DevTools listening on ws://127.0.0.1:60613/devtools/browser/a23267bc-48c9-4ac5-a570-6afce608a1f2
[18204:14524:0920/190208.050:ERROR:google_api\gcm\engine\registration_request.cc:291] Registration response error message: DEPRECATED_ENDPOINT

Orden seleccionada para eliminación: #161
Cliente asociado: Smyrna Supermercado
✓ Eliminación física confirmada en base de datos.
PASS Tests\Browser\DeleteOrderTest
✓ cp25 eliminacion fisica de orden existente sin modificar b... 6.03s

Tests: 1 passed (4 assertions)
Duration: 6.60s

```

Orders

Order and its related data have been deleted successfully.

Customer: Seller: From Date:

To Date: Payment Status: Delivered Status:

Origin:

| Order ID | Customer | Order Date | Payment Status | Origin | Tracking Status | Tracking Number | Actions |
|----------|-------------------------|------------|----------------|--------------|------------------|-----------------|---------------------------------------|
| 160 | Smyrna Supermercado | 2025-04-27 | Pending | Customer App | Processing Order | N/A | <input type="button" value="Delete"/> |
| 159 | Los Pasaños 2 Nashville | 2025-04-27 | Pending | Customer App | Processing Order | N/A | <input type="button" value="Delete"/> |

Tabla 72. Caso de prueba 25.

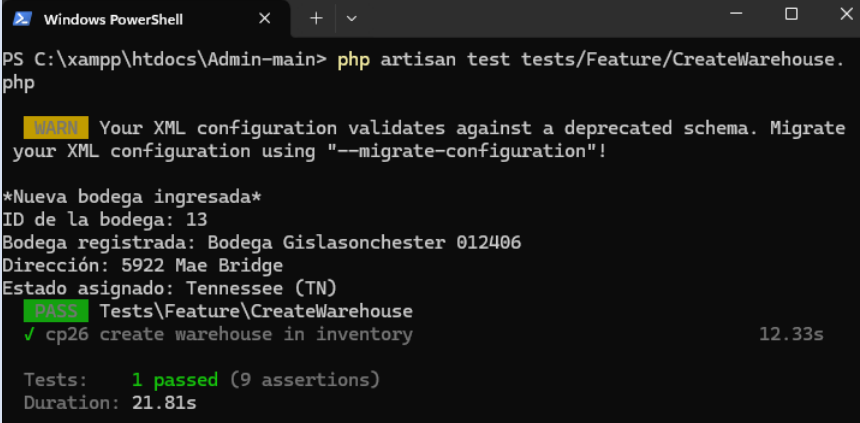
| | | | |
|------------------------------------|---|----------------------|-------------------------|
| Identificador del caso | CP26 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Registro exitoso de bodega | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Validar que el sistema permita registrar una nueva bodega con datos válidos, asegurando trazabilidad en la base de datos. | | |
| Módulo/Componente | Administración/Inventory-Warehouse/Estados | | |
| Condiciones de prueba | Usuario autenticado como administrador Tabla de Estados poblada con múltiples estados Modelo Warehouse con realacion a State | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al sistema con un usuario administrador. 2. Acceder al formulario de creación de bodega. 3. Completar los campos de nombre, dirección y seleccionar un estado. 4. Guardar la bodega. 5. Validar en la base de datos que la bodega se creó correctamente. | | |
| Resultados esperados | La bodega se crea correctamente en la base de datos, el registro contiene: nombre, dirección, estado. | | |
| Resultado obtenido | La Bodega/Warehouse se registra correctamente, sin embargo, el state_id no corresponde al guardado lo que sugiere una sobrescritura en el controlador. Además, debido a la paginación limitada a 10 registros, las bodegas recién creadas pueden no ser visibles en la primera página. | | |
| Observaciones/Recomendación | La bodega se registra correctamente, pero el state_id no se reflejaba en la interfaz por una sobrescritura fija en el controlador (state_id = 1). La vista está limitada a 10 registros por paginación, lo que puede ocultar bodegas nuevas. Se recomienda agregar navegación visible y revisar filtros para asegurar trazabilidad visual. | | |
| Evidencia |  <pre> Windows PowerShell PS C:\xampp\htdocs\Admin-main> php artisan test tests/Feature/CreateWarehouse.php [WARN] Your XML configuration validates against a deprecated schema. Migrate your XML configuration using "--migrate-configuration"! *Nueva bodega ingresada* ID de la bodega: 13 Bodega registrada: Bodega Gislasonchester 012406 Dirección: 5922 Mae Bridge Estado asignado: Tennessee (TN) [PASS] Tests\Feature\CreateWarehouse ✓ cp26 create warehouse in inventory 12.33s Tests: 1 passed (9 assertions) Duration: 21.81s </pre> | | |

Tabla 73. Caso de prueba 26.

| Identificador del caso | CP27 | Ejecutado por | Nathaly Rodríguez | | | | | | | | | | | | | | | |
|--------------------------------|--|----------------------|---------------------|-----------|---------|----------|--------------|---------------------|---------------------------|-----|---------------------|---------------------|----------------------|-----|---------------------|---------------------|---------------------|-----|
| Nombre del caso | Filtro por bodega en inventario | Prioridad | Alta | | | | | | | | | | | | | | | |
| Ciclo | Sprint 1 | Estado | Aprobada | | | | | | | | | | | | | | | |
| Propósito del caso | Mostrar productos según bodega seleccionada | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Gestión de inventario | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Inventario cargado | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ir al módulo de inventario 2. Aplicar filtro 3. Revisar resultados | | | | | | | | | | | | | | | | | |
| Resultados esperados | El sistema debe filtrar correctamente los productos del inventario según la bodega seleccionada, mostrando únicamente aquellos que corresponden a dicha bodega. La operación debe ejecutarse sin errores | | | | | | | | | | | | | | | | | |
| Resultado obtenido | El sistema aplicó el filtro de manera satisfactoria, desplegando únicamente los productos asociados a la bodega seleccionada. | | | | | | | | | | | | | | | | | |
| Observaciones | N/A | | | | | | | | | | | | | | | | | |
| Evidencia | <pre>C:\Users\narm0\Desktop\Tesina\selenium-tests>php tests/InventoryFilterTest.php Filtro funcionando correctamente. Main Warehouse: 357 productos Secondary Warehouse: 358 productos</pre> | | | | | | | | | | | | | | | | | |
| | <p>The screenshot shows a web application interface for 'shop'. The main heading is 'Inventory by Warehouse'. Below the heading, there are two dropdown menus: 'Select Warehouse' (set to 'Secondary Warehouse') and 'Select Product' (set to 'Select a product'). There are 'Filter' and 'Download Excel' buttons. Below these is a table with the following data:</p> <table border="1"> <thead> <tr> <th>Warehouse</th> <th>Product</th> <th>Quantity</th> <th>Last Updated</th> </tr> </thead> <tbody> <tr> <td>Secondary Warehouse</td> <td>Tribedoce Bebible 15m x10</td> <td>100</td> <td>2025-11-05 23:19:57</td> </tr> <tr> <td>Secondary Warehouse</td> <td>Pepeto Pastilaas x24</td> <td>100</td> <td>2025-11-05 23:19:57</td> </tr> <tr> <td>Secondary Warehouse</td> <td>Panadol Extrafuerte</td> <td>100</td> <td>2025-11-05 23:19:57</td> </tr> </tbody> </table> | | | Warehouse | Product | Quantity | Last Updated | Secondary Warehouse | Tribedoce Bebible 15m x10 | 100 | 2025-11-05 23:19:57 | Secondary Warehouse | Pepeto Pastilaas x24 | 100 | 2025-11-05 23:19:57 | Secondary Warehouse | Panadol Extrafuerte | 100 |
| Warehouse | Product | Quantity | Last Updated | | | | | | | | | | | | | | | |
| Secondary Warehouse | Tribedoce Bebible 15m x10 | 100 | 2025-11-05 23:19:57 | | | | | | | | | | | | | | | |
| Secondary Warehouse | Pepeto Pastilaas x24 | 100 | 2025-11-05 23:19:57 | | | | | | | | | | | | | | | |
| Secondary Warehouse | Panadol Extrafuerte | 100 | 2025-11-05 23:19:57 | | | | | | | | | | | | | | | |

Tabla 74. Caso de prueba 27.

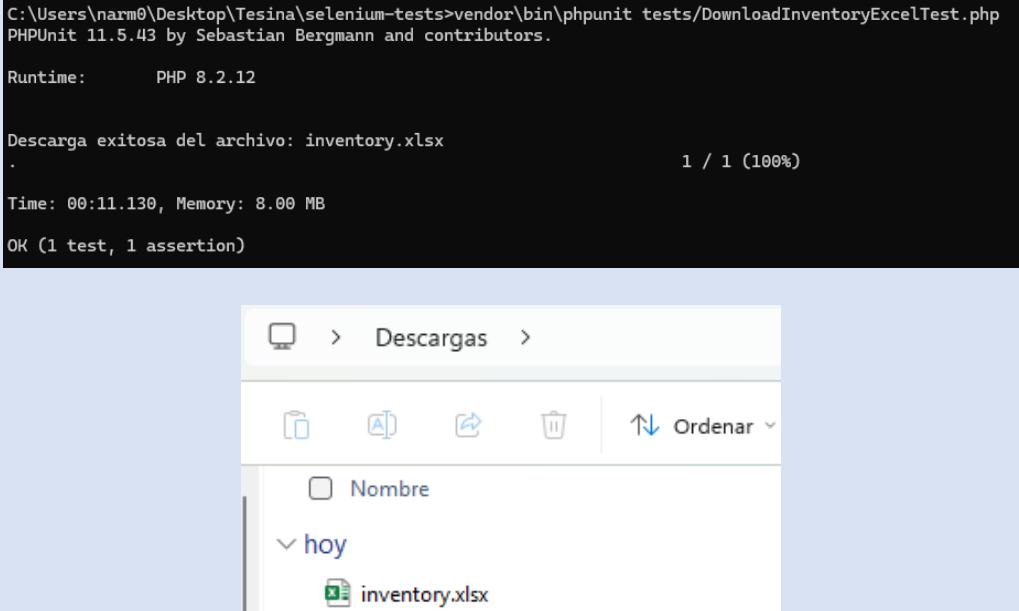
| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP28 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Exportación de inventario a Excel | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Aprobada |
| Propósito del caso | Generar archivo Excel con reporte visible. | | |
| Módulo/Componente | Gestión de inventario | | |
| Condiciones de prueba | Inventario disponible. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ir al módulo de inventario 2. Aplicar filtro 3. Click en “Download Excel” | | |
| Resultados esperados | El sistema debe generar un archivo Excel con los datos del inventario. La operación debe ejecutarse sin errores | | |
| Resultado obtenido | El sistema exportó correctamente el archivo Excel. La funcionalidad se ejecutó de manera satisfactoria, sin incidencias ni inconsistencias. | | |
| Observaciones | N/A | | |
| Evidencia |  <p>The evidence consists of two parts. The top part is a terminal window showing the execution of a PHPUnit test. The command is: <code>C:\Users\narm0\Desktop\Tesina\selenium-tests>vendor\bin\phpunit tests/DownloadInventoryExcelTest.php</code>. The output shows: <code>PHPUnit 11.5.43 by Sebastian Bergmann and contributors.</code>, <code>Runtime: PHP 8.2.12</code>, <code>Descarga exitosa del archivo: inventory.xlsx</code>, <code>1 / 1 (100%)</code>, <code>Time: 00:11.130, Memory: 8.00 MB</code>, and <code>OK (1 test, 1 assertion)</code>. The bottom part is a screenshot of a Windows File Explorer window showing the 'Descargas' (Downloads) folder. It displays a folder named 'hoy' which contains a file named 'inventory.xlsx'.</p> | | |

Tabla 75. Caso de prueba 28.

| Identificador del caso | CP29 | Ejecutado por | Nathaly Rodríguez | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|---|--|-------------------|-----------------------|----------------|---------------------------|----------|---------|-------|--|--|--|--|-----------------------|-------------|------------|-------|--|---|-------|-------|------------|-------|--|---|-------|-------|------------|-------|--|---|-------|--------|------------|-------|--|---|-------|--------|------------|-------|--|---|-------|--------|------------|-------|--|---|-------|--------|------------|-------|--|---|-------|--------|------------|-------|--|---|-------|--------|--|--|--|--|--------------------|
| Nombre del caso | Consulta de historial Kardex mensual | Prioridad | Media | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 2 | Estado | Aprobado | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Validar que se muestra el historial de movimientos filtrado por producto y fecha | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Gestión de inventario/Kardex. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Inventario y Kardex disponibles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | 1. Ingresar al Kardex 2. Seleccionar producto, mes y año 3. Aplicar filtro | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | El sistema debe mostrar el historial mensual del producto seleccionado, con cálculos de entradas y salidas correctos y totales precisos. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | El sistema muestra el historial del producto seleccionado en el mes indicado, incluyendo los movimientos y cálculos correctos. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | Se sugiere agregar la opción de ver el historial solo seleccionando el producto sin necesidad de seleccionar un mes en específico, ya que la búsqueda se vuelve tediosa para el usuario final. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Fecha</th> <th>Tipo Documento</th> <th>Descripción de Movimiento</th> <th>Entradas</th> <th>Salidas</th> <th>Saldo</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td>SALDO ANTERIOR</td> <td>0.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-1.00</td> <td>-1.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-5.00</td> <td>-6.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-6.00</td> <td>-12.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-6.00</td> <td>-18.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-7.00</td> <td>-25.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-7.00</td> <td>-32.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-1.00</td> <td>-33.00</td> </tr> <tr> <td>26/10/2025</td> <td>OUT01</td> <td>Salida de productos por venta en almacén principal</td> <td>-</td> <td>-2.00</td> <td>-35.00</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>SALDO FINAL</td> <td>-35.00</td> </tr> </tbody> </table> | | | Fecha | Tipo Documento | Descripción de Movimiento | Entradas | Salidas | Saldo | | | | | SALDO ANTERIOR | 0.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -1.00 | -1.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -5.00 | -6.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -6.00 | -12.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -6.00 | -18.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -7.00 | -25.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -7.00 | -32.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -1.00 | -33.00 | 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -2.00 | -35.00 | | | | | SALDO FINAL |
| Fecha | Tipo Documento | Descripción de Movimiento | Entradas | Salidas | Saldo | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | SALDO ANTERIOR | 0.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -1.00 | -1.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -5.00 | -6.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -6.00 | -12.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -6.00 | -18.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -7.00 | -25.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -7.00 | -32.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -1.00 | -33.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26/10/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -2.00 | -35.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | SALDO FINAL | -35.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 76. Caso de prueba 29.

| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP30 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Verificación del saldo por movimiento | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobada |
| Propósito del caso | Validar cálculo de saldo en Kardex | | |
| Módulo/Componente | Gestión de inventario/Kardex. | | |
| Condiciones de prueba | El Kardex debe estar cargado | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al Kardex 2. Seleccionar producto, mes y año 3. Revisar cálculos fila por fila | | |
| Resultados esperados | El sistema debe calcular el saldo de cada movimiento como: saldo anterior + entradas – salidas, reflejando valores exactos en cada registro y en el total acumulado. | | |
| Resultado obtenido | El sistema muestra correctamente el saldo por movimiento aplicando la fórmula saldo anterior + entradas – salidas, con totales precisos en el Kardex. | | |
| Observaciones | N/A | | |

Evidencia

Antes:

Filtros de Búsqueda
Selecciona un producto y el periodo para generar el reporte.

Producto: Complejo B Mes: November Año: 2025 [Consultar](#)

Movimientos para: Complejo B
Periodo: November de 2025

| Fecha | Tipo Documento | Descripción de Movimiento | Entradas | Salidas | Saldo |
|------------|----------------|---|----------|---------|-----------------------|
| | | | | | SALDO ANTERIOR -35.00 |
| 10/11/2025 | IN01 | Ingreso de productos al almacén principal | 100.00 | - | 65.00 |
| | | | | | SALDO FINAL 65.00 |

Después:

Filtros de Búsqueda
Selecciona un producto y el periodo para generar el reporte.

Producto: Complejo B Mes: November Año: 2025 [Consultar](#)

Movimientos para: Complejo B
Periodo: November de 2025

| Fecha | Tipo Documento | Descripción de Movimiento | Entradas | Salidas | Saldo |
|------------|----------------|--|----------|---------|-----------------------|
| | | | | | SALDO ANTERIOR -35.00 |
| 10/11/2025 | IN01 | Ingreso de productos al almacén principal | 100.00 | - | 65.00 |
| 11/11/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -1.00 | 64.00 |
| 11/11/2025 | OUT01 | Salida de productos por venta en almacén principal | - | -20.00 | 44.00 |
| | | | | | SALDO FINAL 44.00 |

Tabla 77. Caso de prueba 30.

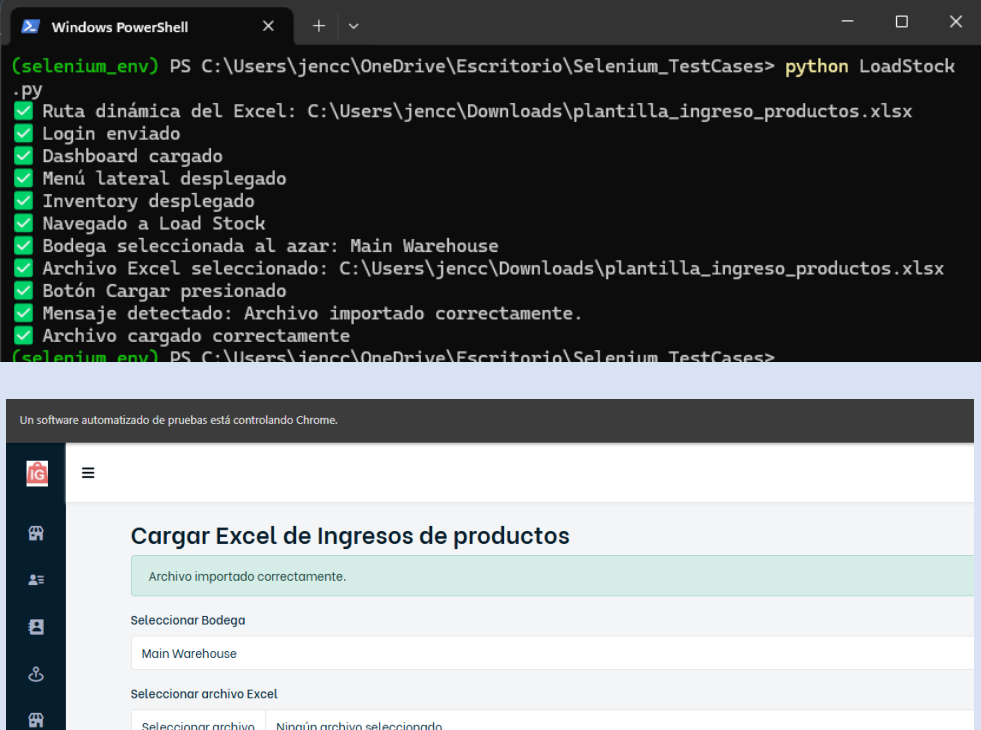
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP31 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Carga masiva de stock desde Excel | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que el módulo de carga de inventario funcione correctamente mediante la carga de un archivo Excel | | |
| Módulo/Componente | Inventory / Load Stock | | |
| Condiciones de prueba | Usuario con permisos de administrador. Archivo Excel válido en la carpeta de Descargas. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Usuario logueado con permisos para crear categorías 2. Navegar a Inventory/Load Stock. 3. Seleccionar una bodega. 4. Seleccionar el archivo Excel. 5. Presionar el botón “Cargar”. 6. Verificar que aparezca el mensaje de éxito. | | |
| Resultados esperados | El archivo se carga correctamente. Mostrar mensaje de confirmación. | | |
| Resultado obtenido | Archivo cargado correctamente. Mensaje “Archivo importado correctamente” | | |
| Observaciones | N/A | | |
| Evidencia |  | | |

Tabla 78. Caso de prueba 31.

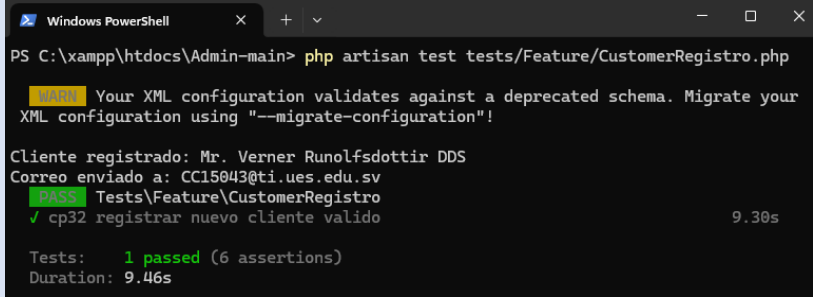
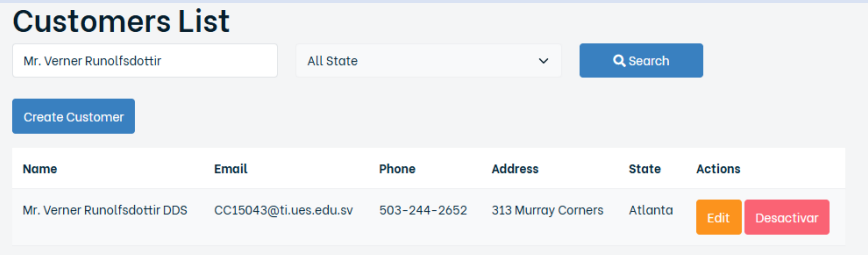
| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP32 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Registrar un nuevo cliente válido | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que el sistema registre correctamente un nuevo cliente con sus datos personales, estado geográfico, usuario asociado. | | |
| Módulo/Componente | Módulo cliente y usuarios. | | |
| Condiciones de prueba | El usuario ejecutor debe estar autenticado como administrador con permisos para registrar clientes. Debe existir al menos un estado válido en la tabla states para asociar al cliente. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticarse como usuario admin 2. Crear un nuevo cliente 3. Verificar que el cliente se haya creado en la tabla customers. 4. Verificar que el usuario asociado se haya creado en la tabla users. 5. Confirmar que la contraseña fue hasheada correctamente. | | |
| Resultados esperados | El sistema registrara un nuevo cliente al sistema de forma exitosa | | |
| Resultado obtenido | Se ejecutó exitosamente con todos los requisitos cumplidos. El sistema registró correctamente un nuevo cliente en la base de datos, y con contraseña hasheada | | |
| Observaciones | Se identificó que el tiempo de ejecución fue de 9.46 segundos. Este incremento se atribuye al envío real del email través de SMTP durante la prueba. | | |
| Evidencia |  <pre> Windows PowerShell PS C:\xampp\htdocs\Admin-main> php artisan test tests/Feature/CustomerRegistro.php [WARN] Your XML configuration validates against a deprecated schema. Migrate your XML configuration using "--migrate-configuration"! Cliente registrado: Mr. Verner Runolfsdottir DDS Correo enviado a: CC15043@ti.ues.edu.sv [PASS] Tests\Feature\CustomerRegistro ✓ cp32 registrar nuevo cliente valido 9.30s Tests: 1 passed (6 assertions) Duration: 9.46s </pre>  | | |

Tabla 79. Caso de prueba 32.

| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP33 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Editar cliente existente | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que el sistema permita editar los datos de un cliente existente y que los cambios se reflejen correctamente en la base de datos. En caso de modificar las credenciales, que este envíe el correo de notificación. | | |
| Módulo/Componente | Gestión de Clientes | | |
| Condiciones de prueba | El cliente debe existir en la base de datos con usuario asociado y estado válido. El usuario administrador debe estar autenticado. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticarse como usuario admin 2. Seleccionar/buscar el cliente a editar 3. Editar cliente 4. Validar cambios en base de datos 5. Confirmar envío de correo | | |
| Resultados esperados | Campos phone, address, email, y password actualizados. Correo enviado a nuevo email. | | |
| Resultado obtenido | Se ejecuto exitosamente. El sistema edito correctamente al cliente existente y se envió el correo con la información requerida. | | |
| Observaciones | N/A | | |
| Evidencia |  <pre> Windows PowerShell PS C:\xampp\htdocs\Admin-main> php artisan test tests/Feature/CustomerEdit.php [WARN] Your XML configuration validates against a deprecated schema. Migrate your XML configuration using "--migrate-configuration"! Correo enviado a: yeneris_s.c@live.com Cliente editado: Elias Garcia (ID: 1) [PASS] Tests\Feature\CustomerEdit ✓ cp33 editar customer con email opcional 8.02s Tests: 1 passed (7 assertions) Duration: 8.55s </pre> | | |

Tabla 80. Caso de prueba 33.

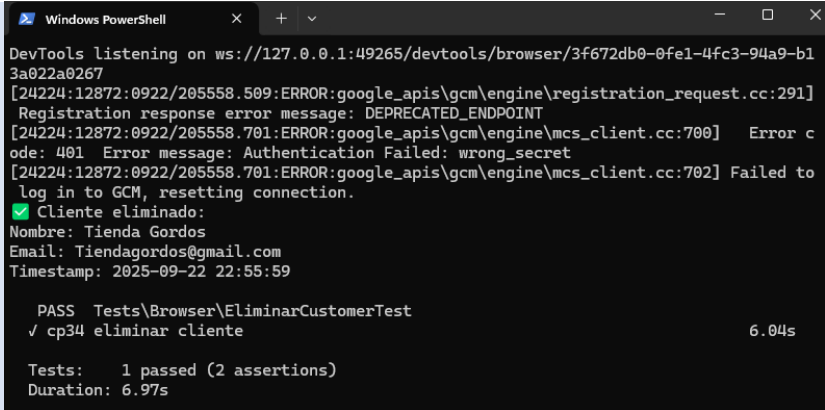
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP34 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Desactivar (eliminar) cliente | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que un cliente puede ser eliminado desde la interfaz de administración, asegurando que ya no aparece en la lista de clientes. | | |
| Módulo/Componente | Gestión de clientes | | |
| Condiciones de prueba | Debe existir al menos un usuario administrador. Debe existir al menos un cliente activo visible en la primera página de la tabla. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al sistema con un usuario administrador. 2. Navegar a la página de lista de clientes 3. Seleccionar un cliente 4. Presionar el botón “Desactivar” y aceptar el diálogo de confirmación. 5. Verificar que el cliente ya no aparezca en la tabla. | | |
| Resultados esperados | El cliente seleccionado debe ser eliminado/desactivado y ya no debe aparecer en la lista de clientes. | | |
| Resultado obtenido | Cliente eliminado correctamente de la primera página de la tabla. Mensaje de éxito mostrado en pantalla. | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 81. Caso de prueba 34.

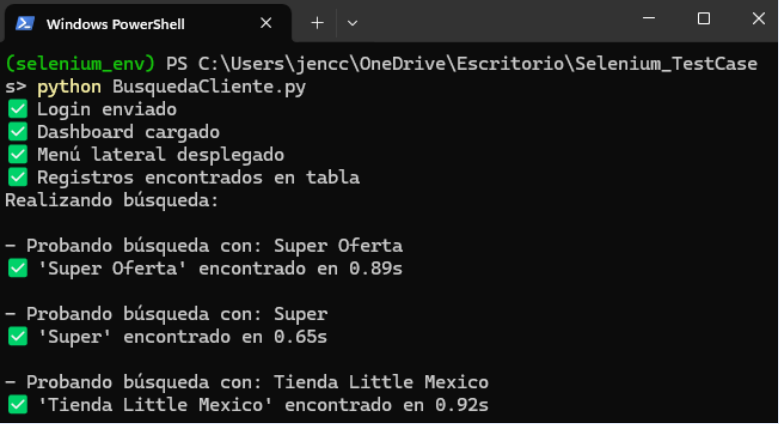
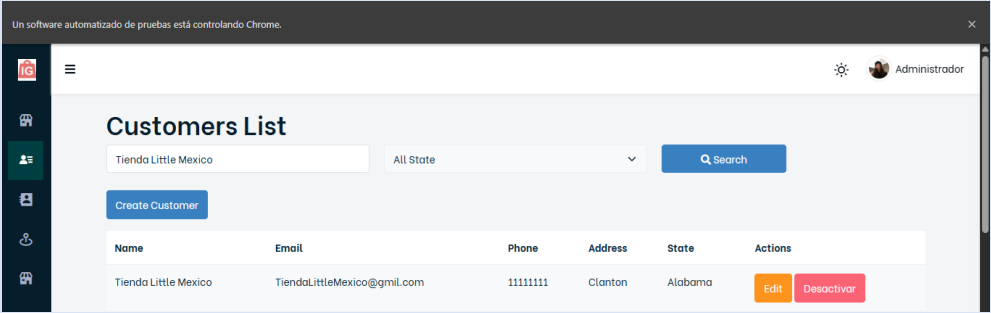
| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP35 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Búsqueda de cliente por nombre | Prioridad | Media |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que el sistema pueda buscar clientes por nombre completo o parcial de manera eficiente, mostrando resultados correctos en menos de 2 segundos. | | |
| Módulo/Componente | Customers / Búsqueda de clientes | | |
| Condiciones de prueba | Ingresar al sistema con un usuario administrador. Existencia de clientes registrados en la tabla. Acceso al módulo Customers. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al sistema con usuario y contraseña válidos. 2. Ir al módulo Customers. 3. Capturar lista de clientes disponibles. 4. Realizar búsqueda por nombre. 5. Medir tiempo de respuesta de cada búsqueda. | | |
| Resultados esperados | Cada cliente buscado debe aparecer en los resultados, y el tiempo de respuesta debe ser menor a 2 segundos. | | |
| Resultado obtenido | Todos los clientes seleccionados fueron encontrados correctamente, con tiempos de respuesta entre 0.32s y 0.98s, cumpliendo el criterio de <2s | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 82. Caso de prueba 35.

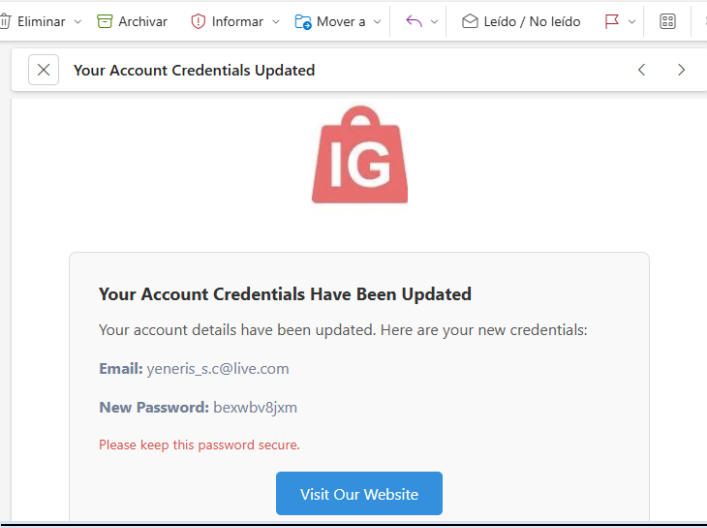
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP36 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Envío de credenciales por email | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que el sistema envíe correos electrónicos con contenido responsivo, seguro, que incluya las credenciales de acceso para nuevos clientes. | | |
| Módulo/ Componente | Módulo de registro de cliente, configuración SMTP, diseño Blade del correo | | |
| Condiciones de prueba | <p>El cliente debe haber sido registrado exitosamente</p> <p>El correo debe enviarse a una dirección válida</p> <p>El sistema debe tener configurado un servidor SMTP funcional</p> <p>El contenido debe incluir nombre, correo y contraseña generada</p> | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Registrar un nuevo cliente 2. Comprobar que el diseño se adapta y no presenta errores de estilo 3. Confirmar que las credenciales fueron enviadas. 4. Validar que el contenido del correo no contiene vulnerabilidades (scripts, enlaces inseguros) | | |
| Resultados esperados | El correo debe enviarse exitosamente, contener los datos correctos, mostrarse con diseño responsivo. | | |
| Resultado obtenido | El correo fue enviado y recibido en cliente de correo. El diseño se adaptó correctamente. Las credenciales fueron enviadas como texto visible. No se detectaron enlaces inseguros. | | |
| Observaciones | El correo envía la contraseña del usuario en texto plano, siendo un riesgo crítico al poder ser interceptadas. Se recomienda reestructurar la lógica para enviar un enlace de restablecimiento de contraseña para mantenerla segura | | |
| Evidencia |  | | |

Tabla 83. Caso de prueba 36.

| | | | |
|--------------------------------|---|----------------------|-------------------------|
| Identificador del caso | CP37 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Generación automática de contraseña | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Ejecución con hallazgos |
| Propósito del caso | Validar que el campo password en el formulario de creación de cliente se autocompleta con una contraseña segura que incluya letras, números y símbolos. | | |
| Módulo/Componente | Customer / Formulario de creación | | |
| Condiciones de prueba | Usuario autenticado con acceso al formulario. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder a customer create como usuario autenticado. 2. Verificar que el campo password esté presente. 3. Capturar el valor generado. 4. Validar que incluya al menos una letra mayúscula, un número y un símbolo | | |
| Resultados esperados | El campo password se autocompleta con una contraseña de al menos 6 caracteres, incluyendo letras, números y símbolos. | | |
| Resultado obtenido | Contraseña generada: q8sT1vaPro – Fallo la validación de símbolo Contraseña generada: Z%hDp\$4on! – Paso la validación | | |
| Observaciones | El generador actual no garantiza la inclusión de símbolos o números. Se recomienda ajustar el script para asegurar que la contraseña autogenerada cumpla con los criterios de seguridad. | | |
| Evidencia | <pre> Windows PowerShell PS C:\xampp\htdocs\Admin-main2> php artisan dusk --filter=GenerateCxPasswordTest Warning: TTY mode is not supported on Windows platform. DevTools listening on ws://127.0.0.1:59017/devtools/browser/6814de25-2ea9-417e-b4b9-df8c3b80578f 🔒 Contraseña generada: IMkYl&54u^ PASS Tests\Browser\GenerateCxPasswordTest ✓ cp37 password is autogenerated on customer create form 2.14s Tests: 1 passed (1 assertions) Duration: 2.82s Windows PowerShell PS C:\xampp\htdocs\Admin-main2> php artisan dusk --filter=GenerateCxPasswordTest Warning: TTY mode is not supported on Windows platform. DevTools listening on ws://127.0.0.1:51358/devtools/browser/78ef9aaa-a0fd-4e92-83b5-23f26f217056 🔒 Contraseña generada: xsKNixUF0U ⚠ Password has no symbol (optional) PASS Tests\Browser\GenerateCxPasswordTest ✓ cp37 password is autogenerated on customer create form 2.66s Tests: 1 passed (1 assertions) Duration: 3.57s </pre> | | |

Tabla 84. Caso de prueba 37.

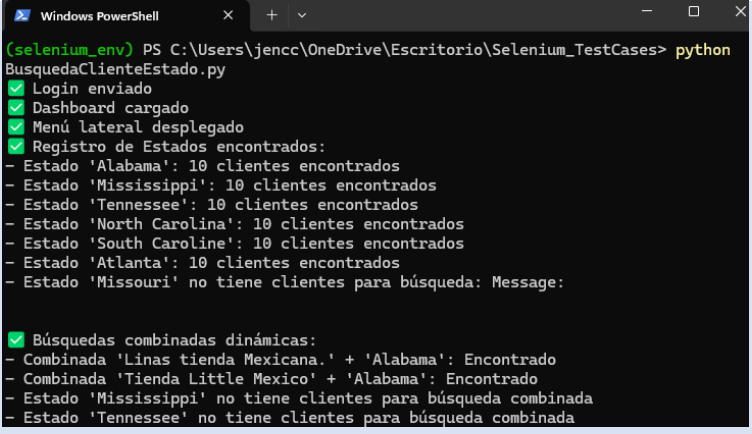
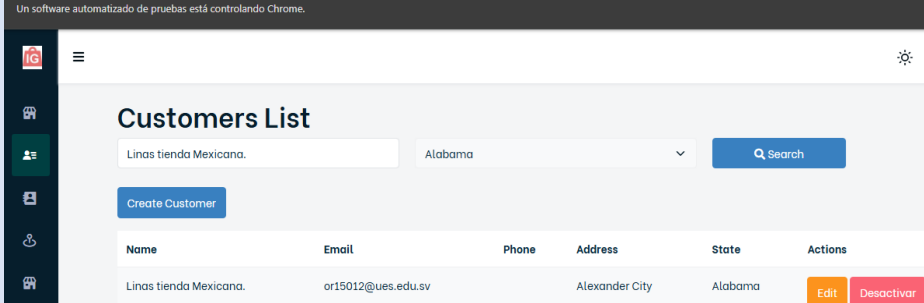
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP38 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Búsqueda de cliente por Estado y combinada | Prioridad | Baja |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que la búsqueda de clientes por estado y la búsqueda combinada (Nombre + Estado) funcione correctamente. | | |
| Módulo/Componente | Customers / Búsqueda de clientes | | |
| Condiciones de prueba | Ingresar al sistema con un usuario administrador. Existencia de clientes registrados en la tabla. Acceso al módulo Customers. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al sistema con el usuario y contraseña. 2. Ir al módulo Customers. 3. Realizar la búsqueda por cada estado y combinada (Nombre + Estado). 4. Registrar resultados. | | |
| Resultados esperados | El sistema debe mostrar clientes por Estado. Las combinaciones válidas de nombre + Estado deben devolver resultados. Las búsquedas sin datos deben excluirse éticamente sin error | | |
| Resultado obtenido | Búsqueda por estado correcta. Búsquedas combinadas correctas | | |
| Observaciones | | | |
| Evidencia |  <pre> Windows PowerShell (seLenium_env) PS C:\Users\jenc\OneDrive\Escritorio\Selenium_TestCases> python BusquedaClienteEstado.py ✓ Login enviado ✓ Dashboard cargado ✓ Menú lateral desplegado ✓ Registro de Estados encontrados: - Estado 'Alabama': 10 clientes encontrados - Estado 'Mississippi': 10 clientes encontrados - Estado 'Tennessee': 10 clientes encontrados - Estado 'North Carolina': 10 clientes encontrados - Estado 'South Caroline': 10 clientes encontrados - Estado 'Atlanta': 10 clientes encontrados - Estado 'Missouri' no tiene clientes para búsqueda: Message: ✓ Búsquedas combinadas dinámicas: - Combinada 'Linás tienda Mexicana.' + 'Alabama': Encontrado - Combinada 'Tienda Little Mexico' + 'Alabama': Encontrado - Estado 'Mississippi' no tiene clientes para búsqueda combinada - Estado 'Tennessee' no tiene clientes para búsqueda combinada </pre> | | |
| |  | | |

Tabla 85. Caso de prueba 38.

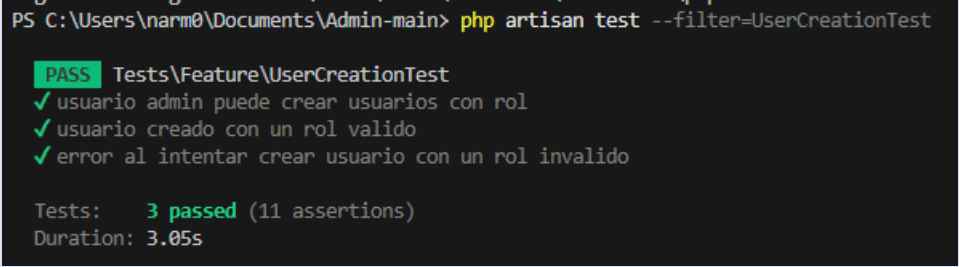
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP39 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Creación de usuario Administrador o Seller | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobada |
| Propósito del caso | Registrar usuario con rol válido | | |
| Módulo/Componente | Gestión de usuarios | | |
| Condiciones de prueba | Usuario Administrador loggeado | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Abrir formulario de la vista de usuarios 2. Click en el botón para crear un nuevo usuario 3. Completar campos 4. Guardar | | |
| Resultados esperados | El sistema debe permitir la creación de un nuevo usuario con rol válido (Administrador o Seller), asignando correctamente el rol seleccionado en el formulario. La operación debe completarse sin errores. | | |
| Resultado obtenido | El sistema creó satisfactoriamente el usuario con el rol asignado, validando que la funcionalidad de registro se ejecuta de manera consistente. El nuevo usuario quedó disponible en la lista de usuarios. | | |
| Observaciones | N/A | | |
| Evidencia |  <pre> PS C:\Users\narm0\Documents\Admin-main> php artisan test --filter=UserCreationTest PASS Tests\Feature\UserCreationTest ✓ usuario admin puede crear usuarios con rol ✓ usuario creado con un rol valido ✓ error al intentar crear usuario con un rol invalido Tests: 3 passed (11 assertions) Duration: 3.05s </pre> | | |

Tabla 86. Caso de prueba 39.

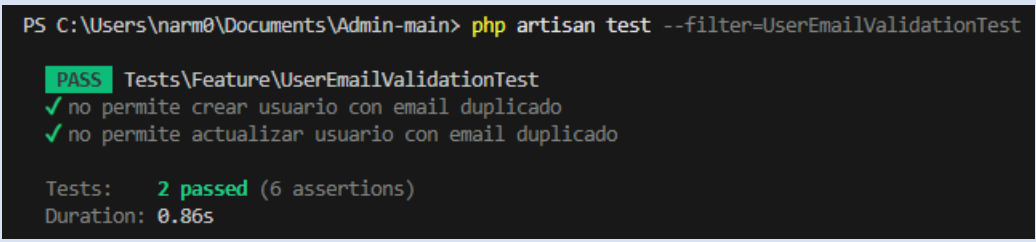
| | | | |
|--------------------------------|---|----------------------|-------------------|
| Identificador del caso | CP40 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Validación de email único | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobada |
| Propósito del caso | Validar que no se permita crear usuario con email duplicado. | | |
| Módulo/Componente | Gestión de usuarios | | |
| Condiciones de prueba | Usuario con email existente. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar a la vista de usuarios 2. Click en el botón para crear un nuevo usuario 3. Intentar crear usuario con email ya registrado | | |
| Resultados esperados | El sistema debe impedir la creación de un nuevo usuario cuando el correo electrónico ingresado ya existe en la base de datos. La validación debe ejecutarse correctamente, mostrando un mensaje de error y garantizando la unicidad del campo email. | | |
| Resultado obtenido | El sistema bloqueó correctamente la creación y actualización de usuarios con correos electrónicos duplicados, confirmando que la validación de unicidad funciona de manera consistente. La operación se ejecutó sin incidencias. | | |
| Observaciones | N/A | | |
| Evidencia |  <pre> PS C:\Users\narm0\Documents\Admin-main> php artisan test --filter=UserEmailValidationTest PASS Tests\Feature\UserEmailValidationTest ✓ no permite crear usuario con email duplicado ✓ no permite actualizar usuario con email duplicado Tests: 2 passed (6 assertions) Duration: 0.86s </pre> | | |

Tabla 87. Caso de prueba 40.

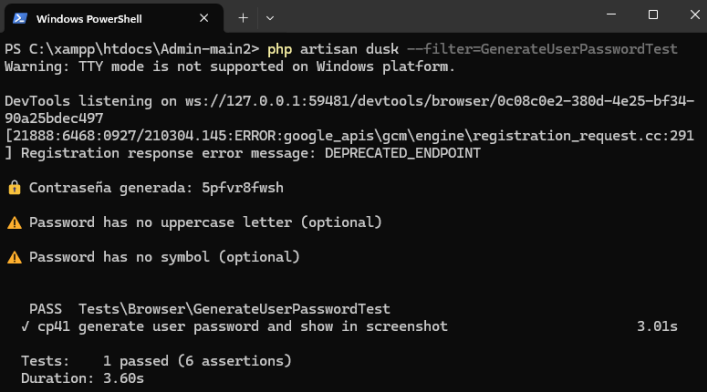
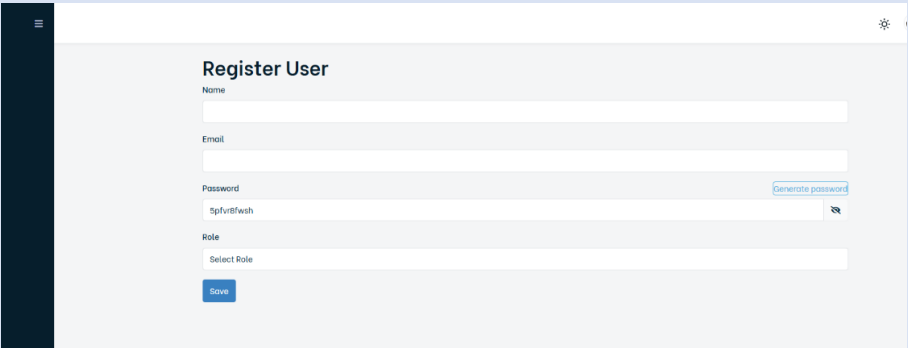
| | | | |
|--------------------------------|--|----------------------|-------------------------|
| Identificador del caso | CP41 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Generación de contraseña de usuario | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Validar que la contraseña de usuario pueda generarse correctamente mediante el botón “Generate password” y que cumpla con los criterios de complejidad. | | |
| Módulo/Componente | Gestión de usuarios / Formulario de registro de usuario | | |
| Condiciones de prueba | Usuario autenticado con permisos. Campo de contraseña inicialmente vacío. Botón “Generate password” disponible. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder al formulario de creación de usuario. 2. Verificar que el campo de contraseña esté vacío. 3. Hacer clic en “Generate password”. 4. Comprobar que el campo se llena con una contraseña. | | |
| Resultados esperados | Campo de contraseña inicialmente vacío. Contraseña generada correctamente al hacer clic en el botón y que cumpla con los criterios mínimos de complejidad | | |
| Resultado obtenido | Contraseña generada mediante botón, campo inicialmente vacío. Los criterios de complejidad de contraseña no siempre son aplicados | | |
| Observaciones | El generador de contraseña actual no garantiza que siempre se cumplan todos los criterios de complejidad (mayúsculas, números y símbolos). Se recomienda mejorar la función para asegurar que cumpla con los requisitos de seguridad | | |
| Evidencia |   | | |

Tabla 88. Caso de prueba 41.

| Identificador del caso | CP42 | Ejecutado por | Nathaly Rodríguez | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|--|---------------------------------|-------------------|---------|------|-------|------|---------|---|---------------|---------------------------------|---------------|------|----|-----------|---------------------|-----------|------|----|---------|---------------------|---------------|------|
| Nombre del caso | Edición de rol de usuario | Prioridad | Alta | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 3 | Estado | Aprobada | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Validar que el usuario admin puede editar el rol de otros usuarios. | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Gestión de usuarios | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Usuario autenticado como Admin. | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder a la vista de usuarios 2. Seleccionar usuario 3. Click en el botón de editar 4. Modificar rol 5. Guardar | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | El sistema debe permitir que un usuario autenticado como Admin edite el rol de otros usuarios, reflejando el cambio de manera inmediata en la base de datos y en la interfaz. La operación debe completarse sin errores. | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | El sistema actualizó correctamente el rol del usuario seleccionado, confirmando que la funcionalidad de edición se ejecuta de forma correcta. El cambio quedó reflejado en la tabla de usuarios y no se presentaron incidencias. | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | N/A | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia |  <p>The evidence consists of two parts. The top part is a terminal screenshot showing a Selenium test command being executed: <code>C:\Users\narm0\Desktop\Tesina\selenium-tests>php EditUserRoleTest.php</code>. The output of the command is <code>TEST PASSED: El usuario se actualizó correctamente.</code>. The bottom part is a screenshot of a web application interface. At the top, it says 'Un software automatizado de pruebas está controlando Chrome.' Below that is a sidebar menu for 'shop' with options like Dashboard, Customers, Sellers, States, Products, and Orders. The main content area is titled 'Users List' and shows a success message: 'User updated successfully'. Below the message is a 'Create User' button and a table of users:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Email</th> <th>Role</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Administrador</td> <td>import.guadalajara.sv@gmail.com</td> <td>Administrator</td> <td>Edit</td> </tr> <tr> <td>84</td> <td>bodeguero</td> <td>bodeguero@gmail.com</td> <td>Bodeguero</td> <td>Edit</td> </tr> <tr> <td>89</td> <td>Adriana</td> <td>adriana@example.com</td> <td>Administrator</td> <td>Edit</td> </tr> </tbody> </table> | | | ID | Name | Email | Role | Actions | 1 | Administrador | import.guadalajara.sv@gmail.com | Administrator | Edit | 84 | bodeguero | bodeguero@gmail.com | Bodeguero | Edit | 89 | Adriana | adriana@example.com | Administrator | Edit |
| ID | Name | Email | Role | Actions | | | | | | | | | | | | | | | | | | | |
| 1 | Administrador | import.guadalajara.sv@gmail.com | Administrator | Edit | | | | | | | | | | | | | | | | | | | |
| 84 | bodeguero | bodeguero@gmail.com | Bodeguero | Edit | | | | | | | | | | | | | | | | | | | |
| 89 | Adriana | adriana@example.com | Administrator | Edit | | | | | | | | | | | | | | | | | | | |

Tabla 89. Caso de prueba 42.


| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP43 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Validación de opciones del campo Rol | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobada |
| Propósito del caso | El dropdown solo debe mostrar roles válidos | | |
| Módulo/Componente | Gestión de usuarios | | |
| Condiciones de prueba | Formulario de usuario abierto | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder a la vista de usuarios 2. Click en el botón para crear un nuevo usuario 3. Click en el botón de editar 4. Clic en el dropdown de roles | | |
| Resultados esperados | El sistema debe mostrar únicamente las opciones válidas en el campo Rol, limitando el dropdown a “Administrador”. La validación debe garantizar que no aparezcan roles no autorizados y que la selección se realice sin errores. | | |
| Resultado obtenido | El sistema desplegó correctamente las opciones esperadas en el dropdown, mostrando únicamente “Administrador”. La funcionalidad se ejecutó de manera correcta, sin incidencias ni roles adicionales visibles. | | |
| Observaciones | N/A | | |
| Evidencia | <pre>C:\Users\narm0\Desktop\Tesina\selenium-tests>php CheckRoleDropdownTest.php TEST PASSED: Las opciones del dropdown son correctas. Roles encontrados: Select Role, Administrator, Bodeguero</pre>  | | |

Tabla 90. Caso de prueba 43.

| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP44 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Validación de campos vacíos | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobada |
| Propósito del caso | Verificar que el sistema no permita crear usuarios con campos obligatorios vacíos. | | |
| Módulo/Componente | Gestión de usuarios | | |
| Condiciones de prueba | Acceso al formulario de "Create User" | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Navegar a la sección de "Users" 2. Hacer clic en el botón "Create User" 3. Dejar todos los campos obligatorios en blanco 4. Hacer clic en "Guardar" 5. Repetir el proceso dejando un campo vacío a la vez | | |
| Resultados esperados | El sistema debe validar que ningún usuario pueda ser creado si existen campos obligatorios vacíos, mostrando mensajes de error específicos como “El nombre es obligatorio”. El formulario debe permanecer en la misma página, sin registrar información incompleta. | | |
| Resultado obtenido | El sistema mostró mensajes de error por cada campo que se dejó vacío y no permitió crear el usuario con los campos obligatorios vacíos. | | |
| Observaciones | N/A | | |
| Evidencia | <pre>C:\Users\narm0\Desktop\Tesina\selenium-tests>php tests/RequiredFieldsUserCreationTest.php TEST 1 (Name vacío): OK → se mostró validación del navegador. TEST 2 (Email vacío): OK → se mostró validación del navegador. TEST 3 (Password vacío): OK → se mostró mensaje del backend.</pre>  | | |

Tabla 91. Caso de prueba 44.

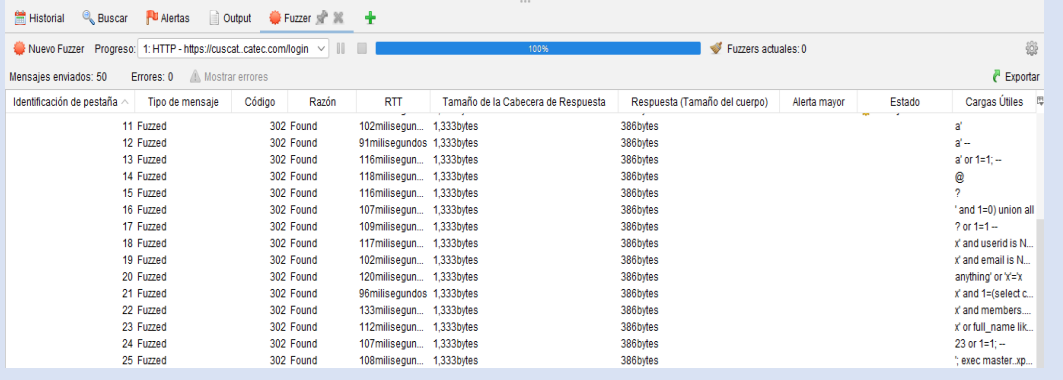
| Identificador del caso | CP45 | Ejecutado por | Jennifer Cruz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|---|----------------------|-----------------|---------------------------|------------------------------------|-------------------------------|--------------|--------|------------------------------------|-------------------------------|--------------|--------|---------------|--|-----------|-----------|-----------------|------------|----------|----|--|--|--|--|-----------|-----------|----------------|------------|----------|------|--|--|--|--|-----------|-----------|-----------------|------------|----------|---------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|---|--|--|--|--|-----------|-----------|-----------------|------------|----------|---|--|--|--|--|-----------|-----------|-----------------|------------|----------|----------------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|-------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|-----------------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|----------------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|------------------|--|--|--|--|-----------|-----------|----------------|------------|----------|-----------------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|-------------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|------------------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|---------------|--|--|--|--|-----------|-----------|-----------------|------------|----------|---------------------|--|--|--|
| Nombre del caso | Inyección SQL | Prioridad | Alta | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 3 | Estado | Aprobado | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Validar que el sistema rechaza intentos de inyección SQL en el formulario de login, sin ejecutar consultas maliciosas ni exponer errores técnicos. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Autenticación | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Usuario no autenticado. Acceso al formulario /login. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Se accede al formulario de login como usuario no autenticado. 2. Se envían múltiples payloads de inyección SQL 3. Se monitorean las respuestas HTTP 4. Verificar si los payloads son reflejados, bloqueados o aceptados. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | <p>El sistema debe rechazar entradas maliciosas. No debe ejecutar consultas SQL no autorizadas. Debe mostrar mensajes de error genéricos o validar el formato antes de enviar. No debe reflejar contenido sin codificación.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | <p>La aplicación no muestra errores SQL, indicando una configuración adecuada No hay No se evidencian vulnerabilidades de SQL Injection por diferencias en las respuestas. Todos los payloads fueron aceptados por el backend con código HTTP 302 Found, sin evidencia de ejecución maliciosa. No se detectó acceso no autorizado ni fuga de datos.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | El código 302 sugiere redirección, probablemente al mismo formulario de login para fallos de autenticación. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia |  <table border="1"> <thead> <tr> <th>Identificación de pestaña</th> <th>Tipo de mensaje</th> <th>Código</th> <th>Razón</th> <th>RTT</th> <th>Tamaño de la Cabecera de Respuesta</th> <th>Respuesta (Tamaño del cuerpo)</th> <th>Alerta mayor</th> <th>Estado</th> <th>Cargas Útiles</th> </tr> </thead> <tbody> <tr><td></td><td>11 Fuzzed</td><td>302 Found</td><td>102milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>a'</td><td></td><td></td><td></td></tr> <tr><td></td><td>12 Fuzzed</td><td>302 Found</td><td>91milisegundos</td><td>1,333bytes</td><td>386bytes</td><td>a'--</td><td></td><td></td><td></td></tr> <tr><td></td><td>13 Fuzzed</td><td>302 Found</td><td>116milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>a' or 1=1; --</td><td></td><td></td><td></td></tr> <tr><td></td><td>14 Fuzzed</td><td>302 Found</td><td>116milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>@</td><td></td><td></td><td></td></tr> <tr><td></td><td>15 Fuzzed</td><td>302 Found</td><td>116milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>?</td><td></td><td></td><td></td></tr> <tr><td></td><td>16 Fuzzed</td><td>302 Found</td><td>107milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>' and 1=0) union all</td><td></td><td></td><td></td></tr> <tr><td></td><td>17 Fuzzed</td><td>302 Found</td><td>109milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>? or 1=1 --</td><td></td><td></td><td></td></tr> <tr><td></td><td>18 Fuzzed</td><td>302 Found</td><td>117milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>x' and userid is N...</td><td></td><td></td><td></td></tr> <tr><td></td><td>19 Fuzzed</td><td>302 Found</td><td>102milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>x' and email is N...</td><td></td><td></td><td></td></tr> <tr><td></td><td>20 Fuzzed</td><td>302 Found</td><td>120milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>anything' or X=x</td><td></td><td></td><td></td></tr> <tr><td></td><td>21 Fuzzed</td><td>302 Found</td><td>96milisegundos</td><td>1,333bytes</td><td>386bytes</td><td>x' and 1=(select c...</td><td></td><td></td><td></td></tr> <tr><td></td><td>22 Fuzzed</td><td>302 Found</td><td>133milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>x' and members...</td><td></td><td></td><td></td></tr> <tr><td></td><td>23 Fuzzed</td><td>302 Found</td><td>112milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>x' or full_name lik...</td><td></td><td></td><td></td></tr> <tr><td></td><td>24 Fuzzed</td><td>302 Found</td><td>107milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>23 or 1=1; --</td><td></td><td></td><td></td></tr> <tr><td></td><td>25 Fuzzed</td><td>302 Found</td><td>108milisegun...</td><td>1,333bytes</td><td>386bytes</td><td>' exec master.xp...</td><td></td><td></td><td></td></tr> </tbody> </table> | | | Identificación de pestaña | Tipo de mensaje | Código | Razón | RTT | Tamaño de la Cabecera de Respuesta | Respuesta (Tamaño del cuerpo) | Alerta mayor | Estado | Cargas Útiles | | 11 Fuzzed | 302 Found | 102milisegun... | 1,333bytes | 386bytes | a' | | | | | 12 Fuzzed | 302 Found | 91milisegundos | 1,333bytes | 386bytes | a'-- | | | | | 13 Fuzzed | 302 Found | 116milisegun... | 1,333bytes | 386bytes | a' or 1=1; -- | | | | | 14 Fuzzed | 302 Found | 116milisegun... | 1,333bytes | 386bytes | @ | | | | | 15 Fuzzed | 302 Found | 116milisegun... | 1,333bytes | 386bytes | ? | | | | | 16 Fuzzed | 302 Found | 107milisegun... | 1,333bytes | 386bytes | ' and 1=0) union all | | | | | 17 Fuzzed | 302 Found | 109milisegun... | 1,333bytes | 386bytes | ? or 1=1 -- | | | | | 18 Fuzzed | 302 Found | 117milisegun... | 1,333bytes | 386bytes | x' and userid is N... | | | | | 19 Fuzzed | 302 Found | 102milisegun... | 1,333bytes | 386bytes | x' and email is N... | | | | | 20 Fuzzed | 302 Found | 120milisegun... | 1,333bytes | 386bytes | anything' or X=x | | | | | 21 Fuzzed | 302 Found | 96milisegundos | 1,333bytes | 386bytes | x' and 1=(select c... | | | | | 22 Fuzzed | 302 Found | 133milisegun... | 1,333bytes | 386bytes | x' and members... | | | | | 23 Fuzzed | 302 Found | 112milisegun... | 1,333bytes | 386bytes | x' or full_name lik... | | | | | 24 Fuzzed | 302 Found | 107milisegun... | 1,333bytes | 386bytes | 23 or 1=1; -- | | | | | 25 Fuzzed | 302 Found | 108milisegun... | 1,333bytes | 386bytes | ' exec master.xp... | | | |
| Identificación de pestaña | Tipo de mensaje | Código | Razón | RTT | Tamaño de la Cabecera de Respuesta | Respuesta (Tamaño del cuerpo) | Alerta mayor | Estado | Cargas Útiles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 Fuzzed | 302 Found | 102milisegun... | 1,333bytes | 386bytes | a' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 Fuzzed | 302 Found | 91milisegundos | 1,333bytes | 386bytes | a'-- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 13 Fuzzed | 302 Found | 116milisegun... | 1,333bytes | 386bytes | a' or 1=1; -- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 14 Fuzzed | 302 Found | 116milisegun... | 1,333bytes | 386bytes | @ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 15 Fuzzed | 302 Found | 116milisegun... | 1,333bytes | 386bytes | ? | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 16 Fuzzed | 302 Found | 107milisegun... | 1,333bytes | 386bytes | ' and 1=0) union all | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 17 Fuzzed | 302 Found | 109milisegun... | 1,333bytes | 386bytes | ? or 1=1 -- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 18 Fuzzed | 302 Found | 117milisegun... | 1,333bytes | 386bytes | x' and userid is N... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 19 Fuzzed | 302 Found | 102milisegun... | 1,333bytes | 386bytes | x' and email is N... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 20 Fuzzed | 302 Found | 120milisegun... | 1,333bytes | 386bytes | anything' or X=x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 21 Fuzzed | 302 Found | 96milisegundos | 1,333bytes | 386bytes | x' and 1=(select c... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 22 Fuzzed | 302 Found | 133milisegun... | 1,333bytes | 386bytes | x' and members... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 23 Fuzzed | 302 Found | 112milisegun... | 1,333bytes | 386bytes | x' or full_name lik... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 24 Fuzzed | 302 Found | 107milisegun... | 1,333bytes | 386bytes | 23 or 1=1; -- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 25 Fuzzed | 302 Found | 108milisegun... | 1,333bytes | 386bytes | ' exec master.xp... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 92. Caso de prueba 45.

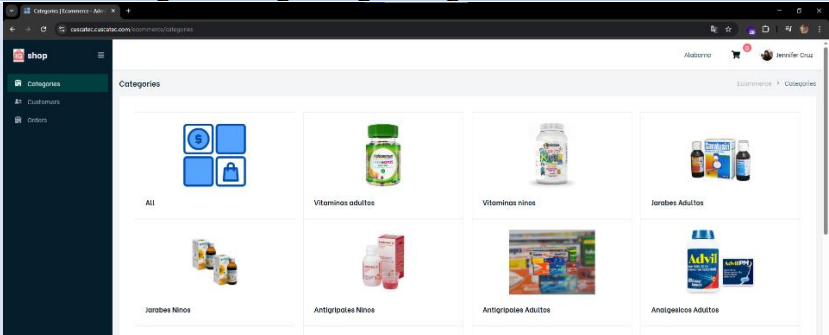
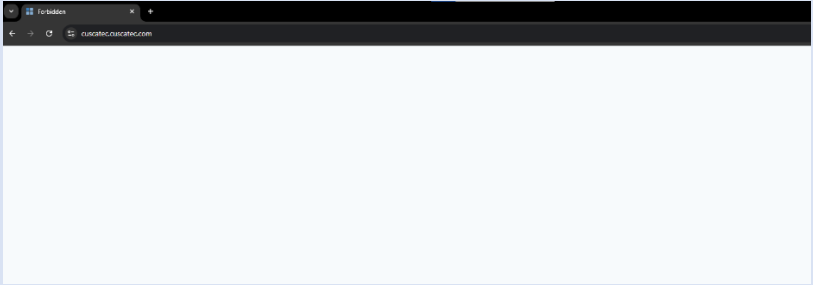
| | | | |
|--------------------------------|---|----------------------|-------------------------|
| Identificador del caso | CP46 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Acceder a vista Admin | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Bloquear acceso a vistas exclusivas de Administrador para usuarios básicos | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | El usuario ejecutor debe estar ingresado en el sistema, y debe intentar ingresar a una url exclusiva de administrador. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Estar ingresado en el sistema como un usuario que no sea de tipo administrador 2. Colocar en la URL la dirección de una vista exclusiva de administrador 3. Entrar a esa URL | | |
| Resultados esperados | El sistema debe mostrar una pantalla informando que el usuario no tiene permisos para visualizar el contenido solicitado, bloqueando completamente el acceso a la vista de administrador. | | |
| Resultado obtenido | El sistema mostró una pantalla informando que el usuario no cuenta con los permisos necesarios para visualizar el contenido solicitado. | | |
| Observaciones | Se recomienda implementar un redireccionamiento automático hacia la pantalla de inicio correspondiente al tipo de usuario que inicia sesión, para mejorar la experiencia, no se registra ningún log de seguridad | | |
| Evidencia |   | | |

Tabla 93. Caso de prueba 46.

| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP47 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Headers HTTP | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Fallido |
| Propósito del caso | Validar la presencia, ausencia y calidad de los headers HTTP en las respuestas del sistema. | | |
| Módulo/Componente | Seguridad / Capa HTTP | | |
| Condiciones de prueba | Navegador Edge configurado con proxy OWASP ZAP. Tráfico interceptado. Solicitudes manuales | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Configurar proxy en Edge 2. Navegar por los módulos del sistema 3. Inspeccionar la respuesta HTTP de las rutas principales 4. Registrar headers presentes. | | |
| Resultados esperados | Headers definidos desde el código deben estar presentes y configurados correctamente. Headers inseguros o ausentes deben ser identificados para corrección. | | |
| Resultado obtenido | Header set-cookie con HttpOnly, Secure, SameSite=Lax, Content-Type, Cache-Control, Content-Security-Policy y Referrer son correctos. Faltan algunos headers de seguridad importantes. El header X-Powered-By está presente. | | |
| Observaciones | X-Powered-By expone la tecnología del backend innecesariamente. Sin embargo, esto no afecta el funcionamiento del sistema. Se recomienda su corrección desde el código fuente por razones de seguridad y buenas prácticas. Adicional, agregar los headers X-Content-Type-Options, X-Frame-Options, Referrer-Policy y Permissions-Policy para reforzar la seguridad del sistema. | | |
| Evidencia | | | |

Tabla 94. Caso de prueba 47.

| | | | |
|--------------------------------|---|----------------------|-------------------------|
| Identificador del caso | CP48 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | 100 usuarios concurrentes | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Simular 100 usuarios haciendo búsquedas simultáneas | | |
| Módulo/Componente | Rendimiento | | |
| Condiciones de prueba | El sistema debe estar activo en un servidor. El test se lleva a cabo con 100 usuarios en 1 segundo. | | |
| Procedimiento de prueba | 1. Abrir archivo .jmx 2. Ejecutar prueba | | |
| Resultados esperados | El sistema debe responder de manera estable, sin caídas ni errores en las solicitudes concurrentes, manteniendo la disponibilidad y tiempos de respuesta. | | |
| Resultado obtenido | Se tuvo un 1.05% de Error en cuanto a caídas de las 100 requests concurrentes | | |
| Observaciones | Se desvía por un margen menor a 2% en cuanto al objetivo esperado de la prueba, de ser necesario esta concurrencia o una un poco superior se debería ver la manera de ampliar los recursos del servidor | | |

Evidencia

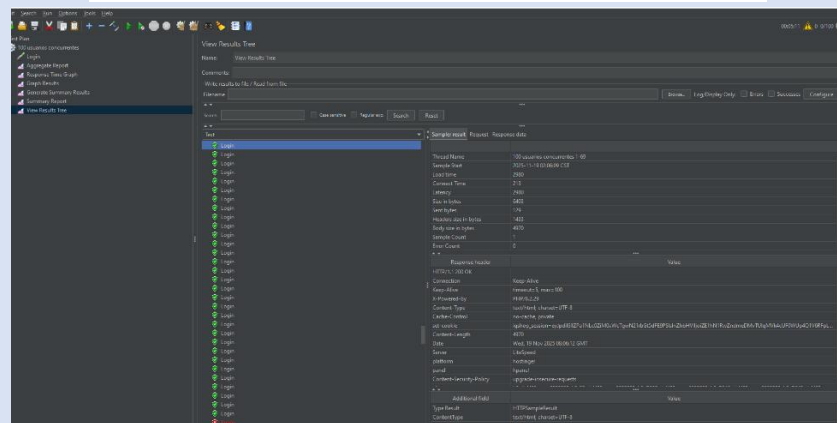
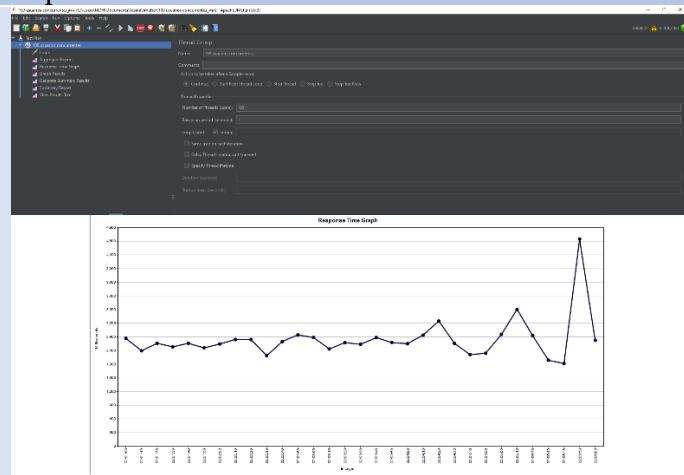


Tabla 95. Caso de prueba 48.

| | | | |
|--------------------------------|--|----------------------|-------------------------|
| Identificador del caso | CP49 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Saturación del sistema | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Simular saturación con 500+ usuarios | | |
| Módulo/Componente | Rendimiento | | |
| Condiciones de prueba | El test se lleva a cabo con 500 usuarios en 1 segundo. | | |
| Procedimiento de prueba | 1. Abrir archivo .jmx 2. Ejecutar archivo | | |
| Resultados esperados | El sistema debe manejar la saturación recuperándose automáticamente, manteniendo la disponibilidad y evitando bloqueos críticos. | | |
| Resultado obtenido | El servidor rechazó las solicitudes cuando se superaron las 100 peticiones provenientes de una misma dirección IP. | | |
| Observaciones | El servidor bloquea cuando son demasiadas solicitudes de una misma dirección IP, esto es excelente en cuanto a seguridad, pero complica pruebas de saturación al sistema | | |

Evidencia

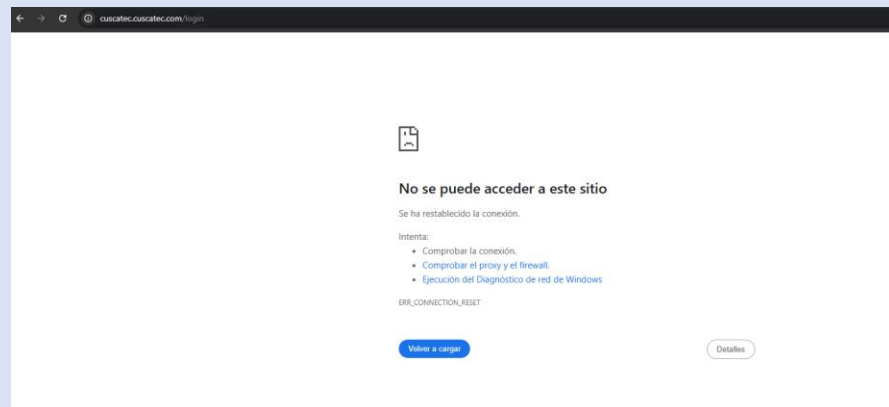
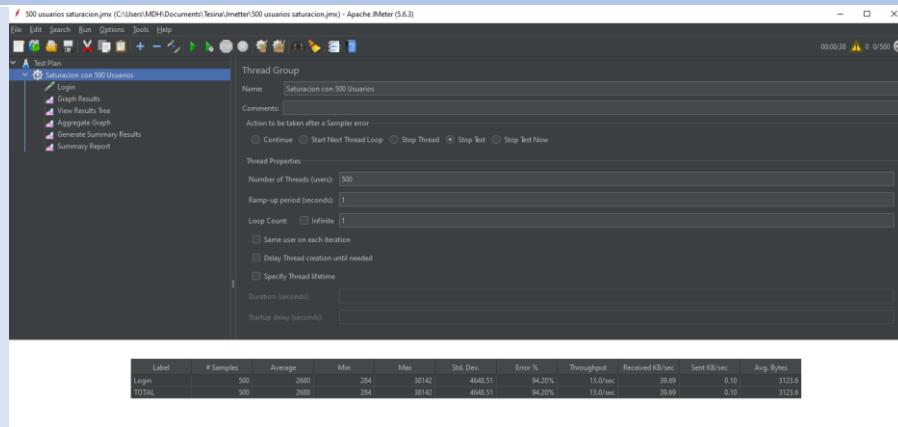


Tabla 96. Caso de prueba 49.

| | | | |
|--------------------------------|---|----------------------|--------------------|
| Identificador del caso | CP50 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Edición del mismo producto | Prioridad | Alta |
| Ciclo | Sprint 1 | Aprobado | Fallido |
| Propósito del caso | Validar edición simultánea sin conflictos | | |
| Módulo/Componente | Concurrencia | | |
| Condiciones de prueba | Dos usuarios ejecutores están modificando el mismo producto al mismo tiempo | | |
| Procedimiento de prueba | 1. Usuario A y Usuario B se encuentran en el mismo producto y guardan la modificación del mismo producto. | | |
| Resultados esperados | El sistema debería permitir que dos usuarios editen el mismo producto al mismo tiempo sin que se pierdan cambios, manteniendo la información correcta y completa. | | |
| Resultado obtenido | Al guardar las ediciones, el sistema reemplaza los valores y solo conserva la última modificación, lo que provoca que se pierdan los cambios anteriores. | | |
| Observaciones | Se deberían actualizar únicamente los campos que sean diferentes al contenido actual en la base de datos | | |

Evidencia

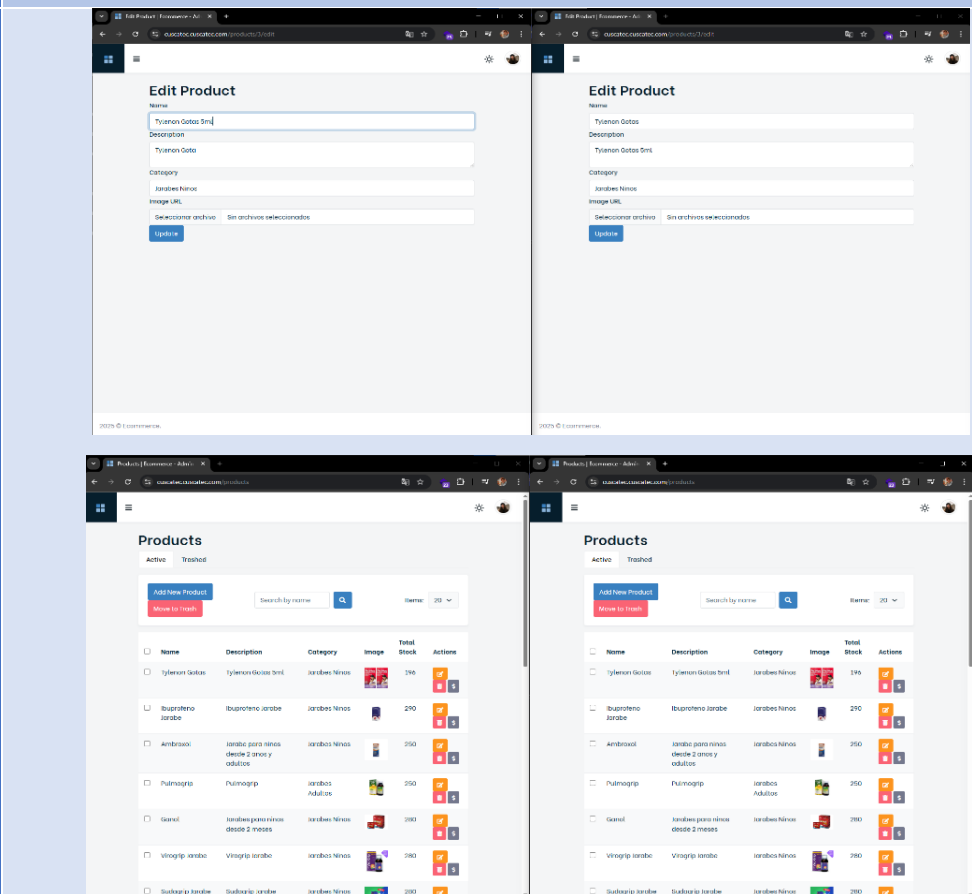


Tabla 97. Caso de prueba 50.

| Identificador del caso | CP51 | Ejecutado por | Nathaly Rodríguez | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|--|----------------------|-------------------|---|--------|-------------------|-----------------|-------------------------------|--------------|-------|-----------|-------------------------------|--------------|------|-----------|-----|-------|-------------------|-----|---|-----|----|--------------|-------------|--|--|--|-----|-------|-------------------|-----|--|-----|----|--------------|--------------|--|--|--|-----|-------|-------------------|-----|---|-----|----|--------------|-------------|--|--|--|-----|-------|-------------------|-----|---|-----|----|-----------------|--------------|--|--|--|-----|-------|-------------------|------|-----------------------------------|-----|-------|----------------|----------|--|--|--|-----|-------|-------------------|-----|---------------------------------------|-----|----|-----------------|-------------|--|--|--|-----|-------|-------------------|-----|---------------------------------------|-----|----|----------------|-------------|--|--|--|-----|-------|-------------------|-----|------------------------------|-----|----|-----------------|-------------|--|--|--|-----|--------|-------------------|-----|------------------------------|-----|----|--------------|-------------|--|--|--|
| Nombre del caso | Gestión de sesión – time-out | Prioridad | Media | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 3 | Estado | Fallida | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Validar que la sesión se cierre tras detectar inactividad. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Seguridad | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Usuario autenticado en el sistema. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Iniciar sesión 2. No realizar ninguna acción por 30 minutos o más. 3. Intentar usar sistema | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | Sesión expira automáticamente y se redirige al login. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | Después de 40 min de inactividad, la sesión no se cerró, lo cual significa que no se está aplicando el timeout por inactividad, | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | Esto representa un riesgo medio. OWASP lo clasifica como una debilidad en el manejo de sesiones. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia | <table border="1"> <thead> <tr> <th>ID</th> <th>Fuente</th> <th>Petición (Tiempo)</th> <th>Método</th> <th>URL</th> <th>Código</th> <th>Razón</th> <th>RTT</th> <th>Respuesta (Tamaño del cuerpo)</th> <th>Alerta mayor</th> <th>Nota</th> <th>Etiquetas</th> </tr> </thead> <tbody> <tr> <td>205</td> <td>Proxy</td> <td>17/11/25 21:52:48</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/assets/fonts/bootsnatch</td> <td>200</td> <td>OK</td> <td>1.47segundos</td> <td>80,374bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>206</td> <td>Proxy</td> <td>17/11/25 21:52:48</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/assets/css/app.min.css</td> <td>200</td> <td>OK</td> <td>1.65segundos</td> <td>102,356bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>208</td> <td>Proxy</td> <td>17/11/25 21:52:48</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/assets/fonts/feather-icom</td> <td>200</td> <td>OK</td> <td>1.44segundos</td> <td>75,098bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>215</td> <td>Proxy</td> <td>17/11/25 21:52:50</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/assets/fonts/material</td> <td>200</td> <td>OK</td> <td>431milisegundos</td> <td>303,580bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>216</td> <td>Proxy</td> <td>17/11/25 22:01:53</td> <td>POST</td> <td>https://cuscatc.cuscatc.com/login</td> <td>302</td> <td>Found</td> <td>92milisegundos</td> <td>402bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>217</td> <td>Proxy</td> <td>17/11/25 22:01:54</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/customers</td> <td>200</td> <td>OK</td> <td>430milisegundos</td> <td>77,255bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>218</td> <td>Proxy</td> <td>17/11/25 22:02:30</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/customers</td> <td>200</td> <td>OK</td> <td>65milisegundos</td> <td>77,255bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>219</td> <td>Proxy</td> <td>17/11/25 22:02:32</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/</td> <td>200</td> <td>OK</td> <td>371milisegundos</td> <td>95,582bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>221</td> <td>Manual</td> <td>17/11/25 22:42:21</td> <td>GET</td> <td>https://cuscatc.cuscatc.com/</td> <td>200</td> <td>OK</td> <td>1.35segundos</td> <td>95,582bytes</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> | | | ID | Fuente | Petición (Tiempo) | Método | URL | Código | Razón | RTT | Respuesta (Tamaño del cuerpo) | Alerta mayor | Nota | Etiquetas | 205 | Proxy | 17/11/25 21:52:48 | GET | https://cuscatc.cuscatc.com/assets/fonts/bootsnatch | 200 | OK | 1.47segundos | 80,374bytes | | | | 206 | Proxy | 17/11/25 21:52:48 | GET | https://cuscatc.cuscatc.com/assets/css/app.min.css | 200 | OK | 1.65segundos | 102,356bytes | | | | 208 | Proxy | 17/11/25 21:52:48 | GET | https://cuscatc.cuscatc.com/assets/fonts/feather-icom | 200 | OK | 1.44segundos | 75,098bytes | | | | 215 | Proxy | 17/11/25 21:52:50 | GET | https://cuscatc.cuscatc.com/assets/fonts/material | 200 | OK | 431milisegundos | 303,580bytes | | | | 216 | Proxy | 17/11/25 22:01:53 | POST | https://cuscatc.cuscatc.com/login | 302 | Found | 92milisegundos | 402bytes | | | | 217 | Proxy | 17/11/25 22:01:54 | GET | https://cuscatc.cuscatc.com/customers | 200 | OK | 430milisegundos | 77,255bytes | | | | 218 | Proxy | 17/11/25 22:02:30 | GET | https://cuscatc.cuscatc.com/customers | 200 | OK | 65milisegundos | 77,255bytes | | | | 219 | Proxy | 17/11/25 22:02:32 | GET | https://cuscatc.cuscatc.com/ | 200 | OK | 371milisegundos | 95,582bytes | | | | 221 | Manual | 17/11/25 22:42:21 | GET | https://cuscatc.cuscatc.com/ | 200 | OK | 1.35segundos | 95,582bytes | | | |
| ID | Fuente | Petición (Tiempo) | Método | URL | Código | Razón | RTT | Respuesta (Tamaño del cuerpo) | Alerta mayor | Nota | Etiquetas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 205 | Proxy | 17/11/25 21:52:48 | GET | https://cuscatc.cuscatc.com/assets/fonts/bootsnatch | 200 | OK | 1.47segundos | 80,374bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 206 | Proxy | 17/11/25 21:52:48 | GET | https://cuscatc.cuscatc.com/assets/css/app.min.css | 200 | OK | 1.65segundos | 102,356bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 208 | Proxy | 17/11/25 21:52:48 | GET | https://cuscatc.cuscatc.com/assets/fonts/feather-icom | 200 | OK | 1.44segundos | 75,098bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 215 | Proxy | 17/11/25 21:52:50 | GET | https://cuscatc.cuscatc.com/assets/fonts/material | 200 | OK | 431milisegundos | 303,580bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 216 | Proxy | 17/11/25 22:01:53 | POST | https://cuscatc.cuscatc.com/login | 302 | Found | 92milisegundos | 402bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 217 | Proxy | 17/11/25 22:01:54 | GET | https://cuscatc.cuscatc.com/customers | 200 | OK | 430milisegundos | 77,255bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 218 | Proxy | 17/11/25 22:02:30 | GET | https://cuscatc.cuscatc.com/customers | 200 | OK | 65milisegundos | 77,255bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 219 | Proxy | 17/11/25 22:02:32 | GET | https://cuscatc.cuscatc.com/ | 200 | OK | 371milisegundos | 95,582bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 221 | Manual | 17/11/25 22:42:21 | GET | https://cuscatc.cuscatc.com/ | 200 | OK | 1.35segundos | 95,582bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 98. Caso de prueba 51

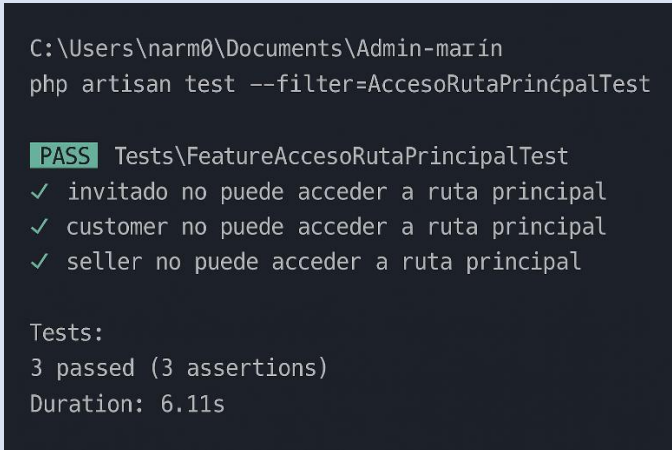
| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP52 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Restricción de acceso por rol | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que usuarios no autorizados no puedan acceder a funciones restringidas | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | Un usuario logueado o anónimo intenta acceder a una vista que necesita permisos con los que no cuenta | | |
| Procedimiento de prueba | 1. Ingresamos como seller al sistema 2. Intentamos acceder al dashboard de administración a través de la URL | | |
| Resultados esperados | El sistema debe denegar el acceso y mostrar un mensaje indicando que el usuario no cuenta con los permisos necesarios. No debe exponerse información sensible ni permitir acceso parcial a la vista restringida. | | |
| Resultado obtenido | El sistema denegó el acceso y mostró un mensaje informando que el usuario no cuenta con los permisos requeridos. No se expuso información sensible ni se permitió acceso al contenido restringido. | | |
| Observaciones | Se debería redireccionar luego de informar que no se tiene acceso. | | |
| Evidencia |  <pre> C:\Users\narm0\Documents\Admin-marín php artisan test --filter=AccesoRutaPrincipalTest PASS Tests\FeatureAccesoRutaPrincipalTest ✓ invitado no puede acceder a ruta principal ✓ customer no puede acceder a ruta principal ✓ seller no puede acceder a ruta principal Tests: 3 passed (3 assertions) Duration: 6.11s </pre> | | |

Tabla 99. Caso de prueba 52.

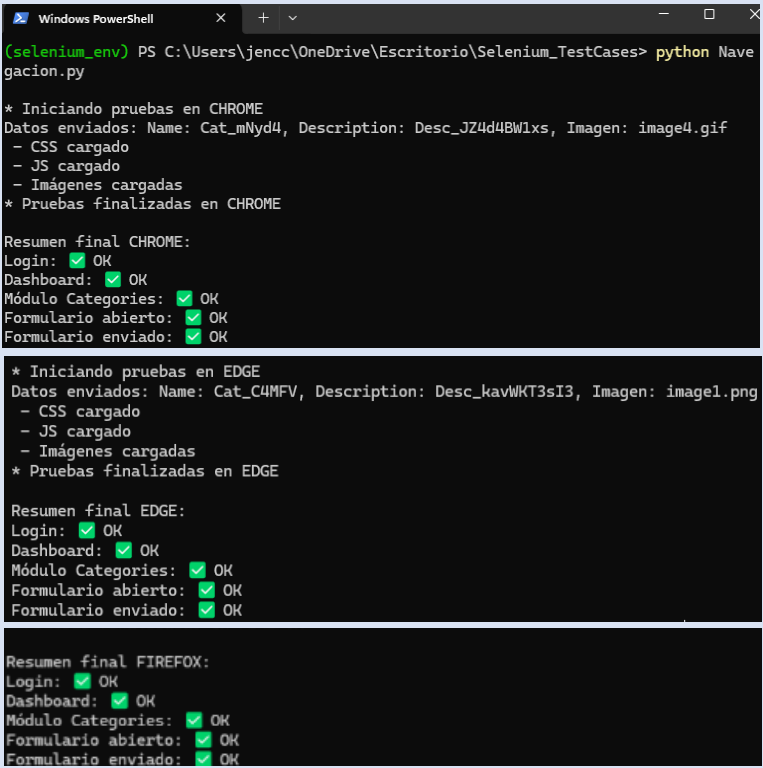
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP53 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Compatibilidad con navegadores | Prioridad | Media |
| Ciclo | Sprint 3 | Estado | Aprobado |
| Propósito del caso | Validar que el sistema funciona correctamente en los diferentes navegadores en sus versiones recientes | | |
| Módulo/Componente | UI | | |
| Condiciones de prueba | Acceso al sistema y acceso a diferentes navegadores (Chrome, Edge y Firefox). | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Login en el sistema. 2. Abrir el módulo a probar para la compatibilidad 3. Abrir formulario de creación de categoría. 4. Ingresar datos. 5. Enviar formulario. 6. Validar que CSS, JS e imágenes se cargan correctamente. 7. Repetir en Chrome, Edge y Firefox | | |
| Resultados esperados | Login y dashboard cargado. Formulario abierto y cargado sin errores. Recursos (CSS, JS, imágenes) cargados. | | |
| Resultado obtenido | Prueba ejecutada correctamente en Chrome, Edge y Firefox. Datos enviado y CSS, JS e imágenes cargadas correctamente. | | |
| Observaciones | N/A | | |
| Evidencia |  <pre> Windows PowerShell (selenium_env) PS C:\Users\jence\OneDrive\Escritorio\Selenium_TestCases> python Navegacion.py * Iniciando pruebas en CHROME Datos enviados: Name: Cat_mNyd4, Description: Desc_JZ4d4BW1xs, Imagen: image4.gif - CSS cargado - JS cargado - Imágenes cargadas * Pruebas finalizadas en CHROME Resumen final CHROME: Login: ✓ OK Dashboard: ✓ OK Módulo Categories: ✓ OK Formulario abierto: ✓ OK Formulario enviado: ✓ OK * Iniciando pruebas en EDGE Datos enviados: Name: Cat_C4MFV, Description: Desc_kavWKT3sI3, Imagen: image1.png - CSS cargado - JS cargado - Imágenes cargadas * Pruebas finalizadas en EDGE Resumen final EDGE: Login: ✓ OK Dashboard: ✓ OK Módulo Categories: ✓ OK Formulario abierto: ✓ OK Formulario enviado: ✓ OK Resumen final FIREFOX: Login: ✓ OK Dashboard: ✓ OK Módulo Categories: ✓ OK Formulario abierto: ✓ OK Formulario enviado: ✓ OK </pre> | | |

Tabla 100. Caso de prueba 53.

| Identificador del caso | CP54 | Ejecutado por | Mauricio Hernández | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------------|---|----------------------|--------------------|-----------------------------------|---------|--|--|--|--|-----------------|------|-------|-------|----------|---------|-----|---|-----|-----|---|---|---------------|---|---|---|---|---|------------|---|-----|-----|-----|----|
| Nombre del caso | Revisión de código seguro | Prioridad | Alta | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 3 | Estado | Fallido | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Confirmar que se aplican prácticas seguras de programación, minimizando vulnerabilidades comunes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Seguridad | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | 1. Se debe tener acceso al código del proyecto. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | 1. Tener acceso al repositorio/carpeta del sistema 2. Contar con la extensión de generación de reportes de SonarQube 3. Ejecutar análisis al código del proyecto. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | El reporte final no debe mostrar vulnerabilidades críticas y el código debe cumplir con los estándares de seguridad definidos, garantizando estabilidad y mantenibilidad. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | El reporte final muestra 412 issues críticos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | Es urgente la refactorización de código el análisis del código revela un sistema cuya estructura interna presenta serios problemas que afectan directamente su calidad, estabilidad y capacidad de evolucionar de forma segura. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia | <table border="1"> <thead> <tr> <th colspan="6">ISSUES COUNT BY SEVERITY AND TYPE</th> </tr> <tr> <th>Type / Severity</th> <th>INFO</th> <th>MINOR</th> <th>MAJOR</th> <th>CRITICAL</th> <th>BLOCKER</th> </tr> </thead> <tbody> <tr> <td>BUG</td> <td>0</td> <td>165</td> <td>129</td> <td>0</td> <td>0</td> </tr> <tr> <td>VULNERABILITY</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>CODE_SMELL</td> <td>2</td> <td>287</td> <td>880</td> <td>412</td> <td>14</td> </tr> </tbody> </table> | | | ISSUES COUNT BY SEVERITY AND TYPE | | | | | | Type / Severity | INFO | MINOR | MAJOR | CRITICAL | BLOCKER | BUG | 0 | 165 | 129 | 0 | 0 | VULNERABILITY | 0 | 0 | 0 | 0 | 1 | CODE_SMELL | 2 | 287 | 880 | 412 | 14 |
| ISSUES COUNT BY SEVERITY AND TYPE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type / Severity | INFO | MINOR | MAJOR | CRITICAL | BLOCKER | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BUG | 0 | 165 | 129 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VULNERABILITY | 0 | 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CODE_SMELL | 2 | 287 | 880 | 412 | 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 101. Caso de prueba 54.

| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP55 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Validación de certificado SSL/TLS | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobado |
| Propósito del caso | Verificar que el certificado SSL/TLS del sitio web sea válido, esté correctamente configurado y utilice protocolos seguros | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | Se debe tener arriba el servidor | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Acceder al sitio web 2. Validar el certificado SSL/TLS mediante herramientas de inspección. 3. Revisar la vigencia del certificado. 4. Confirmar que no se generen advertencias ni errores de seguridad durante la conexión. | | |
| Resultados esperados | El certificado es válido, está vigente, la conexión utiliza protocolos y cifrados seguros sin advertencias. | | |
| Resultado obtenido | El certificado es válido, está vigente, la conexión utiliza protocolos y cifrados seguros sin advertencias | | |
| Observaciones | N/A | | |

Evidencia

Qualys. SSL Labs

Home Projects Qualys Free Trial Contact

You are here: Home > Projects > SSL_Server_Test > cuscatec.cuscatec.com

SSL Report: cuscatec.cuscatec.com

Assessed on: Thu, 23 Oct 2025 12:27:52 UTC | Hide | Clear cache

[Scan Another >>](#)

| | Server | Test time | Grade |
|---|---|---|-------|
| 1 | 212.85.28.193 Ready | Thu, 23 Oct 2025 12:24:32 UTC Duration: 97.883 sec | A |
| 2 | 2a02:4780:1:549:0:77c:3469:5 Ready | Thu, 23 Oct 2025 12:26:10 UTC Duration: 102.64 sec | A |

SSL Report v2.4.1

Copyright © 2009-2025 Qualys, Inc. All Rights Reserved. [Privacy Policy](#)

[Try Qualys for free!](#) Experience the award-winning [Qualys Cloud Platform](#) and the entire collection of [Qualys Cloud Apps](#), including [certificate security](#) solutions. [Terms and Conditions](#)

Tabla 102. Caso de prueba 55.

| Identificador del caso | CP57 | Ejecutado por | Jennifer Cruz | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|--|----------------------|---------------|------------------|------------------------------------|-------------------------------|--------------|--------|------------------------------------|-------------------------------|--------------|--------|---|----------|-----------|--|------------------|------------|----------|-------|--|---|--------|-----------|--|------------------|------------|----------|--|--|
| Nombre del caso | Protección contra fuerza bruta | Prioridad | Alta | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 3 | Estado | Fallido | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Verificar que el módulo de login implemente mecanismos efectivos contra ataques de fuerza bruta.y presencia de CAPTCHA o mecanismos similares. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Módulo de autenticación/login | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Acceso a entorno de pruebas. Herramientas OWASP ZAP para fuzzing. Conjunto de contraseñas para testear. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ejecutar Fuzzer en ZAP /login con payloads de prueba (12–15 contraseñas por sesión). 2. Registrar timestamps y respuestas del servidor. 3. Inspeccionar Responses 3. Ejecutar Fuzzer OWASP ZAP con múltiples contraseñas 4. Realizar intentos manuales consecutivos para confirmar bloqueo temporal. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | El sistema debe activar mecanismos de defensa ante múltiples intentos fallidos: bloqueo temporal, CAPTCHA, rate limit, etc. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | <p>Pruebas manuales mostraron bloqueo temporal al superar el umbral de intentos con el mensaje: "Too many login attempts. Please try again in X seconds." pero el tiempo de espera no escala progresivamente. Pruebas automáticas con Fuzzer no activaron defensa visible. Todas las respuestas HTTP fueron 302 Found. No se detectó CAPTCHA en ninguno de los intentos.</p> <p>Mensaje de error mostrado y capturado en el body HTTP: These credentials do not match our records.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | La defensa contra fuerza bruta está parcialmente implementada. Utiliza un bloqueo donde el tiempo de espera cambia, pero sin un patrón progresivo acumulativo. Funciona en sesiones manuales, pero no se activa ante ataques automatizados sin estado. Dado que el sistema solo bloquea por sesión/cookie, podría ser evadido con ataques sin estado (no mantiene cookies, tokens, variables de sesión). Se sugiere fortalecer la protección con mecanismos como backoff progresivo, bloqueo por IP/usuario y CAPTCHA o 2FA tras múltiples intentos. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia | <p>The screenshot shows the OWASP ZAP interface with a 302 Found response. The response headers are visible, including 'Location: https://cuscat.catec.com/login' and a 'set-cookie' header for 'XSRF-TOKEN'. The status bar at the bottom indicates 'Errores: 0' and 'Fuzzers actuales: 0'. A table below the interface shows the response details:</p> <table border="1"> <thead> <tr> <th>Ítem</th> <th>Tipo de mensaje</th> <th>Código</th> <th>Razón</th> <th>RTT</th> <th>Tamaño de la Cabecera de Respuesta</th> <th>Respuesta (Tamaño del cuerpo)</th> <th>Alerta mayor</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Original</td> <td>302 Found</td> <td></td> <td>625millisegun...</td> <td>1,511bytes</td> <td>386bytes</td> <td>Medio</td> <td></td> </tr> <tr> <td>1</td> <td>Fuzzed</td> <td>302 Found</td> <td></td> <td>328millisegun...</td> <td>1,333bytes</td> <td>386bytes</td> <td></td> <td></td> </tr> </tbody> </table> | | | Ítem | Tipo de mensaje | Código | Razón | RTT | Tamaño de la Cabecera de Respuesta | Respuesta (Tamaño del cuerpo) | Alerta mayor | Estado | 0 | Original | 302 Found | | 625millisegun... | 1,511bytes | 386bytes | Medio | | 1 | Fuzzed | 302 Found | | 328millisegun... | 1,333bytes | 386bytes | | |
| Ítem | Tipo de mensaje | Código | Razón | RTT | Tamaño de la Cabecera de Respuesta | Respuesta (Tamaño del cuerpo) | Alerta mayor | Estado | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Original | 302 Found | | 625millisegun... | 1,511bytes | 386bytes | Medio | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Fuzzed | 302 Found | | 328millisegun... | 1,333bytes | 386bytes | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 104. Caso de prueba 57.

| | | | |
|--------------------------------|--|----------------------|-------------------|
| Identificador del caso | CP58 | Ejecutado por | Nathaly Rodríguez |
| Nombre del caso | Detección de credenciales predeterminadas/débiles | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Fallida |
| Propósito del caso | Intentar crear usuario con credenciales comunes o vacías. | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | Acceso a la creación o modificación de usuarios. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> Ingresar contraseñas débiles como “admin” o “123456” Guardar usuario | | |
| Resultados esperados | Acceso denegado, mensaje genérico. | | |
| Resultado obtenido | El sistema permite la creación de usuarios con contraseñas que no cumplen con estándares mínimos de seguridad, como longitud, complejidad o uso de caracteres especiales. Esto representa una vulnerabilidad que puede facilitar ataques de fuerza bruta o acceso no autorizado. | | |
| Observaciones | El servidor respondió con redirección (302 Found), lo cual indica que las contraseñas fueron aceptadas sin validación. | | |

Evidencia

The screenshot shows a web browser window with the address bar displaying `https://cusatec.cusatec.com/users`. The browser's developer tools are open, showing the network tab with a single request. The request is a POST method to `https://cusatec.cusatec.com/users`. The response is a 302 Found status with a Location header pointing to `https://cusatec.cusatec.com/users/123456`. The browser's address bar shows the response status as 302 Found and the Location as `https://cusatec.cusatec.com/users/123456`. The browser's cookies are visible, showing a session cookie and a CSRF token.

| ID | Fuente | Fecha (Tiempo) | Método | URL | Código | Razón | RTT | Respuesta (Tamaño del cuerpo) | Alerta mayor | Nota | Etiquetas |
|-----|--------|-------------------|--------|-----------------------------------|--------|--------------------|-------|-------------------------------|--------------|------|-----------|
| 90 | Proxy | 17/11/25 21:08:11 | GET | https://cusatec.cusatec.com/users | 200 | OK | 471ms | 60,164 bytes | | | |
| 93 | Proxy | 17/11/25 21:09:04 | POST | https://cusatec.cusatec.com/users | 302 | Found | 304ms | 41 bytes | | | |
| 95 | Proxy | 17/11/25 21:09:23 | POST | https://cusatec.cusatec.com/users | 302 | Found | 373ms | 38 bytes | | | |
| 96 | Proxy | 17/11/25 21:09:23 | GET | https://cusatec.cusatec.com/users | 200 | OK | 244ms | 65,679 bytes | | | |
| 97 | Proxy | 17/11/25 21:12:39 | GET | https://cusatec.cusatec.com/users | 500 | Internal Server E. | 1.98s | 1,218,443 bytes | | | |
| 98 | Manual | 17/11/25 21:27:23 | POST | https://cusatec.cusatec.com/users | 302 | Found | 1.22s | 41 bytes | | | |
| 101 | Manual | 17/11/25 21:28:18 | POST | https://cusatec.cusatec.com/users | 302 | Found | 420ms | 41 bytes | | | |

Tabla 105. Caso de prueba 58

| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP59 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Enumeración de cuentas | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobado |
| Propósito del caso | Verificar si el sistema revela la existencia de cuentas de usuario mediante mensajes de error diferenciados durante el proceso de autenticación. | | |
| Módulo/Componente | Login / Formulario de autenticación | | |
| Condiciones de prueba | Acceso al formulario de login. Proxy configurado en el navegador Se accede al formulario de login como credenciales incorrectas o campos vacíos. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Capturar la petición del formulario de login. 2. Enviar intentos manuales de login con credenciales invalidas o campos vacíos. 3. Ejecutar Fuzzer en el parámetro email/password usando payload. 4. Comparar mensajes en el body, códigos HTTP, tamaños de respuesta, headers y tiempos entre los distintos intentos. | | |
| Resultados esperados | Mensaje genérico ante credenciales inválidas que no indique existencia del usuario. Códigos HTTP consistentes para fallos de autenticación (no devolver 404 específico para usuario no existente ni otros códigos diferenciadores). No diferencias observables en headers/cookies que permitan inferir existencia. | | |
| Resultado obtenido | Mensaje de error mostrado y capturado en el body HTTP: These credentials do not match our records. Resultado del Fuzzer: todas las peticiones devolvieron HTTP 302 Found y el tamaño del cuerpo de respuesta fue consistente (386 bytes) para todos los intentos. Prueba con campo password vacío retornó HTTP 200 OK y mostró validación de campo: The password field is required. No se detectaron diferencias en cabeceras relevantes ni emisión de cookie de sesión en respuestas de fallo que permitan identificar cuentas. | | |
| Observaciones | Asegurar mecanismos contra fuerza bruta (bloqueo por IP/usuario, CAPTCHA o 2FA). | | |
| Evidencia | <p>The screenshot shows two HTTP responses from a browser's developer tools. The top response is a 200 OK status with a body containing HTML code for a password validation error: <code>The password field is required.</code>. The bottom response is also a 200 OK status with a body containing HTML code for a login form, including a message: <code>These credentials do not match our records.</code>.</p> | | |

Tabla 106. Caso de prueba 59.

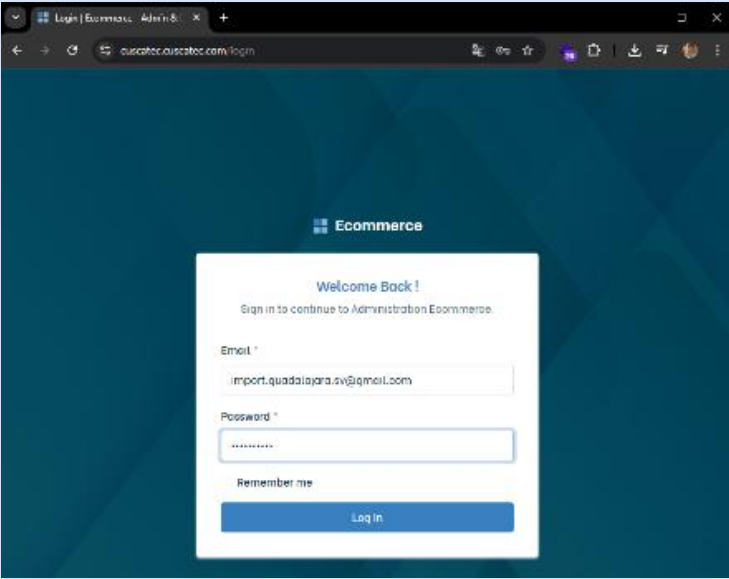
| | | | |
|--------------------------------|---|----------------------|-------------------------|
| Identificador del caso | CP60 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Validación de “Remember Me” | Prioridad | Media |
| Ciclo | Sprint 3 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Evaluar si la opción “recordarme” conserva sesiones de forma segura | | |
| Módulo/ Componente | Seguridad | | |
| Condiciones de prueba | El usuario no debe estar con la sesión iniciada al momento de la prueba, deben estar vacías las cajas de texto y el checkbox de “remember” | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Abrir la URL de login 2. Llenar las cajas de texto de “Email” y “Password” con las credenciales de un usuario válido. 3. Dar click en el checkbox “Remember” 4. Una vez iniciada la sesión, cerrar la pestaña del navegador 5. Abrir una nueva pestaña en el navegador 5. Ingresar a cualquier URL del sistema y verificar que ya está la sesión iniciada | | |
| Resultados esperados | La sesión debe mantenerse activa únicamente si el usuario seleccionó la opción “Remember Me”. | | |
| Resultado obtenido | Si mantiene la sesión activa durante horas, aunque se cierre el navegador, pero es independiente sobre si se presiona el botón "Remember me" | | |
| Observaciones | Independientemente si se presiona o no el botón la sesión queda guardada por un lapso de unas horas | | |
| Evidencia |  | | |

Tabla 107. Caso de prueba 60.

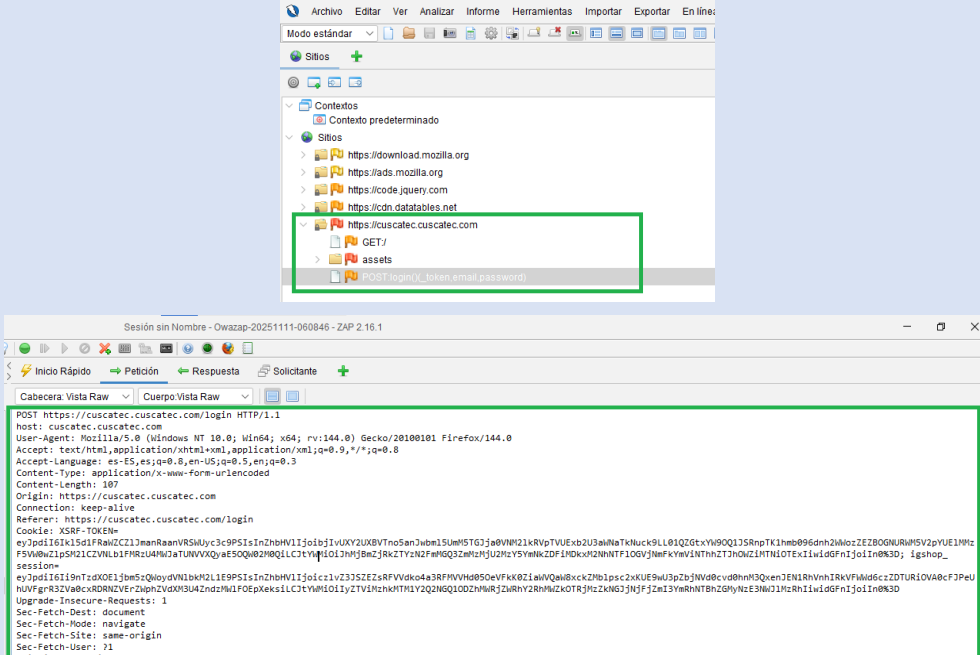
| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP61 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Gestión segura de sesiones | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Fallido |
| Propósito del caso | Verificar atributos de cookies, expiración y control de sesiones concurrentes | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | Servidor debe encontrarse activo No debe estar iniciado en el sistema o tener la sesión activa por cuestiones de cookies | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Se modifica el proxy para que las peticiones pasen por OWASP ZAP 2. Se abre en el navegador la URL de login 3. Se colocan las credenciales en sus cajas de textos correspondientes 4. Se da click en el botón de login 5. Se verifica la información capturada en OWASP ZAP en la sección correspondiente al POST y lo enviado en la requests | | |
| Resultados esperados | El sistema debe generar cookies seguras, expirar la sesión correctamente tras el tiempo definido y controlar la concurrencia. | | |
| Resultado obtenido | Durante la ejecución de la prueba se verificó que las credenciales de acceso fueron enviadas en texto plano dentro de la petición interceptada, lo que expone directamente la información sensible. | | |
| Observaciones | | | |
| Evidencia |  | | |

Tabla 108. Caso de prueba 61.

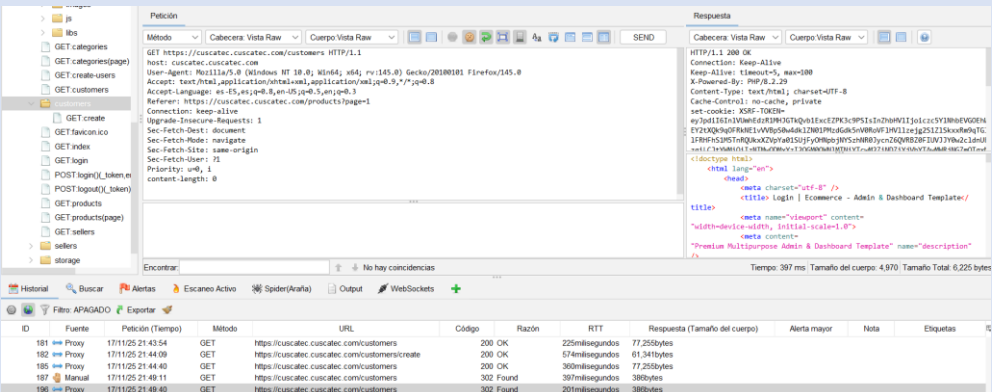
| Identificador del caso | CP62 | Ejecutado por | Nathaly Rodríguez | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|--|----------------------|-------------------|--|--------|-------------------|-----------------|-------------------------------|--------------|-------|-----------|-------------------------------|--------------|------|-----------|-----|-------|------------------|-----|---------------------------------------|-----|----|-----------------|------------|--|--|--|-----|-------|------------------|-----|--|-----|----|-----------------|------------|--|--|--|-----|-------|------------------|-----|---------------------------------------|-----|----|-----------------|------------|--|--|--|-----|--------|------------------|-----|---------------------------------------|-----|-------|-----------------|----------|--|--|--|-----|-------|------------------|-----|---------------------------------------|-----|-------|-----------------|----------|--|--|--|
| Nombre del caso | Bypass de autenticación | Prioridad | Alta | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ciclo | Sprint 3 | Estado | Aprobada | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Propósito del caso | Intentar saltarse el login o manipular parámetros. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Módulo/Componente | Seguridad | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones de prueba | Usuario deslogueado. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Procedimiento de prueba | 1. Manipular parámetros. 2. Intentar acceso directo a rutas protegidas. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultados esperados | No permite saltarse el login. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Resultado obtenido | El servidor respondió con 302 Found, redireccionando al login, lo cual indica que bloqueó el acceso por falta de sesión. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones | N/A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evidencia |  <p>The screenshot shows a network request in a browser's developer tools. The request is a GET to https://cusatec.cusatec.com/customers. The response is a 302 Found status with a Location header pointing to the login page. The response body contains HTML meta tags for the login page.</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Fuente</th> <th>Petición (Tiempo)</th> <th>Método</th> <th>URL</th> <th>Código</th> <th>Razón</th> <th>RTT</th> <th>Respuesta (Tamaño del cuerpo)</th> <th>Alerta mayor</th> <th>Nota</th> <th>Etiquetas</th> </tr> </thead> <tbody> <tr> <td>181</td> <td>Proxy</td> <td>1711:25:21:43:54</td> <td>GET</td> <td>https://cusatec.cusatec.com/customers</td> <td>200</td> <td>OK</td> <td>225milisegundos</td> <td>77,25bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>182</td> <td>Proxy</td> <td>1711:25:21:44:09</td> <td>GET</td> <td>https://cusatec.cusatec.com/customers/create</td> <td>200</td> <td>OK</td> <td>574milisegundos</td> <td>61,34bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>185</td> <td>Proxy</td> <td>1711:25:21:44:40</td> <td>GET</td> <td>https://cusatec.cusatec.com/customers</td> <td>200</td> <td>OK</td> <td>360milisegundos</td> <td>77,25bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>187</td> <td>Manual</td> <td>1711:25:21:48:11</td> <td>GET</td> <td>https://cusatec.cusatec.com/customers</td> <td>302</td> <td>Found</td> <td>397milisegundos</td> <td>380bytes</td> <td></td> <td></td> <td></td> </tr> <tr> <td>196</td> <td>Proxy</td> <td>1711:25:21:48:40</td> <td>GET</td> <td>https://cusatec.cusatec.com/customers</td> <td>302</td> <td>Found</td> <td>201milisegundos</td> <td>380bytes</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> | | | ID | Fuente | Petición (Tiempo) | Método | URL | Código | Razón | RTT | Respuesta (Tamaño del cuerpo) | Alerta mayor | Nota | Etiquetas | 181 | Proxy | 1711:25:21:43:54 | GET | https://cusatec.cusatec.com/customers | 200 | OK | 225milisegundos | 77,25bytes | | | | 182 | Proxy | 1711:25:21:44:09 | GET | https://cusatec.cusatec.com/customers/create | 200 | OK | 574milisegundos | 61,34bytes | | | | 185 | Proxy | 1711:25:21:44:40 | GET | https://cusatec.cusatec.com/customers | 200 | OK | 360milisegundos | 77,25bytes | | | | 187 | Manual | 1711:25:21:48:11 | GET | https://cusatec.cusatec.com/customers | 302 | Found | 397milisegundos | 380bytes | | | | 196 | Proxy | 1711:25:21:48:40 | GET | https://cusatec.cusatec.com/customers | 302 | Found | 201milisegundos | 380bytes | | | |
| ID | Fuente | Petición (Tiempo) | Método | URL | Código | Razón | RTT | Respuesta (Tamaño del cuerpo) | Alerta mayor | Nota | Etiquetas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 181 | Proxy | 1711:25:21:43:54 | GET | https://cusatec.cusatec.com/customers | 200 | OK | 225milisegundos | 77,25bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 182 | Proxy | 1711:25:21:44:09 | GET | https://cusatec.cusatec.com/customers/create | 200 | OK | 574milisegundos | 61,34bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 185 | Proxy | 1711:25:21:44:40 | GET | https://cusatec.cusatec.com/customers | 200 | OK | 360milisegundos | 77,25bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 187 | Manual | 1711:25:21:48:11 | GET | https://cusatec.cusatec.com/customers | 302 | Found | 397milisegundos | 380bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 196 | Proxy | 1711:25:21:48:40 | GET | https://cusatec.cusatec.com/customers | 302 | Found | 201milisegundos | 380bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 109. Caso de prueba 62.


| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP63 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Respuestas ante errores y mensajes | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobado |
| Propósito del caso | Evaluar si los mensajes de error revelan información sensible. | | |
| Módulo/Componente | Seguridad | | |
| Condiciones de prueba | 1-Se debe tener previamente el proxy haciendo referencia a OWASP ZAP 2-Se debe identificar un formulario o login para hacer la prueba. | | |
| Procedimiento de prueba | 1. Provocar errores (login, formularios). 2. Se verifica la información capturada en OWASP ZAP en la sección correspondiente al POST y lo enviado en la respuesta 3. Analizar mensajes | | |
| Resultados esperados | El sistema debía mostrar mensajes de error simples, sin incluir información sensible ni detalles internos en la respuesta o en el código HTML. | | |
| Resultado obtenido | Durante la ejecución de la prueba, los mensajes de error se presentaron correctamente y no revelaron información sensible en la respuesta ni en el HTML. | | |
| Observaciones | N/A | | |
| Evidencia |  <p>The screenshot displays two parts of a web page. The top part is an HTTP response from a browser's developer tools, showing a 200 OK status and various headers including 'Connection: Keep-Alive', 'Cache-Control: no-cache, private', and 'set-cookie: XSRF-TOKEN=...'. The bottom part is a snippet of HTML code for a login form. It includes a form with fields for 'email' and 'password'. A message is displayed: 'These credentials do not match our records.' The message is enclosed in a red box in the original image.</p> | | |

Tabla 110. Caso de prueba 63.

| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP64 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Pruebas de UI/UX y resiliencia | Prioridad | Alta |
| Ciclo | Sprint 3 | Estado | Aprobado |
| Propósito del caso | Verificar funcionamiento de CAPTCHA, enlaces, validaciones, y responsive. | | |
| Módulo/Componente | UI/UX | | |
| Condiciones de prueba | 1. Se debe entrar como un usuario administrador | | |
| Procedimiento de prueba | Se recorren todos los enlaces y se identifica que no exista un quiebre o error al cargar la página que sigue el enlace. Se almacena el éxito o el fallo al ingresar a las secciones. Y se imprime en pantalla cuales han sido exitosos y cuáles han sido fallidos | | |
| Resultados esperados | La interfaz debía mostrarse correctamente, con mensajes claros, enlaces funcionando sin errores y un diseño adaptable a distintos dispositivos. | | |
| Resultado obtenido | La interfaz se mostró de forma correcta y los enlaces funcionaron adecuadamente. | | |
| Observaciones | N/A | | |
| Evidencia |  <p>The evidence consists of two parts. The top part is a terminal window showing a series of test results for various pages, all marked as successful. The bottom part is an email notification from 'CUSCATEC TEST' containing the same test results.</p> <pre> print(status_result) Problems Output Debug Console Terminal Ports Azure ✓ Se ingresó exitosamente a la página 'States List' ✓ Se ingresó exitosamente a la página 'Products' ✓ Se ingresó exitosamente a la página 'Orders' ✓ Se ingresó exitosamente a la página 'Warehouses' ✓ Se ingresó exitosamente a la página 'Inventory by Warehouse' ✓ Se ingresó exitosamente a la página 'Categories' ✓ Se ingresó exitosamente a la página 'Suppliers' ✓ Se ingresó exitosamente a la página 'Purchase Orders' ✓ Se ingresó exitosamente a la página 'Coupons List' ✓ Se ingresó exitosamente a la página 'Users List' 280 {'Messages': [{'Status': 'success', 'CustomID': '', 'To': [{'Email': 'mauricioricaldone14@gmail.com', 'MessageUUID': '10884bf1-6764-4190-a926-ssage/288230409830550819'}], 'Cc': [], 'Bcc': []}]}</pre> <p>CP64 - Verificar funcionamiento de CAPTCHA, enlaces, validaciones, y responsive. Recibidos</p> <p>CUSCATEC TEST a través de bnc3.mailjet.com 12:01 a.m. (hace 4 minutos) ☆ ☺ ↶ ⋮ para mí</p> <p>Parece que este mensaje está en inglés X Traducir al español</p> <p>CP64 - Verificar funcionamiento de CAPTCHA, enlaces, validaciones, y responsive. ✓ Se ingresó exitosamente a la página 'Customers List' ✓ Se ingresó exitosamente a la página 'Sellers List' ✓ Se ingresó exitosamente a la página 'States List' ✓ Se ingresó exitosamente a la página 'Products' ✓ Se ingresó exitosamente a la página 'Orders' ✓ Se ingresó exitosamente a la página 'Warehouses' ✓ Se ingresó exitosamente a la página 'Inventory by Warehouse' ✓ Se ingresó exitosamente a la página 'Categories' ✓ Se ingresó exitosamente a la página 'Suppliers' ✓ Se ingresó exitosamente a la página 'Purchase Orders' ✓ Se ingresó exitosamente a la página 'Coupons List' ✓ Se ingresó exitosamente a la página 'Users List'</p> | | |

Tabla 111. Caso de prueba 64.

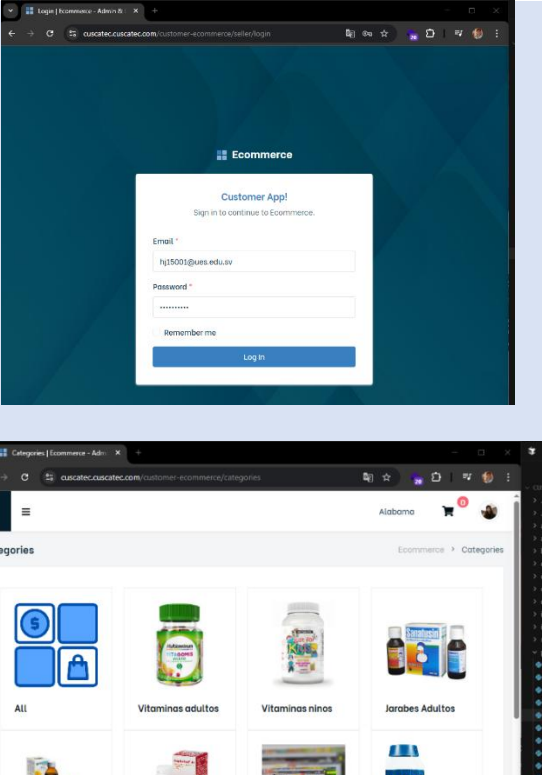
| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP65 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Vista inicial de categorías | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que al iniciar sesión el usuario sea redirigido a la pantalla de categorías | | |
| Módulo/Componente | Purchase | | |
| Condiciones de prueba | El usuario no debe estar logueado en el sistema.sz | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1.Abrir la URL de login 2.Llenar las cajas de texto de “Email” y “Password” con las credenciales de un usuario valido. 3.La vista que cargue debe ser la de categorias | | |
| Resultados esperados | El sistema debe redirigir al usuario autenticado directamente a la pantalla de categorías, mostrando de manera clara y organizada todas las categorías disponibles. | | |
| Resultado obtenido | El sistema redirigió correctamente al usuario a la pantalla de categorías, mostrando todas las categorías disponibles de forma completa y sin errores de visualización. El comportamiento observado coincide con lo esperado. | | |
| Observaciones | N/A | | |
| Evidencia |  | | |

Tabla 112. Caso de prueba 65.

| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP66 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Vista inicial de categorías | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Visualización de productos por categoría | | |
| Módulo/Componente | Purchase | | |
| Condiciones de prueba | <ol style="list-style-type: none"> 1. El usuario ejecutor debe ser de tipo Customer y debe estar dentro del sistema 2. Debe estar en el inicio del usuario que es la vista de todas las categorías 3. Debe haber al menos una categoría en la base de datos | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Entrar a la vista de inicio del usuario, la cual es la de categorías. 2. Seleccionar una categoría de la lista de categorías | | |
| Resultados esperados | El sistema debe mostrar los productos asociados a la categoría seleccionada, sin errores de visualización. | | |
| Resultado obtenido | El sistema mostró correctamente los productos correspondientes a la categoría seleccionada, cumpliendo con la funcionalidad esperada y sin presentar inconsistencias en la visualización. | | |
| Observaciones | N/A | | |

Evidencia

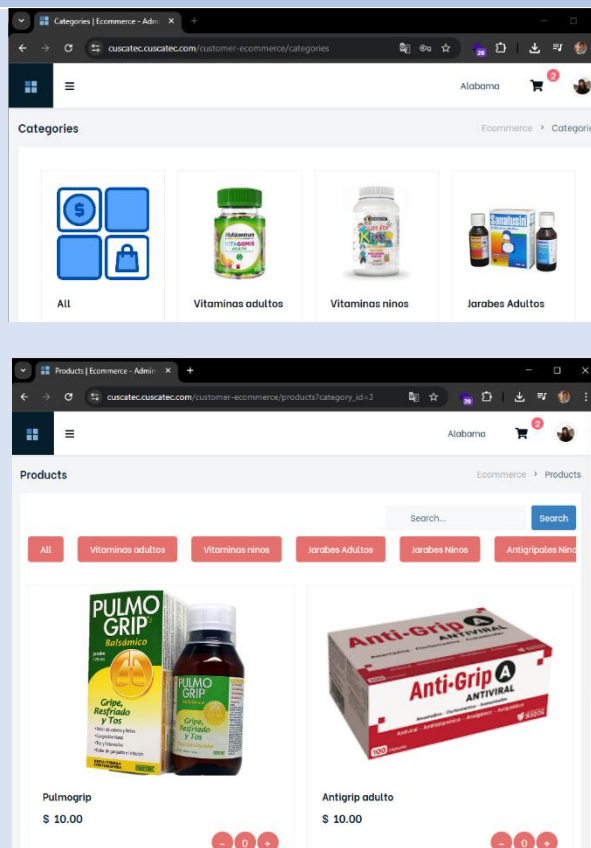


Tabla 113. Caso de prueba 66.

| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP67 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Gestión de productos en carrito | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar agregar y remover productos desde la categoría seleccionada | | |
| Módulo/Componente | Purchase | | |
| Condiciones de prueba | El usuario ejecutor debe estar dentro del sistema Deben existir categorías existentes Deben existir productos dentro de esa categoría | | |
| Procedimiento de prueba | 1.Se ingresa al sistema 2.Se selecciona una categoría 3.Dentro de esa categoría, se da click en el botón de agregar producto 4.Se verifica que se agrega el producto 5.Se da click en el botón de restar producto 6.Se verifica que se quita el producto del carro. | | |
| Resultados esperados | El sistema debe permitir agregar y remover productos del carrito de manera inmediata, reflejando los cambios en la vista del carrito y manteniendo la consistencia de los datos | | |
| Resultado obtenido | El sistema permitió agregar y remover productos del carrito de manera correcta, reflejando los cambios en tiempo real y cumpliendo con la funcionalidad esperada. No se presentaron errores ni inconsistencias. | | |
| Observaciones | N/A | | |

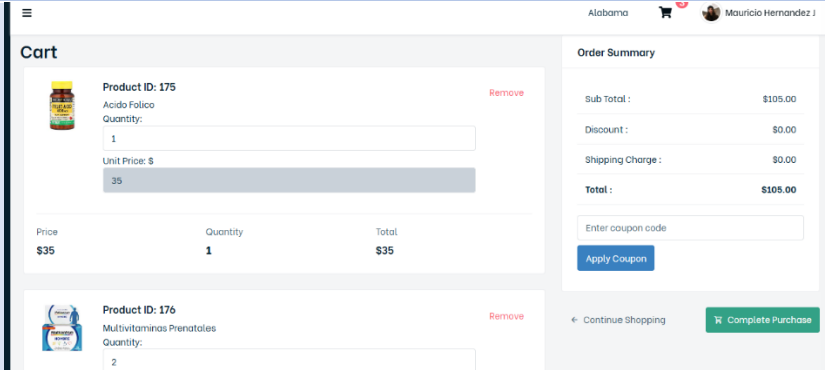
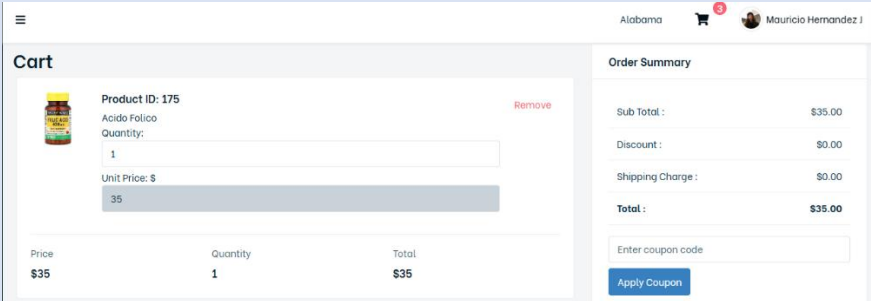
| | |
|------------------|--|
| Evidencia |   |
|------------------|--|

Tabla 114. Caso de prueba 67.

| | | | |
|--------------------------------|--|----------------------|--------------------|
| Identificador del caso | CP68 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Cálculo de totales | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que se muestren los cálculos correctos (subtotal, descuentos, total final) | | |
| Módulo/Componente | Purchase | | |
| Condiciones de prueba | Debe haber productos que agregar al carrito Deben agregarse productos al carrito | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Sumar todos los totales 2. Almacenarlo en una variable 3. Después comparar esta variable con el Subtotal 4. Si son iguales que imprima "subtotal correcto" y luego que tome el valor de descuento se lo reste a la variable, y finalmente que tome el valor del total y compare resultado, si son iguales que imprima que todo el proceso de cálculo | | |
| Resultados esperados | El sistema debía mostrar correctamente los cálculos de subtotal, descuentos y total final, reflejando valores precisos en cada etapa. | | |
| Resultado obtenido | Durante la prueba, los cálculos se mostraron de manera correcta y los valores de subtotal, descuentos y total final coincidieron con lo esperado. | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 115. Caso de prueba 68.

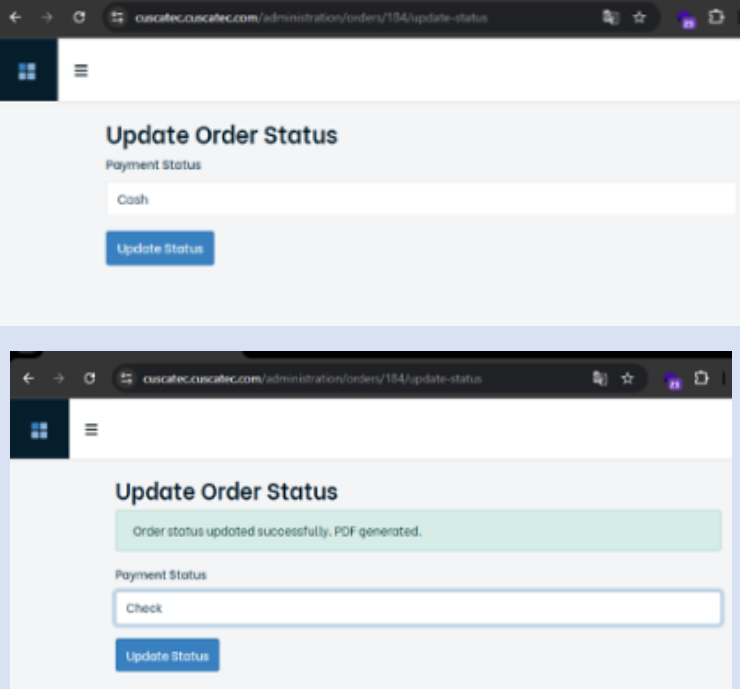
| | | | |
|--------------------------------|---|----------------------|--------------------|
| Identificador del caso | CP69 | Ejecutado por | Mauricio Hernández |
| Nombre del caso | Actualización de estado de pago (Admin) | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que desde la vista Admin se pueda cambiar estado de pago y se refleje en facturación | | |
| Módulo/Componente | Purchase | | |
| Condiciones de prueba | Se debe tener credenciales de administrador Se debe tener ordenes registradas en el sistema | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Se ingresa al sistema. 2. Se navega a las órdenes. 3. Se selecciona la primera orden para cambiar el status del pago. 4. Carga el status de pago y se cambia la opción del select. 5. Se da click en “Save Update”. 6. Se ve que se ve reflejado el cambio del status del pago. | | |
| Resultados esperados | El sistema debía permitir que el administrador cambiara el estado de pago de una orden y que este cambio se reflejara correctamente en la base de datos y en la facturación. | | |
| Resultado obtenido | Durante la prueba, el estado de pago se actualizó de forma correcta y el cambio quedó reflejado tanto en la base de datos como en la facturación. | | |
| Observaciones | N/A | | |
| Evidencia |  | | |

Tabla 116. Caso de prueba 69.

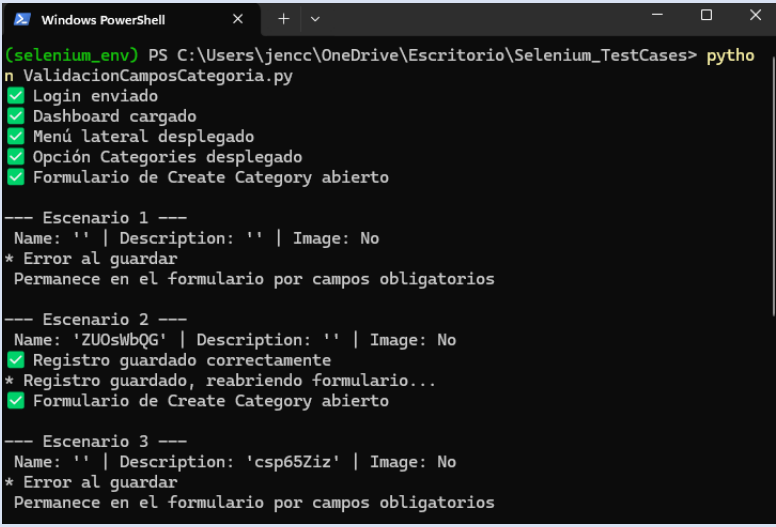
| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP70 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Validación de campos obligatorios | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que el formulario de crear categorías valide correctamente los campos, asegurando que se muestren errores cuando los datos sean inválidos. | | |
| Módulo/Componente | Gestión de categorías | | |
| Condiciones de prueba | <ol style="list-style-type: none"> 1. Ingresar al sistema con un usuario administrador. 2. El formulario de creación de categoría está accesible. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Navegar a la sección de Categories. 2. Hacer clic en “Add New Category”. 3. Observar si se muestran los mensajes de error para campos obligatorios o si el registro se guarda correctamente. 4. Repetir para todos los escenarios definidos. | | |
| Resultados esperados | <p>Si los campos obligatorios están vacíos, se muestran mensajes de error y el formulario permanece abierto.</p> <p>Si los campos obligatorios se completan correctamente, el registro se guarda y el sistema redirige a la lista de categorías.</p> | | |
| Resultado obtenido | <p>Coincide con los resultados esperados:</p> <ul style="list-style-type: none"> - Escenarios con campos vacíos muestran errores inmediatamente. - El campo "Name" es obligatorio, ya que su ausencia impide el guardado del formulario, independientemente de la presencia o ausencia de los campos "Description" e "Image", los cuales son opcionales. | | |
| Observaciones | Se sugiere mostrar mensajes y señales visuales claras para facilitar la validación y mejorar la trazabilidad del formulario, contribuyendo a una mejor experiencia del usuario. | | |
| Evidencia |  <pre> Windows PowerShell (selenium_env) PS C:\Users\jenc\OneDrive\Escritorio\Selenium_TestCases> python ValidacionCamposCategoria.py [✓] Login enviado [✓] Dashboard cargado [✓] Menú lateral desplegado [✓] Opción Categories desplegado [✓] Formulario de Create Category abierto --- Escenario 1 --- Name: '' Description: '' Image: No * Error al guardar Permanece en el formulario por campos obligatorios --- Escenario 2 --- Name: 'ZUOswbQG' Description: '' Image: No [✓] Registro guardado correctamente * Registro guardado, reabriendo formulario... [✓] Formulario de Create Category abierto --- Escenario 3 --- Name: '' Description: 'csp65Ziz' Image: No * Error al guardar Permanece en el formulario por campos obligatorios </pre> | | |

Tabla 117. Caso de prueba 70.

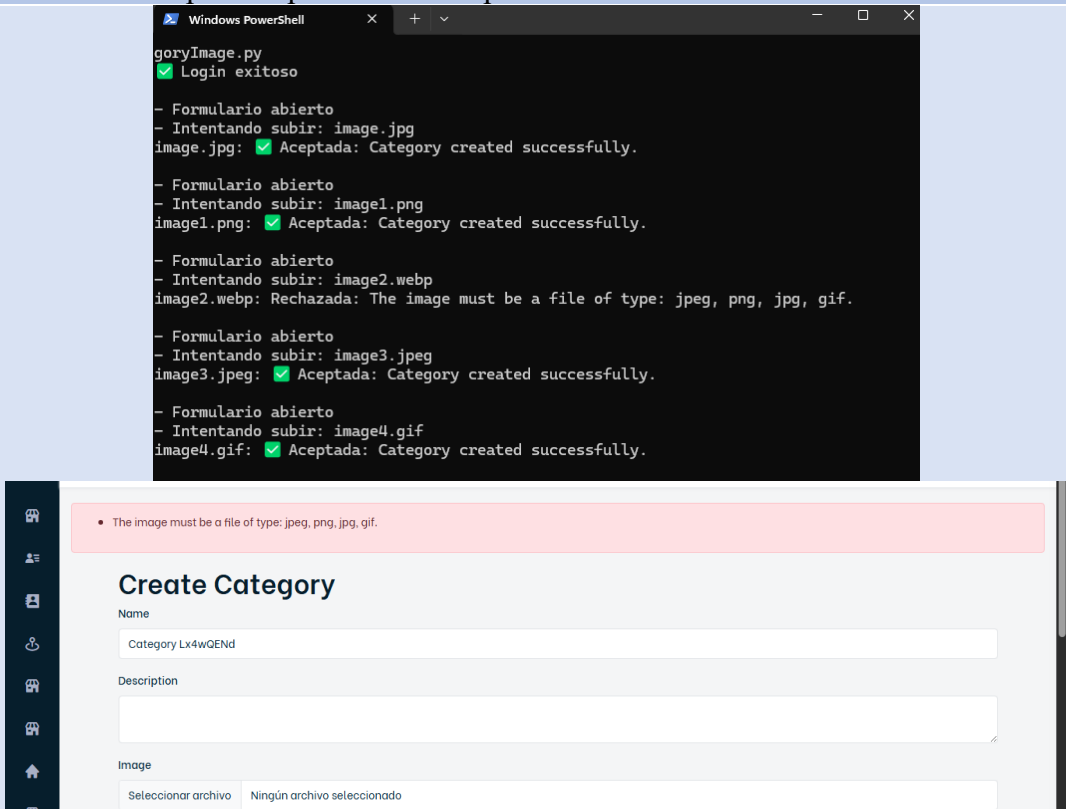
| | | | |
|--------------------------------|--|----------------------|---------------|
| Identificador del caso | CP71 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Validación de carga de imagen | Prioridad | Media |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que el sistema acepte solo formatos válidos de imagen y rechace los no permitidos | | |
| Módulo/Componente | Categorías / Formulario de creación | | |
| Condiciones de prueba | Usuario logueado, formulario de “Add New Category” accesible, imágenes disponibles | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Abrir login y autenticarse. 2. Abrir formulario “Add New Category”. 3. Subir cada imagen disponible. 4. Verificar mensajes de éxito o error. | | |
| Resultados esperados | Categoría guardada correctamente junto con su imagen. Formatos válidos: mensaje de éxito. Formatos no válidos: mensaje de error. | | |
| Resultado obtenido | Categoría guardada correctamente Archivo imagen guardado Mensajes correctamente validados. | | |
| Observaciones | Formato del tipo .webp o .avif no aceptada | | |
| Evidencia |  | | |

Tabla 118. Caso de prueba 71.

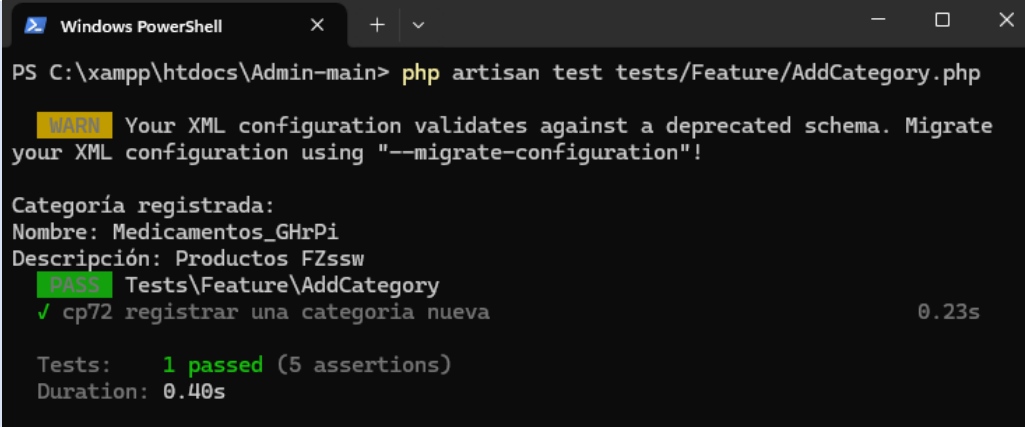
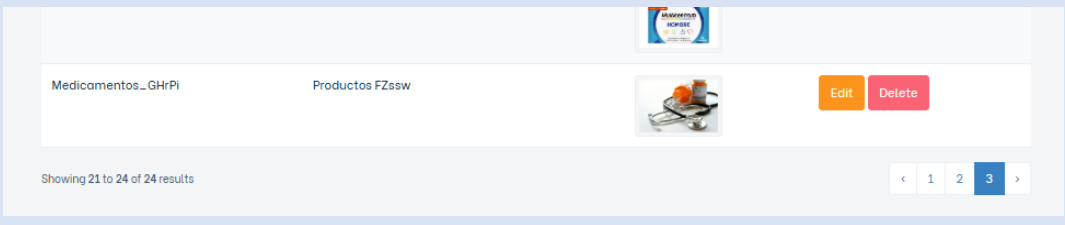
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP72 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Registro exitoso de categoría | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que al crear una categoría con datos válidos se registre correctamente en la base de datos y aparezca en la lista de categorías. | | |
| Módulo/Componente | Category / Administración de categorías | | |
| Condiciones de prueba | Usuario logueado con permisos para crear categorías Formulario de creación accesible | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Ingresar al formulario “Create Category”. 2. Llenar los campos 3. Subir una imagen válida 4. Presionar “Create Category”. 5. Verificar redirección al listado de categorías. | | |
| Resultados esperados | <p>La categoría se guarda en la base de datos.</p> <p>La categoría aparece en la lista de categorías.</p> <p>El campo name, description e imagen se muestran y guardan correctamente.</p> | | |
| Resultado obtenido | La categoría fue registrada correctamente con nombre, descripción e imagen visibles en la lista. | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 119. Caso de prueba 72.

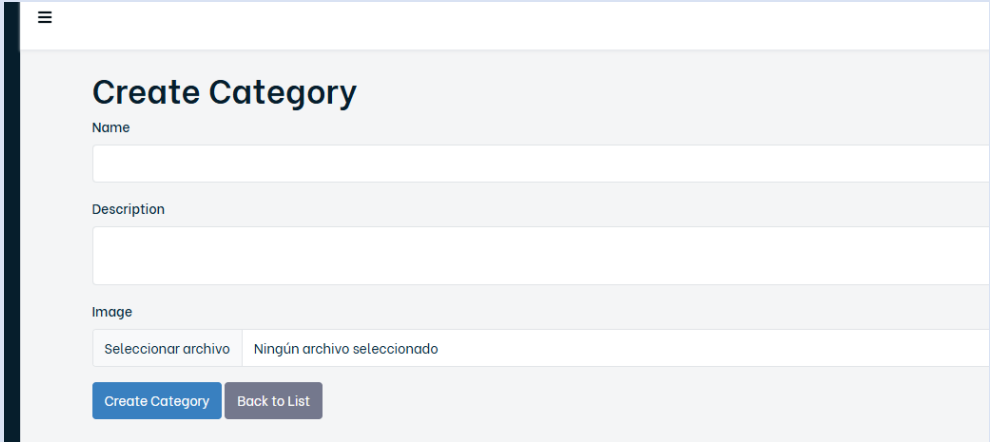
| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP73 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Retorno a lista des de Create Category | Prioridad | Baja |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Verificar que al presionar el botón “Back to List” desde el formulario de creación de categoría, el sistema regrese correctamente a la vista de lista de categorías. | | |
| Módulo/ Componente | Gestión de categorías / Create Category | | |
| Condiciones de prueba | El usuario debe estar logueado en el sistema. La opción de Categories y el botón Add New Category deben estar accesibles. El formulario de creación debe estar abierto y parcialmente completado o vacío | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Navegar a la sección Categories. 2. Hacer clic en Add New Category para abrir el formulario. 3. Sin completar campos o parcialmente completo, presionar el botón Back to List. 4. Verificar que se redirige a la vista lista de categorías (Categories List). | | |
| Resultados esperados | El sistema regresa a la lista de categorías. Los datos existentes en la lista se muestran correctamente. No se presentan mensajes de error. | | |
| Resultado obtenido | Funciona correctamente, la lista se muestra correctamente tras presionar Back to List. | | |
| Observaciones | N/A | | |
| Evidencia |  | | |

Tabla 120. Caso de prueba 73.

| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP74 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Eliminar categoría | Prioridad | Media |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que se pueda eliminar una categoría existente de la lista correctamente y que los cambios se reflejen en el sistema. | | |
| Módulo/Componente | Category / Administración de categorías | | |
| Condiciones de prueba | Usuario logueado con permisos para crear categorías. Debe existir al menos una categoría en la base de datos, con o sin imagen asociada. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Seleccionar una categoría existente. 2. Ejecutar la acción de eliminar. 3. Confirmar la eliminación. 4. Verificar que la categoría ya no esté en la lista ni en la base de datos. | | |
| Resultados esperados | La categoría seleccionada se elimina correctamente. La lista de categorías se actualiza y no muestra la categoría eliminada. | | |
| Resultado obtenido | La categoría fue eliminada exitosamente. BD actualizado | | |
| Observaciones | N/A | | |
| Evidencia |  <pre> Windows PowerShell PS C:\xampp\htdocs\Admin-main> php artisan test tests/Feature/DeleteCategory.php [PASS] Your XML configuration validates against a deprecated schema. Migrate your XML configuration using "--migrate-configuration"! [✓] Eliminación confirmada ID: 15 Nombre: Caramelos Descripción: Caramelos [PASS] Tests\Feature>DeleteCategory [✓] cp74 eliminar categoria existente 0.23s Tests: 1 passed (3 assertions) Duration: 0.39s </pre> | | |

Tabla 121. Caso de prueba 74.

| | | | |
|--------------------------------|---|----------------------|---------------|
| Identificador del caso | CP75 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Editar categoría | Prioridad | Alta |
| Ciclo | Sprint 2 | Estado | Aprobado |
| Propósito del caso | Validar que se pueda editar una categoría existente y que los cambios se reflejen correctamente en la BD y en la vista de lista. | | |
| Módulo/Componente | Category / Administración de categorías | | |
| Condiciones de prueba | Usuario logueado con permisos para crear categorías. Debe existir al menos una categoría en los registros | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Seleccionar una categoría 2. Hacer click en el botón Edit 3. Modificar los campos 4. Click en Update Category | | |
| Resultados esperados | La categoría se actualiza correctamente y se muestran los cambios en la lista, incluyendo la imagen. | | |
| Resultado obtenido | La categoría fue editada correctamente. Los cambios se reflejan correctamente. | | |
| Observaciones | N/A | | |
| Evidencia |   | | |

Tabla 122. Caso de prueba 75.

| | | | |
|--------------------------------|--|----------------------|-------------------------|
| Identificador del caso | CP76 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Agregar un producto a la orden | Prioridad | Alta |
| Ciclo | Sprint 1 | Estado | Ejecutado con hallazgos |
| Propósito del caso | Validar que el sistema permita agregar dinámicamente un producto existente a una orden real, recalculando totales y aplicando descuentos si corresponde. | | |
| Módulo/Componente | Gestión de órdenes / Detalles de orden | | |
| Condiciones de prueba | La orden debe existir en la base de datos. Debe haber al menos un producto con precio definido. | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Autenticarse como usuario administrador 2. Seleccionar una orden existente por ID 3. Seleccionar aleatoriamente un producto con precio y cantidad 3. Calcular subtotal por línea y total final y validar persistencia. | | |
| Resultados esperados | <p>El producto se agrega correctamente a la orden</p> <p>El total se actualiza</p> <p>El total esperado refleja correctamente el nuevo estado</p> | | |
| Resultado obtenido | Todos los pasos ejecutados correctamente. Producto agregado, totales recalculados. El sistema no permite agregar producto, si la orden está vacía. | | |
| Observaciones | Además, el sistema debe permitir regenerar dinámicamente order_details al agregar nuevos productos, evitando bloqueos cuando la orden está vacía. | | |

Evidencia

```

Windows PowerShell
[1864:4892:0920/005526.711:ERROR:google_api\gcm\engine\registration_request.c
c:291] Registration response error message: DEPRECATED_ENDPOINT
Created TensorFlow Lite XNNPACK delegate for CPU.

Orden modificada: #81
Cliente asociado a la orden: Tienda super Sabor
Producto agregado: Ax + Amb Jarabe
Cantidad: 6, Precio unitario: 10.00, Subtotal: 60
Descuento aplicado: 24
Total sin descuento: 175
Total esperado: 151
PASS Tests\Browser\AddProductTest
✓ cp74 agregar producto y guardar descuento desde la vista 13.67s

Tests: 1 passed (5 assertions)
Duration: 14.31s

```

| Order Details | | | | |
|-----------------|----------|-------------|-------|--------------------------|
| Product | Quantity | Final Price | Total | Remove |
| Anodent x4 | 1 | 40.00 | 40.00 | <input type="checkbox"/> |
| Doloultraestres | 1 | 25.00 | 25.00 | <input type="checkbox"/> |
| Naproxeno 550 | 1 | 50.00 | 50.00 | <input type="checkbox"/> |
| Ax + Amb Jarabe | 6 | 10.00 | 60.00 | <input type="checkbox"/> |

Add Product

Customer
Name: Tienda super Sabor - State: Tennessee

Total Price (Before Discount):
\$151.00

Discount Amount:
24.00

Tabla 123. Caso de prueba 76.

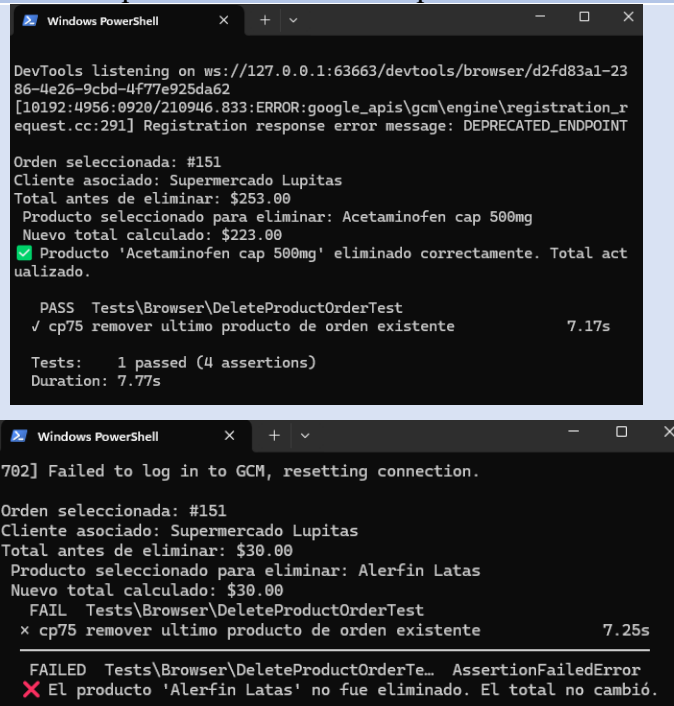
| | | | |
|--------------------------------|---|----------------------|-------------------------|
| Identificador del caso | CP77 | Ejecutado por | Jennifer Cruz |
| Nombre del caso | Remove un producto de una orden | Prioridad | Media |
| Ciclo | Sprint 1 | Estado | Ejecución con hallazgos |
| Propósito del caso | Validar que el sistema permite eliminar un producto específico de una orden sin afectar la integridad de los datos ni generar errores silenciosos. | | |
| Módulo/Componente | Administración de Orders | | |
| Condiciones de prueba | Orden existente con al menos 2 productos Autenticarse como usuario administrador Productos con cantidades y precios definidos | | |
| Procedimiento de prueba | <ol style="list-style-type: none"> 1. Seleccionar orden 2. Seleccionar el producto al azar 3. Marcar checkbox de eliminación y click en "Update Order" 4. Validar que el producto fue eliminado y verificar el total. | | |
| Resultados esperados | Productos seleccionados son eliminados. El total se actualiza correctamente. No se generan errores y la orden permanece activa y editable. | | |
| Resultado obtenido | Productos fueron seleccionados y eliminados correctamente. Al borrar duplicados, se genera error SQL si esta no es la última instancia registrada. Si la orden queda vacía, no permite agregar nuevos productos. | | |
| Observaciones | Se recomienda revisar la lógica de eliminación para que cada instancia de producto sea tratada de forma independiente y no agrupada. Además, el sistema debe permitir regenerar dinámicamente order details al agregar nuevos productos, evitando bloqueos cuando la orden queda vacía. | | |
| Evidencia |  | | |

Tabla 124. Caso de prueba 77.

6.27. Pruebas de código

El análisis del código revela un sistema cuya estructura interna presenta serios problemas que afectan directamente su calidad, estabilidad y capacidad de evolucionar de forma segura. Aunque el sistema está actualmente en producción, el estado del código muestra patrones claros de improvisación, falta de refactorización, ausencia de estándares sólidos y una organización deficiente. Esto lo convierte en un producto técnicamente frágil, donde su funcionamiento actual no es consecuencia de una arquitectura sólida, sino más bien de coincidencia operativa: funciona porque nadie ha tocado las partes críticas o porque los errores aún no se han manifestado de manera visible para los usuarios.

Uno de los elementos más preocupantes es la complejidad interna del sistema. Valores como una complejidad lógica que supera los 700 y una complejidad cognitiva que sobrepasa los 500 son indicadores de módulos que han crecido sin control. Este tipo de complejidad no ocurre por accidente: ocurre porque se dejan expandir funciones o componentes sin modularización y sin una revisión seria del diseño. Para un sistema en producción, este nivel de complejidad significa que el código es difícil de entender, extremadamente delicado de modificar y altamente propenso a errores cuando se requiere mantenimiento o nuevas funcionalidades. En la práctica, cualquier intervención en esas zonas del código tiene altas probabilidades de romper algo, y es probable que esos módulos se hayan convertido en "cajas negras" que los desarrolladores evitan tocar.

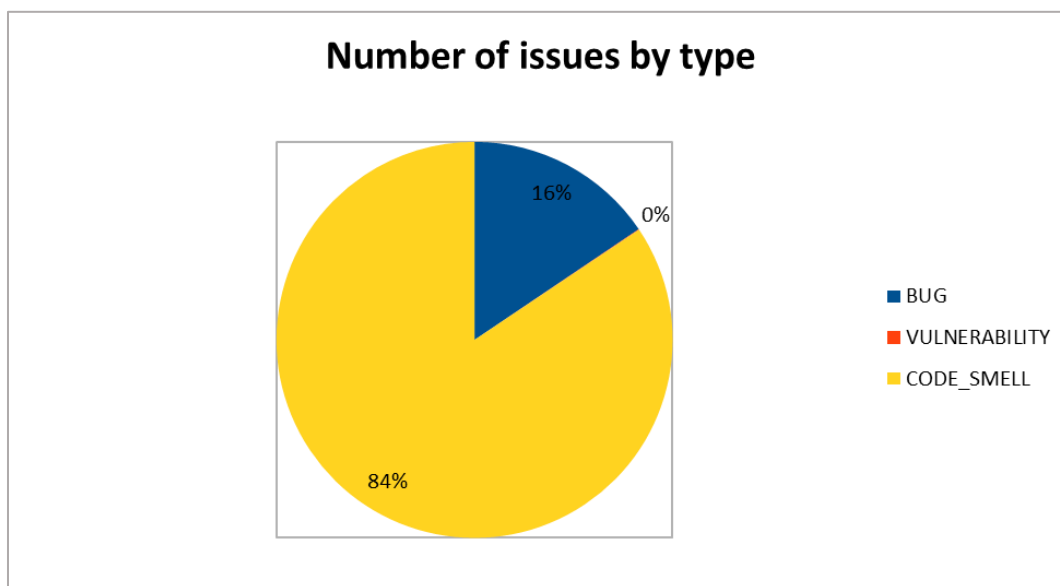


Ilustración 16. Porcentaje de incidencias por categoría.

6.27.1. Análisis de calidad del código

El análisis del código revela un sistema cuya estructura interna presenta serios problemas que afectan directamente su calidad, estabilidad y capacidad de evolucionar de forma segura. Aunque el sistema está actualmente en producción, el estado del código muestra patrones claros de improvisación, falta de refactorización, ausencia de estándares sólidos y una organización deficiente. Esto lo convierte en un producto técnicamente frágil, donde su funcionamiento actual no es consecuencia de una arquitectura sólida, sino más bien de coincidencia operativa: funciona porque nadie ha tocado las partes críticas o porque los errores aún no se han manifestado de manera visible para los usuarios.

Uno de los elementos más preocupantes es la complejidad interna del sistema. Valores como una complejidad lógica que supera los 700 y una complejidad cognitiva que sobrepasa los 500 son indicadores de módulos que han crecido sin control. Este tipo de complejidad no ocurre por accidente: ocurre porque se dejan expandir funciones o componentes sin modularización y sin una revisión seria del diseño. Para un sistema en producción, este nivel de complejidad significa que el código es difícil de entender, extremadamente delicado de modificar y altamente propenso a errores cuando se requiere mantenimiento o nuevas funcionalidades. En la práctica, cualquier intervención en esas zonas del código tiene altas probabilidades de romper algo, y es probable que esos módulos se hayan convertido en "cajas negras" que los desarrolladores evitan tocar.

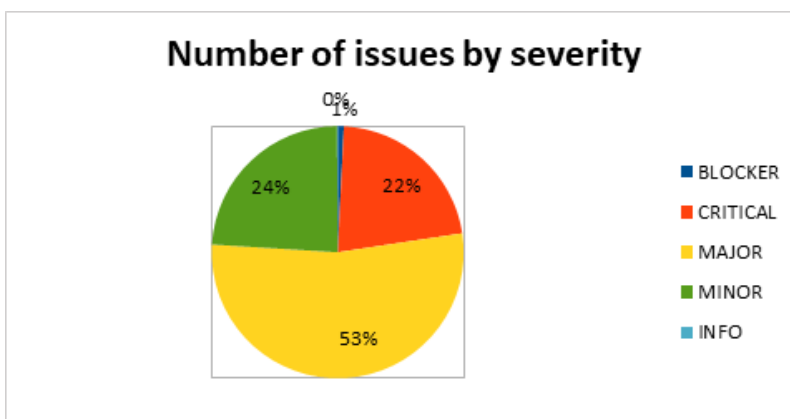


Ilustración 17. Porcentaje de issues por severidad.

La enorme cantidad de code smells encontrados, incluyendo cientos catalogados como major y critical, evidencia un código saturado de malas prácticas. La existencia de variables mal declaradas, uso excesivo de var en JavaScript, duplicación de literales, lógica mal estructurada, falta de llaves en bloques condicionales, y estructuras de control incompletas (como switch sin default) indica un estilo de programación inconsistente, donde

no se aplican estándares de codificación ni revisiones sistemáticas. Este tipo de problemas no es solo una cuestión estética: los code smells son predictores de fallas futuras. Un proyecto en producción con tal volumen de malas prácticas está acumulando deuda técnica que luego se convierte en errores visibles, degradación del rendimiento y comportamientos inesperados.

El uso de archivos con decenas de miles de líneas refuerza la idea de que no existe una arquitectura modular. En un sistema sano, los archivos deberían ser pequeños, enfocados y con responsabilidades delimitadas. Aquí ocurre lo contrario: existen archivos que parecen concentrar gran parte de la lógica, lo que implica que varias capas del sistema están mezcladas en un mismo espacio. Esto dificulta la comprensión del flujo completo y hace que el código sea extremadamente dependiente, lo que aumenta el riesgo ante cualquier cambio. Un archivo tan grande en un sistema en producción es una señal de peligro: probablemente no existe documentación suficiente para entenderlo, y su alto nivel de acoplamiento impide que el equipo pueda intervenirlos sin temor a generar inconsistencias.

Los problemas detectados en HTML, CSS y la capa de presentación complementan este panorama. Las tablas sin estructura semántica, las imágenes sin textos alternativos, los roles ARIA aplicados incorrectamente y la duplicación masiva de CSS revelan que el frontend se ha construido de manera poco disciplinada. Esto se traduce en un código difícil de mantener, difícil de escalar y que además compromete la accesibilidad y la consistencia visual del sistema. Este tipo de deficiencias en un código de producción provoca efectos directos: interfaces inconsistentes, componentes que se comportan de forma desigual en distintos navegadores y dificultades para realizar cambios sin romper la apariencia o el funcionamiento de partes del sitio.

La presencia de bugs reales en el análisis, incluyendo problemas como código inalcanzable, variables no usadas, objetos creados que luego no se utilizan, duplicación de nombres y uso de elementos obsoletos, indica que existen errores que no solo afectan la claridad del código, sino que también pueden provocar fallos en tiempo de ejecución o estados incorrectos de la aplicación. En producción, estos errores pueden generar comportamientos aleatorios, flujos interrumpidos o condiciones que son difíciles de rastrear porque provienen de patrones incorrectos a nivel de código, no de inputs externos.

Finalmente, los hallazgos de seguridad derivados directamente del análisis del código como credenciales incrustadas, uso de protocolos claros, cookies sin atributos de seguridad y manejo inseguro de configuraciones muestran un sistema que expone datos y comportamientos críticos a vulnerabilidades comunes. Estos problemas no requieren un atacante sofisticado para ser explotados, y su origen está precisamente en malas prácticas dentro del código. Cuando un sistema en producción contiene este tipo de fallos, la seguridad no es un concepto abstracto: es una responsabilidad incumplida.

6.27.2. Conteo de incidencias

El análisis de incidencias permite clasificar los problemas detectados en el código según su naturaleza (errores, vulnerabilidades o malas prácticas) y su nivel de severidad. Esta visión facilita comprender no solo la magnitud de los hallazgos, sino también su impacto potencial en la estabilidad, seguridad y mantenibilidad del sistema. La distribución muestra que, la mayor carga proviene de las malas prácticas acumuladas, lo que refleja un código saturado de deuda técnica y con riesgos crecientes a futuro.

| Type / Severity | INFO | MINOR | MAJOR | CRITICAL | BLOCKER |
|-----------------|------|-------|-------|----------|---------|
| BUG | 0 | 165 | 129 | 0 | 0 |
| VULNERABILITY | 0 | 0 | 0 | 0 | 1 |
| CODE SMELL | 2 | 287 | 880 | 412 | 14 |

Tabla 125. Conteo de incidencias por severidad y tipo.

1. Bugs (294 en total)

- Son errores en el código que pueden provocar fallos en ejecución.
- La mayoría son menores (165) y mayores (129), lo que indica problemas que no bloquean el sistema, pero sí afectan su estabilidad y confiabilidad.

2. Vulnerabilidades (1 en total)

- Aunque solo hay una, está catalogada como blocker., esto significa que es un fallo de seguridad crítico que debe corregirse de inmediato porque expone el sistema

3. Code Smells (1,595 en total)

- Son malas prácticas o patrones de programación deficientes.
- La mayoría son mayor (880) y critical (412), lo que refleja un código saturado de problemas de diseño y estilo que dificultan el mantenimiento.
- Además, hay 14 blocker, lo que indica prácticas tan graves que comprometen directamente la calidad y confiabilidad del sistema.

6.27.3. Rango de métricas

El análisis de métricas de rango permite identificar los valores extremos alcanzados por el sistema en aspectos clave. Estos indicadores no muestran promedios, sino los límites mínimos y máximos encontrados durante el análisis, lo que ofrece una visión clara de la disparidad interna del código.

| | Cyclomatic Complexity | Cognitive Complexity | Lines of code per file | Comment density (%) | Duplication (%) |
|------------|------------------------------|-----------------------------|-------------------------------|----------------------------|------------------------|
| Min | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Max | 751.0 | 537.0 | 37480.0 | 100.0 | 100.0 |

Tabla 126. Rango de métricas.

1. Cyclomatic Complexity

- **Mínimo 0.0:** Fragmentos del sistema que son básicos y directos, sin giros ni condiciones.
- **Máximo 751.0:** existe al menos un módulo con una complejidad ciclomática extremadamente alta. Esto significa que dicho módulo tiene varios caminos lógicos posibles, lo que lo hace muy difícil de probar y mantener. En auditoría, un valor >50 ya es preocupante; >700 es crítico.

2. Cognitive Complexity

- **Mínimo 0.0:** Funciones simples.
- **Máximo 537.0:** Funciones con una carga cognitiva enorme. La complejidad cognitiva mide lo difícil que es entender el flujo del código. Un valor >20 ya indica problemas de legibilidad; >500 implica que el código es prácticamente una “caja negra”.

3. Lines of Code per File

- **Mínimo 0:** Archivos vacíos o triviales.
- **Máximo 37,480:** Un archivo concentra casi 40 mil líneas. Esto es una señal clara de falta de modularidad. En buenas prácticas, un archivo debería tener responsabilidades delimitadas y rara vez superar unas pocas centenas de líneas.

4. Comment Density

- **Mínimo 0%:** archivos sin comentarios.
- **Máximo 100%:** archivos que son solo comentarios. La variabilidad extrema refleja inconsistencia en la documentación: algunos módulos están comentados y otros carecen de explicación.

5. Duplication

- **Mínimo 0%:** Código limpio en algunos archivos.
- **Máximo 100%:** archivos completamente duplicados. Esto indica que hay módulos copiados sin refactorización, lo que aumenta la deuda técnica y el riesgo de inconsistencias.

DISEÑO E IMPLEMENTACIÓN DE UN PLAN INTEGRAL DE PRUEBAS MANUALES Y AUTOMATIZADAS PARA ASEGURAR LA CALIDAD DE UN SISTEMA DE COMERCIO ELECTRÓNICO EN PRODUCCIÓN



 Ecommerce

INFORME DE DEFECTOS ENCONTRADOS ECOMMERCE - IMPORTACIONES GUADALAJARA

| Versión | Fecha | Equipo | Descripción |
|---------|------------|---|---|
| V1.0 | 21/11/2025 | Jennifer Tatiana Cruz Nathaly Adriana Rodríguez Mauricio Daniel Hernández | Se consolida los defectos encontrados en el sistema a través de herramientas automatizadas y revisión técnica. Se clasifican los hallazgos por severidad y módulo afectado con el objetivo de facilitar su revisión, corrección y trazabilidad por parte del desarrollador responsable. |

6.28. Informe de defectos/hallazgos encontrados

En el Informe de defectos encontrados se documenta las incidencias identificadas durante la ejecución del ciclo de pruebas, clasificándolas por severidad, módulo afectado y estado. Su propósito es proporcionar una vista consolidada del comportamiento del sistema ante las pruebas funcionales, automatizadas, de seguridad y de rendimiento, y servir como insumo para la toma de decisiones y para el Informe final de cierre.

6.28.1. Resumen

Durante la ejecución de los 77 casos de prueba incluidos en la matriz del proyecto, se identificaron tres tipos principales de resultados:

1. **Casos aprobados:** aquellos donde el resultado obtenido coincide totalmente con el resultado esperado.
2. **Casos fallidos:** aquellos donde se identificó un comportamiento incorrecto que afecta la funcionalidad y genera un defecto.
3. **Casos ejecutados con hallazgos:** aquellos que cumplen la funcionalidad principal, pero presentan observaciones, inconsistencias menores o aspectos susceptibles de mejora.

Estos últimos no se consideran fallas funcionales críticas, pero requieren seguimiento, ya sea en forma de defectos de baja severidad o de mejoras registradas para futuras iteraciones del sistema.

Esta distinción permitió evaluar el comportamiento del sistema desde una perspectiva integral, identificando no solo errores funcionales relevantes, sino también oportunidades de mejora y hallazgos técnicos derivados de pruebas automatizadas, de rendimiento y de seguridad.

6.28.2. Objetivo

El objetivo de este informe es documentar de manera estructurada todos los defectos identificados durante la fase de pruebas, clasificados por severidad, tipo e impacto operativo. Asimismo, busca:

1. Determinar el estado de calidad del sistema.
2. Proveer evidencia sólida para la toma de decisiones.
3. Facilitar la trazabilidad entre casos de prueba y defectos.
4. Apoyar la priorización de correcciones.
5. Servir como insumo para el cierre formal del ciclo de pruebas.

6.28.3. Metodología

La metodología empleada para la identificación y análisis de defectos se estructuró con el fin de asegurar una evaluación completa del sistema, abarcando tanto su funcionamiento interno como su interacción con el usuario final. Para ello, se integraron distintos enfoques de prueba (funcionales, dinámicos, estáticos, de rendimiento y de seguridad), los cuales permitieron detectar incidencias en diferentes niveles del software.

Esta combinación de técnicas facilitó no solo la identificación de fallas visibles durante la ejecución, sino también la detección de vulnerabilidades, inconsistencias de diseño y riesgos operativos que podrían comprometer la estabilidad del sistema. A continuación, se describen los tipos de análisis realizados y los criterios adoptados para la clasificación y priorización de los defectos encontrados.

a) Tipo de análisis realizado

Durante la ejecución del proyecto se aplicaron los siguientes enfoques:

1. **Análisis funcional:** validación de flujos del sistema, reglas de negocio, CRUD, navegación y respuestas esperadas.
2. **Análisis dinámico:** ejecución del sistema en tiempo real mediante pruebas manuales y automatizadas.
3. **Análisis estático:** revisión de código y estándares mediante SonarQube.
4. **Análisis de rendimiento:** utilizando JMeter para medir tiempos de respuesta, throughput y tasa de errores.
5. **Análisis de seguridad:** mediante OWASP ZAP para detectar vulnerabilidades.

b) Criterios de clasificación de defectos

Los defectos se clasificaron según:

1. Severidad

- Alto: Afecta funciones importantes, pero con alternativas temporales.
- Medio: Afecta la experiencia o genera inconsistencias, sin bloquear el flujo.
- Bajo: Errores estéticos, mejoras, textos o componentes visuales.

2. Tipo

- Funcional
- Interfaz / usabilidad
- Automatización
- Seguridad
- Rendimiento

6.28.4. Hallazgos

Se identificaron un total de 20 defectos/hallazgos que requieren acción.

| Estado de la Prueba | Cantidad | Porcentaje del Total |
|--------------------------------|----------|----------------------|
| Fallido | 9 | 11.68% |
| Ejecutado con hallazgos | 11 | 14.28% |
| TOTAL | 20 | 25.96% |

Tabla 127. Defectos clasificados por estado

La distribución de los defectos se concentra principalmente en Seguridad y Login, seguidos por Products y Orders como se muestra en el siguiente gráfico:

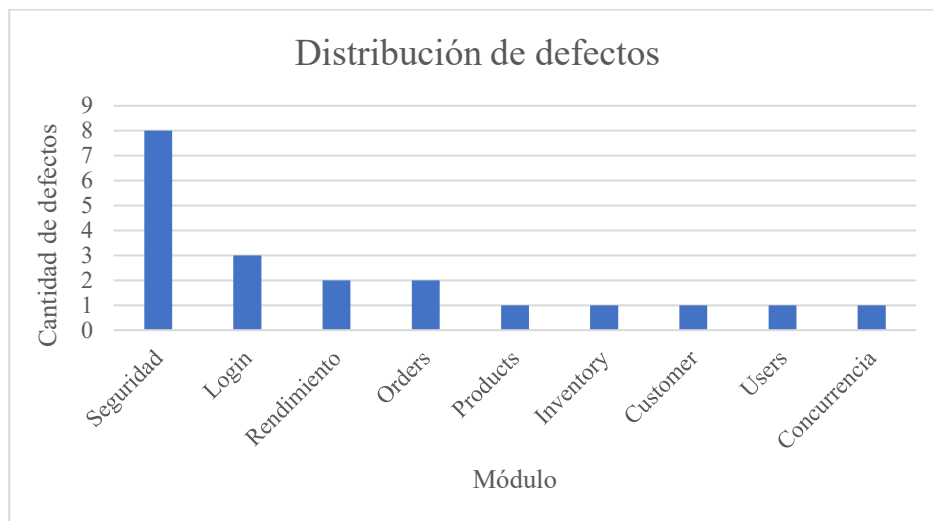


Ilustración 18. Distribución de defectos por módulo

La siguiente tabla desglosa cada uno de los 20 casos de prueba que resultaron en un estado de Fallido o Ejecutado con Hallazgos. El análisis muestra el comportamiento real observado (resultado obtenido), la severidad asignada (basada en el estado de la prueba) y un resumen de las observaciones/recomendaciones clave para su corrección. Los fallos más críticos se concentran en el módulo de Seguridad, comprometiendo la base del sistema.

| ID | Módulo | Título del Caso | Resultado Obtenido | Estado | Criticidad | Observaciones/ Recomendación |
|------|-----------|-----------------------------------|--|---------|------------|---|
| CP56 | Seguridad | Transporte seguro de credenciales | Las credenciales se enviaron en texto plano, lo que representa una vulnerabilidad crítica en el sistema. | Fallido | Crítica | Modificar la forma de envío de las credenciales, ya que si alguien logra interceptarlas quedan grandes vulnerabilidades. |
| CP54 | Seguridad | Revisión de código seguro | El reporte final muestra 412 issues críticos. | Fallido | Crítica | Es urgente la refactorización de código. El análisis del código revela serios problemas que afectan su calidad, estabilidad y capacidad de evolucionar. |
| CP04 | Login | Bloqueo por intentos fallidos | La cuenta no se bloquea por más intentos fallidos de inicio de sesión con un usuario válido. | Fallido | Alta | Crear un espacio donde se registren los intentos fallidos y al pasar de 3 bloquee la cuenta. |

Tabla 128. Detalle de defectos encontrados (Parte 1)

| ID | Módulo | Título del Caso | Resultado Obtenido | Estado | Criticidad | Observaciones/ Recomendación |
|------|--------------|---|--|---------|------------|--|
| CP50 | Concurrencia | Edición del mismo producto | Al guardar las ediciones, el sistema reemplaza los valores y solo conserva la última modificación, lo que provoca que se pierdan los cambios anteriores. | Fallido | Alta | Se deberían actualizar únicamente los campos que sean diferentes al contenido actual en la base de datos. |
| CP16 | Products | Manejo de valores numéricos negativos | El sistema permite guardar el precio negativo sin mostrar ningún mensaje de error y finaliza la operación normalmente. | Fallido | Alta | Se recomienda implementar validación en backend (Request o Model) y opcionalmente validar también en frontend. |
| CP58 | Seguridad | Detección de credenciales predeterminadas/débiles | El sistema permite la creación de usuarios con contraseñas que no cumplen con estándares mínimos de seguridad. | Fallido | Alta | El servidor respondió con redirección (302 Found), lo cual indica que las contraseñas fueron aceptadas sin validación. |
| CP57 | Seguridad | Protección contra fuerza bruta | Bloqueo temporal fijo que no escala progresivamente. No se activa ante ataques automáticos sin estado. No detectó CAPTCHA. | Fallido | Alta | Fortalecer la protección con mecanismos como backoff progresivo, bloqueo por IP/usuario y CAPTCHA o 2FA tras múltiples intentos. |

Tabla 129. Detalle de defectos encontrados (Parte II)

| ID | Módulo | Título del Caso | Resultado Obtenido | Estado | Criticidad | Observaciones/ Recomendación |
|------|-----------|--------------------------------------|--|-------------------------|------------|---|
| CP51 | Seguridad | Gestión de sesión – time-out | Después de 40 min de inactividad, lo cual significa que no se está aplicando el timeout por inactividad. | Fallido | Media | Esto representa un riesgo medio. OWASP lo clasifica como una debilidad en el manejo de sesiones. |
| CP47 | Seguridad | Headers HTTP | Faltan algunos headers de seguridad importantes. El header X-Powered-By está presente (expone tecnología de backend). | Fallido | Media | Agregar headers de seguridad clave (X-Content-Type-Options, X-Frame-Options, etc.) y remover X-Powered-By. |
| CP03 | Login | Campos obligatorios vacíos del login | El formulario se envía y carga nuevamente la vista de login con un mensaje de error de campos requeridos. | Ejecutado con hallazgos | Media | El frontend no valida que los campos se encuentren debidamente llenos antes del envío, por lo que la request de campos vacíos se envía al servidor. |
| CP06 | Login | Redirección por logeo | Usuarios tipo Seller y Customer sí pueden regresar al login sin haber cerrado su sesión. | Ejecutado con hallazgos | Media | Se debería tener un solo login no importando el tipo de usuario que ingrese. |
| CP26 | Inventory | Registro exitoso de bodega | El state_id no corresponde al guardado lo que sugiere una sobrescritura en el controlador (state_id = 1). Paginación limitada. | Ejecutado con hallazgos | Media | Se recomienda agregar navegación visible y revisar filtros para asegurar trazabilidad visual y corregir la sobrescritura. |

Tabla 130. Detalle de defectos encontrados (Parte III)

| ID | Módulo | Título del Caso | Resultado Obtenido | Estado | Criticidad | Observaciones/ Recomendación |
|------|-------------|-------------------------------------|--|-------------------------|------------|--|
| CP37 | Customer | Generación automática de contraseña | El generador actual no garantiza la inclusión de mayúsculas, símbolos o números en la contraseña autogenerada. | Ejecutado con hallazgos | Media | Se recomienda ajustar el script para asegurar que la contraseña autogenerada cumpla con los criterios de seguridad. |
| CP41 | Users | Generación de contraseña de usuario | Los criterios de complejidad de contraseña no siempre son aplicados. | Ejecutado con hallazgos | Media | Se recomienda mejorar la función para asegurar que cumpla con los requisitos de seguridad. |
| CP46 | Seguridad | Acceder a vista Admin | El sistema mostró la pantalla de "Sin permisos", pero no registra ningún log de seguridad. | Ejecutado con hallazgos | Media | Se recomienda implementar un redireccionamiento automático y el registro de logs de seguridad. |
| CP48 | Rendimiento | 100 usuarios concurrentes | Se tuvo un 1.05% de Error en cuanto a caídas de las 100 requests concurrentes. | Ejecutado con hallazgos | Media | Se desvía por un margen menor a 2% del objetivo, se debería ver la manera de ampliar los recursos del servidor si se requiere esta concurrencia. |
| CP49 | Rendimiento | Saturación del sistema | El servidor rechazó las solicitudes cuando se superaron las 100 peticiones provenientes de una misma dirección IP. | Ejecutado con hallazgos | Baja | El servidor bloquea cuando son demasiadas solicitudes de una misma dirección IP, lo cual es bueno para seguridad, pero complica pruebas de saturación. |

Tabla 131. Detalle de defectos encontrados (Parte IV)

| ID | Módulo | Título del Caso | Resultado Obtenido | Estado | Criticidad | Observaciones/ Recomendación |
|------|-----------|--|---|-------------------------|------------|---|
| CP60 | Seguridad | Validación de "Remember Me" | Mantiene la sesión activa durante horas, aunque se cierre el navegador, independiente de si se presiona el botón "Remember me". | Ejecutado con hallazgos | Baja | Independientemente si se presiona o no el botón la sesión queda guardada por un lapso de unas horas. |
| CP76 | Orders | Agregar un producto a la orden | Si la orden se encuentra vacía, el sistema no permite agregar nuevos productos generándose un error no crítico. | Ejecutado con hallazgos | Baja | El sistema debe permitir regenerar dinámicamente order_details al agregar nuevos productos, evitando bloqueos cuando la orden está vacía. |
| CP77 | Orders | Remover producto /productos de una orden | Error SQL al eliminar un producto duplicado. Si la orden queda vacía, el sistema no permite agregar nuevos productos. | Ejecutado con hallazgos | Baja | Revisar la lógica de eliminación para que cada instancia de producto sea tratada de forma independiente y no agrupada. |

Tabla 132. Detalle de defectos encontrados (Parte V)

6.28.5. Defectos más críticos y su impacto

La presente sección resume los defectos que, durante la fase de pruebas, mostraron mayor afectación en la funcionalidad, la estabilidad o la experiencia del usuario dentro del sistema. El análisis permite identificar áreas prioritarias de mejora y orientar los esfuerzos del equipo de desarrollo hacia la mitigación de riesgos que afectan directamente al negocio y a los usuarios finales.

| ID | Módulo /Área | Descripción del defecto | Impacto |
|------|--------------------|--|--|
| CP56 | Seguridad | Envío de credenciales en texto plano | Este hallazgo representa una vulnerabilidad de máxima severidad, ya que permite que cualquier atacante que intercepte el tráfico pueda obtener directamente el usuario y contraseña del sistema. Esto compromete cuentas reales, incluyendo administradores, y habilita accesos no autorizados, robo de información y toma completa del sistema. |
| CP54 | Seguridad / Código | 412 issues críticos en revisión de código | Un volumen tan alto de issues críticos revela un sistema con deficiencias estructurales profundas, donde la calidad del código es tan baja que genera vulnerabilidades recurrentes e inestabilidad. Esto eleva el costo de mantenimiento, dificulta la escalabilidad, aumenta la probabilidad de incidentes de seguridad y ralentiza el desarrollo futuro. |
| CP57 | Seguridad | Protección contra fuerza bruta inefectiva | La protección actual solo funciona parcialmente y puede ser fácilmente evadida por herramientas automatizadas, permitiendo ataques continuos sin ser detectados. Esto abre la puerta a accesos indebidos por predicción de contraseñas, comprometiendo a los usuarios. Un sistema con defensas inconsistentes puede ser vulnerado en rápidamente, afectando continuidad operativa, seguridad y reputación del negocio. |
| CP04 | Login | No se bloquea la cuenta tras múltiples intentos fallidos | La ausencia de bloqueo de cuenta convierte al módulo en un vector de ataque ideal. Un atacante puede intentar miles de combinaciones de contraseñas sin ser detenido, comprometiendo usuarios con credenciales débiles. Esto afecta la seguridad del sistema e incrementa el riesgo de acceso no autorizado. |

Tabla 133. Defectos con mayor criticidad y su impacto (Parte I).

| ID | Módulo /Área | Descripción del defecto | Impacto |
|-------------|--------------|---|---|
| CP58 | Seguridad | Se permiten contraseñas débiles o vacías | Permitir la creación de cuentas con contraseñas débiles los usuarios pueden quedar expuestos a ataques triviales, reduciendo la efectividad de cualquier otro mecanismo de seguridad. Esto compromete la integridad del sistema y facilita accesos |
| CP16 | Products | Manejo de valores numéricos negativos en el precio | El sistema acepta y guarda precios negativos sin mostrar errores, llevando a cálculos de inventario y reportes financieros incorrectos o inconsistentes. |
| CP37 / CP41 | Seguridad | Generación de contraseñas insegura | El sistema genera contraseñas que no cumplen criterios mínimos de robustez. Esto expone cuentas recién creadas, además, crea una falsa sensación de seguridad, ya que la funcionalidad existe, pero no protege realmente. Esto puede provocar accesos no autorizados e incumplimiento de estándares de seguridad de contraseñas. |
| CP50 | Concurrencia | Sobrescritura silenciosa de datos en edición simultánea | Este error provoca pérdida de información sin que el usuario lo note, generando inconsistencias, registros incorrectos y decisiones administrativas basadas en datos erróneos. Este defecto puede escalar a corrupción de datos, pérdida de trazabilidad y problemas operativos graves, afectando inventarios, pedidos, reportes o información crítica de clientes. |

Tabla 134. Defectos con mayor criticidad y su impacto (Parte II).

6.28.6. Patrones de error identificados

Durante el análisis de los casos de prueba ejecutados que resultaron en fallos o hallazgos, se identificaron patrones recurrentes en los defectos reportados. Estos no corresponden a fallos aislados, sino también comportamientos que afectan la consistencia, la usabilidad, la seguridad y la integridad funcional del sistema.

El análisis de los defectos más crítico evidencia que varios de ellos comparten factores relacionados con decisiones de diseño o implementación. Identificar estos patrones es crucial para enfocar la refactorización en las causas raíz que generan la mayoría de las vulnerabilidades y errores de integridad.

- 1. Debilidad en el desarrollo seguro:** Este es el patrón más frecuente y grave. Los defectos demuestran que la seguridad no fue considerada de forma proactiva en el diseño y la implementación. Los fallos no son fallas de lógica de negocio simple, sino vulnerabilidades de infraestructura o código que exponen al usuario y al sistema a ataques de alto impacto.

| Casos de prueba | Consecuencia |
|---------------------------------------|---|
| CP56 (Credenciales en texto plano) | Fallo en la Configuración de Transporte Seguro (Falta de uso o implementación deficiente de HTTPS/TLS). |
| CP54 (412 issues críticos) | Falta de Adopción de Estándares de código seguro, resultando en deficiencias estructurales que generan bugs y vulnerabilidades constantes. |
| CP04, CP57, CP58 (Login) | Fallo en la Implementación de controles de seguridad activos. El sistema carece de defensas efectivas contra ataques automatizados (bloqueo por intentos) y permite la creación de contraseñas débiles. |

Tabla 135. Patrones de error y causas (Parte I).

- 2. Insuficiente en la capa de negocio (Backend):** Este patrón indica que el sistema depende excesivamente del frontend para controlar la calidad de los datos, o que la lógica de negocio esencial no está correctamente implementada en el backend. Al no validar las reglas en el servidor, el sistema es susceptible a la corrupción de datos y a la violación de restricciones de negocio.

| Casos de prueba | Consecuencia |
|--------------------------------------|---|
| CP16 (Precios Negativos) | La ausencia de validación de reglas de negocio, provoca que la restricción de que un precio debe ser positivo solo existe en el frontend o está ausente por completo en el backend (Model/Request). |
| CP26, CP76, CP77 (Varios módulos) | Sobrescritura de id o fallos SQL en eliminación de registros duplicados, indica problemas en la manipulación de claves primarias y foráneas en las consultas. |

Tabla 136. Patrones de error y causas (Parte II).

- 3. Manejo incorrecto de condiciones límite y concurrencia:** Este patrón está asociado con el manejo deficiente de situaciones complejas o simultáneas, provocando la pérdida de datos y la inestabilidad en escenarios de alta demanda.

| Casos de prueba | Consecuencia |
|--|--|
| CP50 (Pérdida de Datos en Concurrencia) | Debido a la ausencia de mecanismos de bloqueo de transacciones, el sistema permite que dos usuarios modifiquen el mismo recurso sin control que genera una condición de carrera y sobrescritura de datos silenciosa. |
| CP48, CP49 (Rendimiento) | Las limitaciones de recursos o configuración ineficiente sugieren que la base de datos no está optimizada o carecen de recursos suficientes para manejar picos de concurrencia. |

Tabla 137. Patrones de error y causas (Parte II).

6.28.7. Análisis de hallazgos de los módulos con bajo desempeño

El análisis de la concentración de fallos identifica al módulo de Seguridad como el módulo con el menor desempeño, convirtiéndose en el área de mayor riesgo y el punto más vulnerable. Este módulo concentra el 60% de los casos con fallos o hallazgos, incluyendo vulnerabilidades críticas como el envío de credenciales en texto plano, código inseguro y defensas inefectivas contra la fuerza bruta.

En segundo lugar, se encuentra el módulo de Login, cuya criticidad radica en su función como punto de acceso principal del sistema, presentando un fallo grave como ser el no bloquear la cuenta tras múltiples intentos fallidos.

Finalmente, el tercer módulo más comprometido es el de Productos/Concurrencia, presenta fallos que comprometen la integridad de los datos del negocio, como la pérdida silenciosa de datos en edición simultáneas y la violación de restricciones de negocio, como permitir precios negativos. Este enfoque confirma que las correcciones deben priorizar la infraestructura de seguridad antes de resolver los problemas de lógica de negocio o usabilidad.

6.28.8. Conclusiones

La ejecución de las pruebas reveló una alta estabilidad funcional en los flujos principales, con un 74% de casos aprobados. Sin embargo, existe una concentración inaceptable de vulnerabilidades con Criticidad Crítica y Alta en el módulo de Seguridad que ponen en riesgo la confidencialidad e integridad del sistema. El problema más urgente es la seguridad en la autenticación, resaltado por el informe de código inseguro (412 issues). La seguridad del sistema no es robusta y requiere una auditoría de código y una refactorización de los componentes de autenticación, autorización y persistencia de datos.

6.28.9. Recomendaciones

Los defectos identificados señalan aspectos del sistema que requieren atención directa. El desarrollador responsable debe priorizar mejoras en los componentes más críticos para fortalecer la seguridad, la integridad de datos y la estabilidad operacional.

Atender estos elementos permitirá corregir comportamientos inconsistentes, reducir vulnerabilidades y asegurar un funcionamiento más confiable y mantenible del sistema en futuras versiones.

1. Seguridad del sistema

- **Transporte seguro de credenciales:** Todo flujo de autenticación debe realizarse bajo HTTPS/TLS, con el objetivo de proteger contra interceptación de credenciales.
- **Refactorización de código inseguro:** Corregir vulnerabilidades críticas, dedicando un sprint específico para abordar fallos de inyección, autenticación y manejo de sesiones, con el fin de reducir la exposición a ataques y fortalece la confiabilidad.
- **Fortalecimiento de defensas:** Implementar mecanismos para evitar ataques de fuerza bruta más robustos (como puede ser la implementación de backoff progresivo, bloqueo por IP/usuario, CAPTCHA/2FA).
- **Gestión de sesiones:** Aplicar timeouts automáticos y corregir el comportamiento inseguro de “Remember Me”, esto también podría evitar el secuestro de sesiones

2. Integridad y consistencia de datos

- **Control de concurrencia:** Garantizar que las ediciones simultáneas no provoquen pérdida silenciosa de información. Lo que ayudaría a proteger la integridad de datos en escenarios multiusuario
- **Validaciones de negocio:** Reforzar reglas en backend para impedir precios negativos y datos inconsistentes. De esta manera se asegura la confiabilidad en operaciones y reportes.
- **Gestión de órdenes y productos:** Ajustar la lógica para permitir operaciones dinámicas sin romper la integridad de la orden, manteniendo la consistencia en inventario y la facturación.

3. Experiencia de usuario y eficiencia operativa

- **Mensajes claros y consistentes:** Mejorar redirecciones y feedback al usuario en login, inventario y órdenes, mejorando así la usabilidad.
- **Navegación y trazabilidad:** Optimizar paginación y filtros para asegurar visibilidad de registros nuevos, agilizando la gestión operativa.

4. Mantenibilidad y escalabilidad

- **Refactorización estructural:** Refactorización estructural centrada en limpiar código duplicado, estandarizar funciones comunes, aumentando la estabilidad y la capacidad de ampliar funcionalidades.
- **Buenas prácticas de desarrollo seguro:** Adoptar estándares o buenas prácticas de desarrollo seguro mediante la aplicación de controles de validación y manejo adecuado de autenticación y sesiones, complementado con principios de Clean Code.

DISEÑO E IMPLEMENTACIÓN DE UN PLAN INTEGRAL DE PRUEBAS MANUALES Y AUTOMATIZADAS PARA ASEGURAR LA CALIDAD DE UN SISTEMA DE COMERCIO ELECTRÓNICO EN PRODUCCIÓN



 Ecommerce

INFORME DE CIERRE DE PRUEBAS ECOMMERCE - IMPORTACIONES GUADALAJARA

| Versión | Fecha | Equipo | Descripción |
|---------|------------|---|---|
| V1.0 | 21/11/2025 | Jennifer Tatiana Cruz Nathaly Adriana Rodríguez Mauricio Daniel Hernández | Se resume el alcance, los resultados, el análisis de defectos y el estado final de la calidad de la versión de software probada. Su objetivo es facilitar la corrección, trazabilidad y cierre técnico por parte del desarrollador responsable. |

6.29. Informe de cierre de pruebas

El presente Informe de cierre de pruebas tiene como finalidad documentar de manera formal las actividades de ejecución realizadas durante los ciclos de pruebas de pruebas del sistema, así como los resultados obtenidos, los defectos identificados y el análisis correspondiente. Este registro consolida la información relevante del proceso de aseguramiento de calidad, proporcionando una visión integral del estado del producto evaluado.

A través de este documento se busca garantizar la transparencia del proceso de pruebas, así como proporcionar una evaluación clara y objetiva del sistema probado, facilitando decisiones relacionadas con su estabilidad, mantenimiento y mejora continua en el ambiente productivo.

6.29.1. Resumen

El presente documento consolida los resultados obtenidos durante la ejecución de pruebas realizadas al sistema E-Commerce – Importaciones Guadalajara. Estas pruebas tuvieron como finalidad validar el cumplimiento de los requisitos funcionales y no funcionales establecidos, asegurar la correcta operación de los módulos críticos y detectar posibles defectos que pudieran comprometer la estabilidad o la experiencia de usuario en el ambiente productivo.

El ciclo de pruebas incluyó la ejecución de casos diseñados para evaluar el comportamiento del sistema, registrando de manera detallada los resultados obtenidos. Cada caso fue documentado con sus observaciones, garantizando la trazabilidad y la reproducibilidad del proceso. Asimismo, se presentan recomendaciones derivadas de los hallazgos, orientadas a la mejora continua y al aseguramiento de la calidad del sistema.

Los hallazgos permiten identificar el nivel de calidad alcanzado, así como el estado de los defectos encontrados y su impacto potencial en la operación del sistema. Este análisis constituye una base objetiva para la toma de decisiones relacionadas, el mantenimiento y la mejora continua del producto, asegurando que se mantenga conforme a los criterios y que se encuentre en condiciones adecuadas para su uso en producción.

En conjunto, este documento consolida los resultados de las pruebas y las observaciones realizadas, ofreciendo una base objetiva para la gestión de calidad proporcionando una visión clara del estado actual del sistema y de las acciones necesarias para su evolución.

6.29.2. Objetivo

El objetivo de este documento es presentar de manera clara y estructurada los resultados obtenidos durante el ciclo de pruebas del sistema, evaluando el nivel de calidad alcanzado, la cobertura de las pruebas ejecutadas y el estado de los defectos identificados. Con ello se busca apoyar la toma de decisiones orientadas al mantenimiento, la mejora continua y el aseguramiento de que el sistema cumple con los criterios establecidos para su operación en producción.

6.29.3. Alcance de las pruebas

Las pruebas ejecutadas se orientaron a verificar el comportamiento de los componentes esenciales del sistema, asegurando que las funciones críticas operaran de manera estable y confiable. Concentrándose en los módulos que sostienen las operaciones principales, documentando los resultados obtenidos, las observaciones derivadas de la ejecución y las recomendaciones para su mantenimiento y mejora continua.

Se aplicaron diferentes enfoques de validación, combinando pruebas funcionales, de integración y de seguridad, junto con evaluaciones de rendimiento y usabilidad. Esto permitió observar el sistema desde distintas perspectivas, garantizando que los resultados obtenidos fueran representativos y útiles para la toma de decisiones técnicas:

- **Validación funcional:** Revisión del cumplimiento de las funciones esperadas en cada módulo y de la correcta ejecución de los procesos clave del sistema.
- **Validación de interacción entre componentes:** Comprobación de la comunicación entre módulos y del intercambio adecuado de información.
- **Evaluaciones de desempeño:** Medición del tiempo de respuesta, uso de recursos y estabilidad del sistema bajo distintos escenarios de carga.
- **Revisión de experiencia del usuario:** Análisis del diseño y navegación, identificando posibles fricciones, inconsistencias o mejoras en la interfaz.
- **Verificaciones de seguridad:** Comprobación de controles de acceso, protección de información sensible y exposición ante vulnerabilidades frecuentes.
- **Pruebas de comportamiento:** Confirmación del funcionamiento adecuado con múltiples usuarios simultáneos y a través de distintos navegadores.

El proceso de aseguramiento de calidad se llevó a cabo sobre los siguientes módulos críticos, cuya correcta operación resulta indispensable para garantizar la estabilidad, seguridad y trazabilidad del sistema en su ambiente productivo:

| Módulo | Función principal |
|------------------|--|
| Login | Autenticación y gestión de sesión de usuarios |
| Products | Administración del catálogo de productos |
| Orders | Registro, edición y seguimiento de órdenes de compra |
| Inventory | Control logístico y gestión de existencias |
| Customers | Gestión de clientes y vinculación con órdenes |
| Users | Administración de usuarios y roles de acceso |
| Category | Organización y clasificación de productos |
| Purchase | Registro de compras a proveedores |

Tabla 138. Módulos evaluados en el ciclo de pruebas.

6.29.4. Estrategia y metodología

La estrategia de pruebas se definió con el propósito de garantizar una evaluación completa del sistema antes de su liberación a producción. Se buscó cubrir las funciones críticas, validar la interacción entre módulos y comprobar la seguridad, el rendimiento y la experiencia de usuario.

La metodología aplicada se estructuró en las siguientes etapas:

- **Planificación:** Se establecieron los objetivos de cada ciclo de pruebas, los criterios de aceptación y los casos de prueba priorizados según el impacto en la operación.
- **Diseño de pruebas:** Se elaboraron escenarios representativos que incluyeron validaciones funcionales, de integración, de seguridad, de rendimiento y de usabilidad.
- **Ejecución:** Las pruebas se realizaron de manera controlada por sprint, registrando resultados, incidencias y observaciones en cada ejecución.
- **Gestión de defectos:** La gestión de defectos se llevó a cabo mediante un registro detallado en Excel, Azure DevOps y documentos en Word, utilizando una plantilla estandarizada para garantizar consistencia en la documentación. Cada defecto fue clasificado por prioridad. Para cada caso se incluyó información clave como identificador, módulo, procedimiento, resultados esperados y resultados obtenidos, lo cual facilitó el seguimiento y la verificación de las correcciones aplicadas.

- **Registro y trazabilidad:** Se documentaron las condiciones iniciales, pasos seguidos y resultados obtenidos, asegurando la reproducibilidad y transparencia del proceso.
- **Evaluación final:** Se consolidaron los resultados para determinar el nivel de calidad alcanzado, la cobertura lograda y el estado de los defectos, apoyando la toma de decisiones sobre la liberación del sistema.

6.29.5. Consolidado de resultados

Se resume a continuación el estado de los casos de prueba ejecutados durante el ciclo de validación. En ella se registran los identificadores, nombres y resultados obtenidos para cada caso, permitiendo visualizar de manera estructurada, lo que permite visualizar de manera estructurada cuáles funcionalidades cumplieron con los criterios establecidos y cuáles presentaron fallos u observaciones menores que deben ser atendidas.

Esta información facilita el análisis del nivel de cobertura alcanzado, la identificación de áreas que requieren corrección y el soporte necesario para la toma de decisiones respecto a la liberación del sistema.

| ID | Módulo | Título del Caso | Estado de la prueba |
|------|----------|--|-------------------------|
| CP01 | Login | Acceso exitoso | Aprobado |
| CP02 | Login | Manejo de credenciales incorrectas | Aprobado |
| CP03 | Login | Campos obligatorios vacíos del login | Ejecutado con hallazgos |
| CP04 | Login | Bloqueo por intentos fallidos | Fallido |
| CP05 | Login | Redirección por rol | Aprobado |
| CP06 | Login | Redirección por logueo | Ejecutado con hallazgos |
| CP07 | Products | Crear producto válido | Aprobado |
| CP08 | Products | Agregar Imagen en formato JPG, PNG o ninguna | Aprobado |
| CP09 | Products | Editar producto existente | Aprobado |
| CP10 | Products | Búsqueda de productos | Aprobado |
| CP11 | Products | Envío de producto a estado “Trashed” | Aprobado |
| CP12 | Products | Eliminación en cascada | Aprobado |
| CP13 | Products | Restauración de producto individual | Aprobado |
| CP14 | Products | Restauración masiva de productos | Aprobado |
| CP15 | Products | Guardar con campos obligatorios vacíos | Aprobado |
| CP16 | Products | Manejo de valores numéricos negativos | Fallido |
| CP17 | Orders | Filtrar órdenes por rango de fechas | Aprobado |

Tabla 139. Consolidado de resultados de casos de prueba (Parte I).

| ID | Módulo | Título del Caso | Estado de la prueba |
|------|--------------|--|-------------------------|
| CP18 | Orders | Visualización de detalle de orden | Aprobado |
| CP19 | Orders | Actualización de envío y número de seguimiento | Aprobado |
| CP20 | Orders | Modificación de detalles de orden | Aprobado |
| CP21 | Orders | Actualización de estado de pago | Aprobado |
| CP22 | Orders | Exportación de órdenes a Excel | Aprobado |
| CP23 | Orders | Impresión de detalle de orden | Aprobado |
| CP24 | Orders | Generación de reporte de ventas por producto | Aprobado |
| CP25 | Orders | Eliminación de orden | Aprobado |
| CP26 | Inventory | Registro exitoso de bodega | Ejecutado con hallazgos |
| CP27 | Inventory | Filtro por bodega en inventario | Aprobado |
| CP28 | Inventory | Exportación de inventario a Excel | Aprobado |
| CP29 | Inventory | Consulta de historial Kardex mensual | Aprobado |
| CP30 | Inventory | Verificación del saldo por movimiento | Aprobado |
| CP31 | Inventory | Carga masiva de stock desde Excel | Aprobado |
| CP32 | Customer | Registrar nuevo cliente válido | Aprobado |
| CP33 | Customer | Editar cliente existente | Aprobado |
| CP34 | Customer | Desactivar (eliminar) cliente | Aprobado |
| CP35 | Customer | Búsqueda de cliente por nombre | Aprobado |
| CP36 | Customer | Envío de credenciales por email | Aprobado |
| CP37 | Customer | Generación automática de contraseña | Ejecutado con hallazgos |
| CP38 | Customer | Búsqueda de cliente según Estado y combinada | Aprobado |
| CP39 | Users | Creación de usuario Administrador o Seller | Aprobado |
| CP40 | Users | Validación de email único | Aprobado |
| CP41 | Users | Generación de contraseña de usuario | Ejecutado con hallazgos |
| CP42 | Users | Edición de rol de usuario | Aprobado |
| CP43 | Users | Validación de opciones del campo Rol | Aprobado |
| CP44 | Users | Validación de Campos Vacíos | Aprobado |
| CP45 | Seguridad | Inyección código SQL | Aprobado |
| CP46 | Seguridad | Acceder a vista Admin | Ejecutado con hallazgos |
| CP47 | Seguridad | Headers HTTP | Fallido |
| CP48 | Rendimiento | 100 usuarios concurrentes | Ejecutado con hallazgos |
| CP49 | Rendimiento | Saturación del sistema | Ejecutado con hallazgos |
| CP50 | Concurrencia | Edición del mismo producto | Fallido |

Tabla 140. Consolidado de resultados de casos de prueba (Parte II).

| ID | Módulo | Título del Caso | Estado de la prueba |
|------|-----------|---|-------------------------|
| CP51 | Seguridad | Gestión de sesión – time-out | Fallido |
| CP52 | Seguridad | Restricción de acceso por rol | Aprobado |
| CP53 | UI/UX | Compatibilidad con navegadores | Aprobado |
| CP54 | Seguridad | Revisión de código seguro | Fallido |
| CP55 | Seguridad | Validación de certificado SSL/TLS | Aprobado |
| CP56 | Seguridad | Transporte seguro de credenciales | Fallido |
| CP57 | Seguridad | Protección contra fuerza bruta | Fallido |
| CP58 | Seguridad | Detección de credenciales predeterminadas/débiles | Fallido |
| CP59 | Seguridad | Enumeración de cuentas | Aprobado |
| CP60 | Seguridad | Validación de “Remember Me” | Ejecutado con hallazgos |
| CP61 | Seguridad | Gestión segura de sesiones | Aprobado |
| CP62 | Seguridad | Bypass de autenticación y MFA | Aprobado |
| CP63 | Seguridad | Respuestas ante errores y mensajes | Aprobado |
| CP64 | UI/UX | Pruebas de UI/UX y resiliencia | Aprobado |
| CP65 | Purchase | Vista inicial de categorías | Aprobado |
| CP66 | Purchase | Visualización de productos por categoría | Aprobado |
| CP67 | Purchase | Gestión de productos en carrito | Aprobado |
| CP68 | Purchase | Cálculo de totales | Aprobado |
| CP69 | Purchase | Actualización de estado de pago (Admin) | Aprobado |
| CP70 | Category | Validación de campos obligatorios | Aprobado |
| CP71 | Category | Validación de carga de imagen | Aprobado |
| CP72 | Category | Registro exitoso de categoría | Aprobado |
| CP73 | Category | Retorno a lista desde Create Category | Aprobado |
| CP74 | Category | Eliminar categoría | Aprobado |
| CP75 | Category | Editar una categoría | Aprobado |
| CP76 | Orders | Agregar un producto a la orden | Ejecutado con hallazgos |
| CP77 | Orders | Remover producto /productos de una orden | Ejecutado con hallazgos |

Tabla 141. Consolidado de resultados de casos de prueba (Parte III).

Durante la etapa de ejecución, cada caso de prueba se validó en tres ocasiones considerando los navegadores definidos en el plan de pruebas. El total de ejecuciones fue de 77 casos, de los cuales:

- 57 - Fueron aprobados, cumpliendo con los criterios establecidos.
- 9 - Resultaron fallidos, evidenciando desviaciones respecto al comportamiento esperado.
- 11 - Quedaron ejecutados con hallazgos, presentando observaciones menores que requieren atención.

6.29.6. Resultados de las pruebas por Sprint

Para la ejecución de los casos de pruebas se realizó en tres Sprints, cada uno orientado a validar funcionalidades clave del sistema Ecommerce. En cada iteración se ejecutaron todos los casos de prueba planificados, clasificando los resultados en Aprobado, Fallido y Ejecutados con hallazgos.

○ Sprint 1

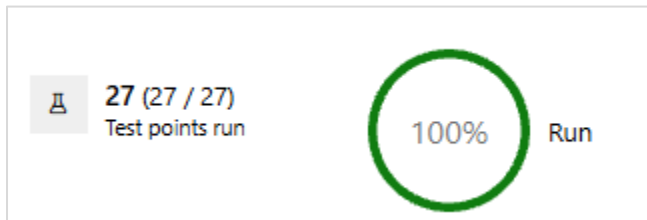


Ilustración 19. Total de casos de pruebas ejecutados – Sprint 1.

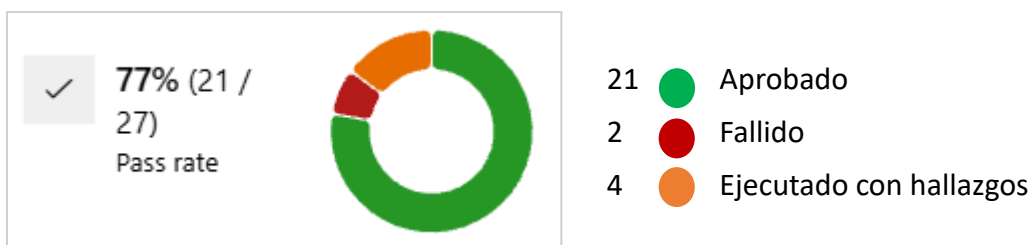


Ilustración 20. Total de casos por estado – Sprint 1

| Métrica | Valor | Interpretación |
|--------------------------------|----------------|---|
| Test Points totales | 27 | Se definieron 27 puntos de prueba. |
| Test Points ejecutados | 27 / 27 (100%) | Todos los puntos fueron ejecutados. No hay pendientes. |
| Casos Passed | 21 | 77% de los casos pasaron exitosamente. Buen indicador de estabilidad. |
| Casos Failed | 2 | 7.4% fallaron. Requieren análisis de causa raíz y posible corrección. |
| Casos Ejecutados con hallazgos | 4 | 14.8% presentaron observaciones que no afecta al funcionamiento del sistema pero que requieren documentación y seguimiento. |

Tabla 142. Resumen de ejecución de casos de pruebas – Sprint 1.

○ **Sprint 2**



Ilustración 21. Total de casos de pruebas ejecutados – Sprint 2.

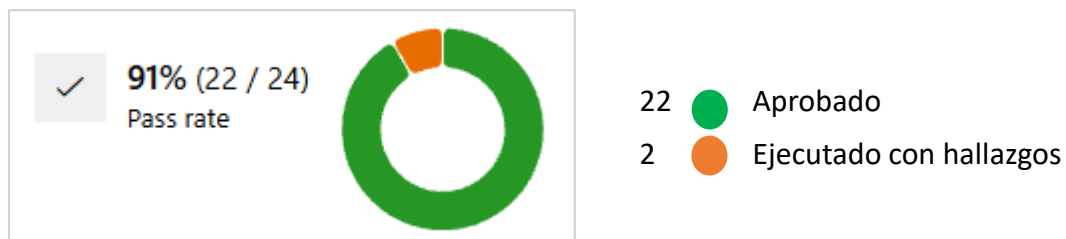


Ilustración 22. Total de casos por estado – Sprint 1

| Métrica | Valor | Interpretación |
|-------------------------------|----------------|--|
| Test Points totales | 24 | Se definieron 24 puntos de prueba para esta iteración. |
| Test Points ejecutados | 24 / 24 (100%) | Todos los puntos fueron ejecutados. No hay pendientes. |
| Casos Passed | 22 | 91.6% pasaron exitosamente. Refleja una mejora significativa respecto al Sprint 1. |
| Casos Failed | 0 | No se registraron fallos. Indicador de estabilidad funcional en esta fase. |
| Casos Ejecutado con hallazgos | 2 | 8.3% presentaron observaciones que no afecta al funcionamiento del sistema pero que requieren documentación y seguimiento. |

Tabla 143. Resumen de ejecución de casos de pruebas – Sprint 2.

○ **Sprint 3**

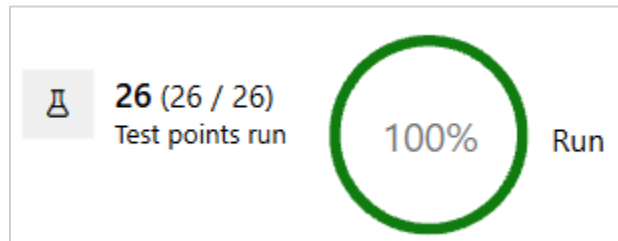


Ilustración 23. Total de casos de pruebas ejecutados – Sprint 3.

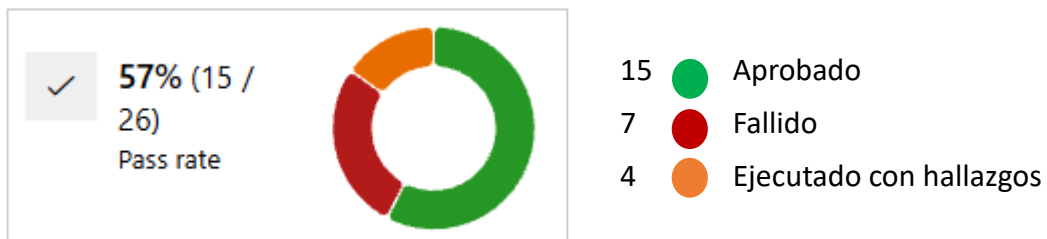


Ilustración 24. Total de casos por estado – Sprint 3.

| Métrica | Valor | Interpretación |
|--------------------------------------|-------------------|---|
| Test Points totales | 26 | Se definieron 26 puntos de prueba para esta iteración. |
| Test Points ejecutados | 26 / 26 (100%) | Todos los puntos fueron ejecutados. No hay pendientes. |
| Casos Passed | 15 | 57.7% pasaron exitosamente. Disminución respecto a sprints anteriores. |
| Casos Failed | 7 | 26.9% fallaron. Requieren análisis técnico y corrección prioritaria. |
| Casos Ejecutado con hallazgos | 4 | 15.4% presentaron observaciones que no afecta al funcionamiento del sistema pero que requieren documentación y seguimiento. |

Tabla 144. Resumen de ejecución de casos de pruebas – Sprint 3

6.29.7. Resultados por módulos

La siguiente sección presenta un resumen técnico de los resultados obtenidos durante la ejecución de pruebas en los distintos módulos del sistema. Cada tabla incluye métricas clave, observaciones funcionales y un desglose por prioridad, facilitando la trazabilidad de hallazgos y la identificación de áreas críticas para revisión.

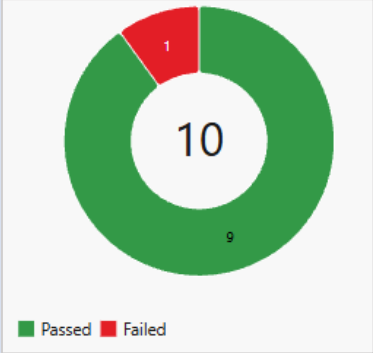
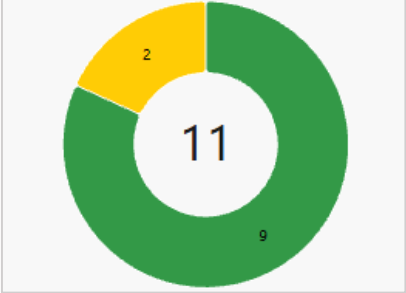
| Módulo | Valor | Ilustración |
|-------------------------|---|---|
| Módulo | Products |  |
| Casos ejecutados | 10/10 | |
| Aprobado | 9 | |
| Fallido | 1 | |
| Con hallazgos | 0 | |
| Interpretación | <p>El módulo presenta alta estabilidad funcional, con solo un caso fallido que requiere revisión puntual.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 3 pasados, 1 fallido • Prioridad 3: 4 pasados, 0 fallidos • Prioridad 2: 2 pasados, 0 fallidos (ejecución correcta en casos de menor) | |
| Módulo | Orders |  |
| Casos ejecutados | 11/11 | |
| Aprobado | 9 | |
| Fallido | 0 | |
| Con hallazgos | 2 | |
| Interpretación | <p>El módulo muestra comportamiento funcional estable, sin fallos registrados. Dos casos fueron ejecutados con hallazgos técnicos que deben documentarse para seguimiento.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 5 pasados, 1 con hallazgos (requiere revisión técnica por tratarse de alta criticidad). • Prioridad 3: 4 pasados, 1 con hallazgos (observación menor, pero debe quedar registrada para trazabilidad). | |

Tabla 145. Resumen de ejecución de casos de pruebas por módulo (Parte I).

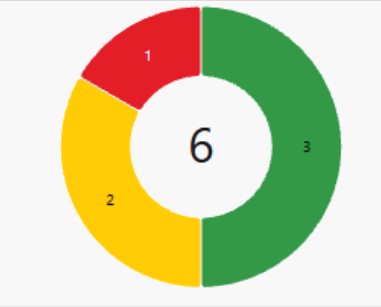
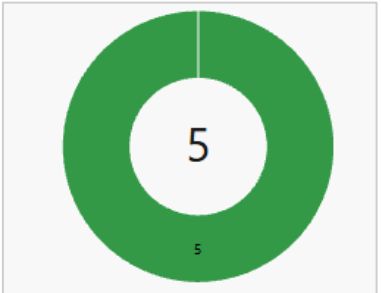
| Módulo | Valor | Ilustración |
|-------------------------|--|---|
| Módulo | Login |  |
| Casos ejecutados | 6/6 | |
| Aprobado | 3 | |
| Fallido | 1 | |
| Con hallazgos | 2 | |
| Interpretación | <p>El módulo Login presenta una cobertura completa con resultados mixtos. Un caso fallido requiere corrección funcional, mientras que dos casos ejecutados con hallazgos deben documentarse para seguimiento técnico.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 2 pasados, 1 fallido, 1 con hallazgos (fallo crítico y observación relevante en casos de alta prioridad). • Prioridad 3: 1 pasado, 1 con hallazgos (comportamiento funcional con observación menor que debe registrarse). | |
| Módulo | Purchase |  |
| Casos ejecutados | 5/5 | |
| Aprobado | 5 | |
| Fallido | 0 | |
| Con hallazgos | 0 | |
| Interpretación | <p>El módulo Purchase muestra comportamiento completamente estable. Todos los casos fueron ejecutados exitosamente sin fallos ni hallazgos técnicos.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: Todos los casos fueron clasificados como pasados. No se registraron observaciones ni fallos en ninguna categoría de prioridad. | |

Tabla 146. Resumen de ejecución de casos de pruebas por módulo (Parte II).

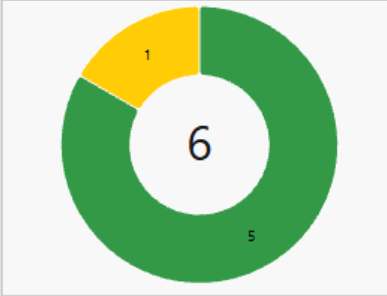
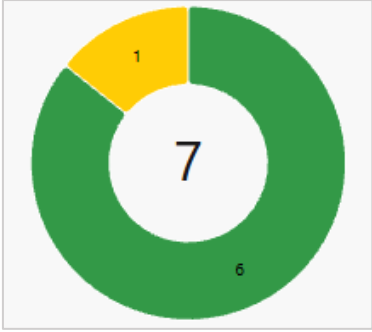
| Módulo | Valor | Ilustración |
|-------------------------|---|---|
| Módulo | Inventory |  |
| Casos ejecutados | 6/6 | |
| Aprobado | 5 | |
| Fallido | 0 | |
| Con hallazgos | 1 | |
| Interpretación | <p>El módulo Inventory muestra comportamiento funcional estable, sin fallos registrados. Un caso fue ejecutado con hallazgos técnicos que deben documentarse para seguimiento.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 2 pasados, 1 con hallazgos (requiere revisión por tratarse de alta criticidad). • Prioridad 3: 2 pasados, 0 con hallazgos • Prioridad 2: 1 pasado, 0 con hallazgos (sin observaciones en casos de baja prioridad). | |
| Módulo | Customer |  |
| Casos ejecutados | 7/7 | |
| Aprobado | 6 | |
| Fallido | 0 | |
| Con hallazgos | 1 | |
| Interpretación | <p>El módulo Customer presenta una ejecución estable, sin fallos registrados. Un caso fue ejecutado con hallazgos técnicos que deben documentarse para trazabilidad.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 3 pasados, 1 con hallazgos (observación relevante en caso de alta criticidad). • Prioridad 3: 2 pasados, 0 con hallazgos • Prioridad 2: 1 pasado, 0 con hallazgos (sin hallazgos en casos de baja prioridad). | |

Tabla 147. Resumen de ejecución de casos de pruebas por módulo (Parte III).

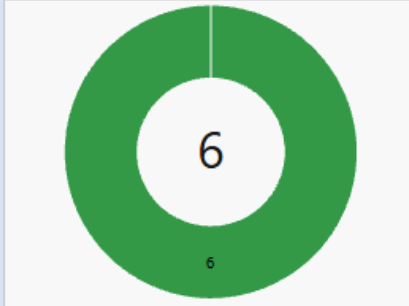
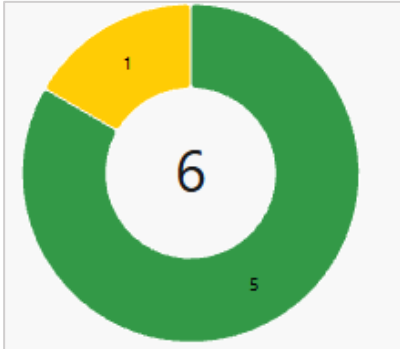
| Módulo | Valor | Ilustración |
|-------------------------|---|---|
| Módulo | Category |  |
| Casos ejecutados | 6/6 | |
| Aprobado | 6 | |
| Fallido | 0 | |
| Con hallazgos | 0 | |
| Interpretación | <p>El módulo Category presenta ejecución completamente estable, sin fallos ni hallazgos técnicos. Todos los casos fueron exitosos, lo que refleja una implementación sólida.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 3 pasados • Prioridad 3: 2 pasados (comportamiento funcional correcto en prioridad media). • Prioridad 2: 1 pasado (sin hallazgos en casos de baja prioridad). | |
| Módulo | Users |  |
| Casos ejecutados | 6/6 | |
| Aprobado | 5 | |
| Fallido | 0 | |
| Con hallazgos | 1 | |
| Interpretación | <p>El módulo Users presenta una ejecución funcional aceptable, sin fallos registrados. Un caso fue ejecutado con hallazgos técnicos que deben documentarse para trazabilidad.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: Todos los casos fueron clasificados como pasados o con hallazgos. No se registraron fallos, pero se requiere seguimiento técnico sobre el caso observado. | |

Tabla 148. Resumen de ejecución de casos de pruebas por módulo (Parte IV).

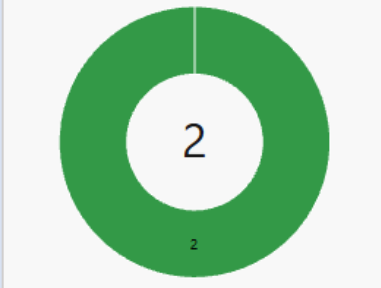
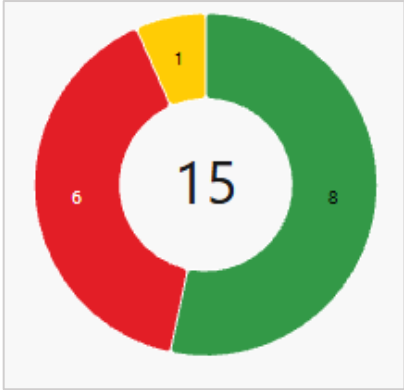
| Módulo | Valor | Ilustración |
|-------------------------|--|--|
| Módulo | UI/UX |  |
| Casos ejecutados | 2/2 | |
| Aprobado | 2 | |
| Fallido | 0 | |
| Con hallazgos | 0 | |
| Interpretación | <p>El módulo UI/UX presenta ejecución completamente estable, sin fallos ni hallazgos técnicos. Ambos casos fueron exitosos, lo que refleja una correcta implementación visual y de interacción.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 3: Todos los casos fueron clasificados como pasados. No se registraron observaciones ni fallos en ninguna categoría de prioridad. | |
| Módulo | Seguridad |  |
| Casos ejecutados | 15/15 | |
| Aprobado | 8 | |
| Fallido | 6 | |
| Con hallazgos | 1 | |
| Interpretación | <p>El módulo Seguridad presenta una ejecución crítica con 6 fallos funcionales que deben ser atendidos con prioridad. Aunque más de la mitad de los casos fueron exitosos, se identificó un caso con hallazgos técnicos que también requiere seguimiento.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: Los fallos se concentran en casos de alta criticidad, lo que sugiere vulnerabilidades funcionales que deben ser corregidas antes de avanzar. La documentación técnica debe incluir trazabilidad de cada hallazgo y fallo para facilitar el análisis posterior. | |

Tabla 149. Resumen de ejecución de casos de pruebas por módulo (Parte V).

| Módulo | Valor | Ilustración |
|-------------------------|---|---|
| Módulo | Rendimiento |  |
| Casos ejecutados | 2/2 | |
| Aprobado | 0 | |
| Fallido | 0 | |
| Con hallazgos | 2 | |
| Interpretación | <p>El módulo Rendimiento fue ejecutado completamente, sin casos exitosos ni fallidos. Ambos casos fueron etiquetados como ejecutados con hallazgos, lo que indica que se identificaron observaciones técnicas relevantes que deben documentarse y analizarse antes de avanzar.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: Ambos casos fueron clasificados como ejecutados con hallazgos. Se requiere trazabilidad técnica para justificar exclusiones y validar criterios de evaluación. | |
| Módulo | Concurrencia |  |
| Casos ejecutados | 1/1 | |
| Aprobado | 0 | |
| Fallido | 1 | |
| Con hallazgos | 0 | |
| Interpretación | <p>El módulo Concurrencia fue ejecutado completamente, pero presentó un fallo funcional que debe ser atendido con prioridad. Al tratarse de un único caso, se recomienda documentar el entorno de ejecución y validar los criterios de evaluación para asegurar reproducibilidad.</p> <p>Detalle por prioridad:</p> <ul style="list-style-type: none"> • Prioridad 4: 1 fallido (fallo crítico que requiere análisis técnico detallado). | |

Tabla 150. Resumen de ejecución de casos de pruebas por módulo (Parte VI).

6.29.9. Resultados globales

Esta sección presenta una visión consolidada del desempeño funcional del sistema Ecommerce a lo largo de sus tres iteraciones. Se incluyen métricas globales de ejecución, tasas de éxito y hallazgos técnicos, permitiendo evaluar la estabilidad del sistema y priorizar acciones correctivas en los módulos con mayor criticidad.

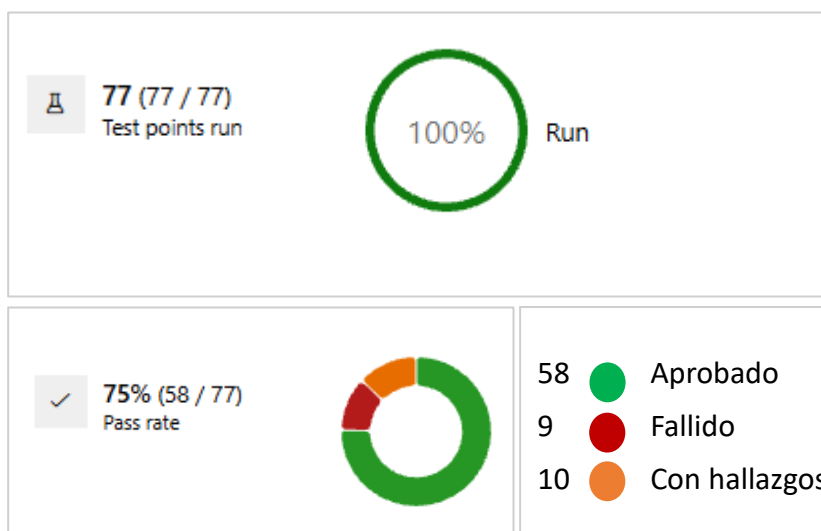


Ilustración 25. Resumen general de ejecución de casos.

| Métrica | Valor |
|--------------------------------------|---|
| Total de planes de prueba | 3 |
| Total de Test ejecutados | 77 / 77 |
| Casos Aprobados | 58 |
| Casos Fallido | 9 |
| Casos Ejecutado con hallazgos | 10 |
| Interpretación | El sistema fue evaluado en tres iteraciones con cobertura total. El 75 % de los casos fueron exitosos, mientras que el 25 % restante incluye fallos funcionales y hallazgos técnicos que deben ser documentados y atendidos. Esta visión consolidada permite identificar módulos críticos y priorizar acciones correctivas. |

Tabla 151. Resumen general de ejecución de casos.

6.29.10. Análisis de métricas

Las métricas de cobertura y desempeño de pruebas permiten evaluar de forma cuantitativa la calidad del sistema y la efectividad del proceso de validación. A través de ellas se mide no solo el cumplimiento de la planificación, sino también la proporción de casos exitosos, fallidos y con hallazgos, además de la distribución de defectos por módulo. Este análisis facilita la trazabilidad y priorización de acciones correctivas en áreas críticas.

- a) **Cobertura de pruebas:** La cobertura total indica que todos los casos planificados fueron ejecutados, lo que asegura que no existen brechas en la validación y que el sistema fue evaluado en su totalidad.

$$\text{Cobertura de pruebas} = \frac{77}{77} \times 100 = 100\%$$

Ecuación 7. Resultados de cobertura de pruebas ejecutadas.

- b) **Porcentaje de casos de prueba aprobados:** Tres de cada cuatro casos fueron exitosos, reflejando un nivel aceptable de estabilidad funcional. Sin embargo, el 26 % restante evidencia áreas que requieren atención para alcanzar estándares óptimos.

$$\text{Casos de casos de prueba aprobados} = \frac{57}{77} \times 100 = 74.03\%$$

Ecuación 8. Resultados de las pruebas aprobadas.

- c) **Porcentaje de casos de prueba fallidos:** Aunque el porcentaje de fallos es relativamente bajo, su concentración en módulos críticos como Seguridad y Concurrencia incrementa el riesgo operativo. Estos defectos deben ser priorizados

$$\text{Casos de casos de prueba fallidos} = \frac{9}{77} \times 100 = 11.68\%$$

Ecuación 9. Resultados de las pruebas fallidas.

- d) **Porcentaje de casos de prueba con hallazgos:** Los hallazgos representan observaciones técnicas que, aunque no son fallos, requieren documentación y seguimiento. Su proporción es significativa y aporta información valiosa para mejorar procesos y configuraciones.

$$\text{Casos de casos de prueba con hallazgos} = \frac{11}{77} \times 100 = 14.28\%$$

Ecuación 10. Resultados de las pruebas ejecutadas con hallazgos.

e) Tasa de defectos por módulo

- El módulo Seguridad concentra el 67 % de los defectos, siendo el área más crítica.
- Productos, Login y Concurrencia aportan defectos aislados (11 % cada uno).
- Los demás módulos no registraron defectos, lo que refleja estabilidad funcional en esas áreas.

| Módulo | Defectos encontrados | Tasa sobre el total |
|--------------|--------------------------|---------------------|
| Productos | $\frac{1}{9} \times 100$ | 11 % |
| Orders | $\frac{0}{9} \times 100$ | 0 % |
| Login | $\frac{1}{9} \times 100$ | 11 % |
| Purchase | $\frac{0}{9} \times 100$ | 0 % |
| Inventory | $\frac{0}{9} \times 100$ | 0 % |
| Customer | $\frac{0}{9} \times 100$ | 0 % |
| Category | $\frac{0}{9} \times 100$ | 0 % |
| Users | $\frac{0}{9} \times 100$ | 0 % |
| UI/UX | $\frac{0}{9} \times 100$ | 0 % |
| Seguridad | $\frac{6}{9} \times 100$ | 67 % |
| Rendimiento | $\frac{0}{9} \times 100$ | 0 % |
| Concurrencia | $\frac{1}{9} \times 100$ | 11 % |

Tabla 152. Porcentaje de defectos encontrados por módulos.

6.29.11. Conclusiones

- Cobertura total de pruebas Se alcanzó una cobertura del 100 %, lo que garantiza que todos los casos planificados fueron ejecutados y que no existen brechas en la validación del sistema.
- Nivel de éxito funcional El 74 % de los casos fueron aprobados, reflejando un nivel aceptable de estabilidad. Sin embargo, el 26 % restante incluye fallos y hallazgos que evidencian áreas críticas que deben ser atendidas.

- **Distribución de fallos** El 11 % de los casos fallidos se concentran principalmente en los módulos Seguridad y Concurrencia, lo que incrementa el riesgo operativo y afecta directamente la confiabilidad del sistema.
- **Hallazgos técnicos** El 14 % de los casos ejecutados con hallazgos aportan observaciones relevantes que, aunque no representan fallos, requieren documentación y seguimiento para garantizar trazabilidad y mejora continua.
- **Tasa de defectos por módulo** El módulo Seguridad concentra el 67 % de los defectos, convirtiéndose en el área más crítica. Los módulos Login, Concurrencia y Productos aportan defectos aislados (11 % cada uno), mientras que el resto de módulos se ejecutaron sin defectos, reflejando estabilidad funcional.

6.29.12. Recomendaciones sobre los casos ejecutados

- **Priorización de correcciones:** Atender de forma inmediata los defectos en Seguridad, dado que representan más de dos tercios del total y afectan directamente la confiabilidad del sistema. Los defectos en Concurrencia, Login y Productos deben ser corregidos en paralelo para evitar riesgos acumulados.
- **Gestión de hallazgos técnicos:** Se debe de registrar y analizar los casos con hallazgos, documentando entorno de ejecución, criterios aplicados. Esto no solo asegura trazabilidad, sino que también permite transformar observaciones en mejoras concretas del código y la arquitectura.
- **Fortalecer la seguridad:** Se debe ampliar la cobertura de validación en los módulos Seguridad y Concurrencia, aplicando buenas prácticas como cifrado actualizado, validación de entradas y pruebas de penetración internas. Esto reducirá la alta concentración de defectos en este módulo.
- **Optimizar concurrencia y rendimiento:** Analizar el manejo de bloqueos y transacciones en el módulo de Concurrencia. Implementar patrones de diseño que eviten condiciones de carrera y optimizar consultas SQL para mejorar la respuesta bajo carga.
- **Auto-revisión disciplinada del código:** El desarrollador puede establecer un checklist personal antes de cerrar cada tarea: seguridad, rendimiento, manejo de errores y trazabilidad. Esto sustituye la revisión externa y asegura consistencia.

7. Recomendaciones

Identificar puntos críticos de la lógica del negocio. Aunque la intención pueda ser probar todo, es crucial priorizar los aspectos que previenen el menor impacto en la aplicación de la empresa. Esto incluye validar las funcionalidades esenciales que sostienen la operación y la experiencia del usuario antes de abordar áreas menos críticas.

Estandarizar y documentar. La documentación clara y consistente es indispensable, incluso al trabajar en solitario. Una buena práctica es escribir notas que no solo sean comprensibles para otros, sino también para uno mismo en el futuro. Esto garantiza continuidad, facilita la colaboración y evita pérdida de tiempo al recordar decisiones tomadas.

Usar métricas cuantitativas. Evita términos vagos como "funciona bien" o "a veces". Mide el desempeño del sistema mediante datos concretos, como tiempos de respuesta, tasa de errores y porcentajes de cumplimiento de casos de prueba. Esto no solo permite evaluar el estado actual, sino también establecer objetivos claros para el futuro.

Automatización estratégica de pruebas. Las pruebas automáticas son esenciales en entornos dinámicos. Si bien las manuales pueden ser más rápidas de implementar inicialmente, la automatización permite obtener datos estadísticos y reducir la carga operativa a largo plazo. Herramientas como Selenium, JMeter o OWA ZAP pueden integrarse en los flujos de trabajo.

Existen varias herramientas que permiten la integración con gestores ágiles, que permiten una mayor funcionalidad y visibilidad al equipo de lo que está sucediendo alrededor de las pruebas, aunque no hemos podido usarlas en el sistema por licencias o pruebas gratuitas limitadas, el uso de ella en un entorno profesional se vuelve un factor importante para facilitar los procesos.

Pruebas en múltiples entornos. Simula distintos entornos para replicar condiciones reales de uso, como diferentes navegadores, sistemas operativos o dispositivos. Esto mejora la confiabilidad de las pruebas y reduce el riesgo de errores en producción.

8. Lecciones aprendidas

- A lo largo del proyecto, el equipo tomó decisiones estratégicas que marcaron la dirección del plan de pruebas. Una de las más importantes fue priorizar únicamente los módulos críticos del sistema como productos, órdenes, inventario, clientes, usuarios y login dejando fuera funciones secundarias que no afectaban directamente los flujos de negocio. Esta decisión permitió concentrar esfuerzos en áreas de alto impacto, optimizando el tiempo y los recursos disponibles. Asimismo, se optó por combinar pruebas manuales con pruebas automatizadas, entendiendo que cada enfoque aporta beneficios distintos y complementarios: las primeras facilitan la evaluación visual y de experiencia de usuario, mientras que las segundas permiten asegurar regresión y estabilidad en flujos repetitivos. También se decidió adoptar herramientas de código abierto compatibles con Laravel, lo cual resultó crucial para garantizar integración técnica, reducir costos y agilizar la curva de aprendizaje del equipo.
- Durante el proceso, el proyecto incorporó varias innovaciones que fortalecieron la metodología de trabajo. La automatización no se aplicó de manera masiva, sino en función del riesgo y del valor que cada flujo aportaba al negocio. Esta priorización inteligente aseguró que los scripts automatizados resolvieran problemas reales, especialmente en login, creación de productos, gestión de órdenes y análisis de seguridad. Además, se desarrolló una matriz integral de trazabilidad que conectaba requerimientos, casos de prueba, resultados y hallazgos, permitiendo visualizar la cobertura total del proceso. Otra innovación significativa fue integrar pruebas de seguridad y rendimiento desde etapas tempranas, en lugar de tratarlas como actividades aisladas al final del ciclo, lo que permitió identificar vulnerabilidades y cuellos de botella desde el principio.
- El equipo también enfrentó múltiples desafíos técnicos y organizativos. Un desafío fue la magnitud del trabajo: documentar 77 casos de prueba, ejecutar pruebas manuales y desarrollar scripts automatizados desde cero, lo cual demandó en un inicio una gestión eficiente del tiempo y de la carga de trabajo. La solución a esto fue distribuir el testing en sprints, priorizar flujos críticos y utilizar dashboards que permitieran visualizar el progreso real. Por último, la integración de herramientas como PHPUnit, Laravel Dusk, Selenium, JMeter y OWASP ZAP supuso un reto de aprendizaje que se superó mediante pruebas piloto y documentación compartida dentro del equipo.

- De forma global, el aprendizaje más significativo fue comprender que la calidad del software no depende únicamente de ejecutar pruebas, sino de tomar decisiones informadas, planificadas y sostenibles. El equipo aprendió que un plan de pruebas bien estructurado reduce el retrabajo, mejora la comunicación interna y aporta evidencia técnica para justificar decisiones.
- Se aprendió que automatizar sin estrategia conduce a resultados frágiles y difíciles de mantener, mientras que la automatización basada en riesgo es eficiente, escalable y alineada a los objetivos del negocio. Las métricas obtenidas como cobertura, tiempos de respuesta y mejora, y severidad de defectos demostraron el valor de medir de forma continua para evaluar avances reales. Finalmente, quedó claro que la calidad es un proceso permanente: en sistemas como un e-commerce, los cambios son constantes y solo un enfoque de mejora continua puede garantizar estabilidad y confianza en el producto.

9. Conclusiones

El desarrollo del plan integral de pruebas permitió fortalecer de manera significativa la calidad del sistema de comercio electrónico sometido a estudio. La aplicación conjunta de pruebas manuales y automatizadas evidenció que ambos enfoques se complementan y aportan valor en distintas fases del proceso: mientras las pruebas manuales facilitaron la identificación de aspectos visuales, de usabilidad y precisión funcional, la automatización aseguró repetibilidad, velocidad y mayor cobertura en los flujos críticos del sistema. Esta combinación metodológica hizo posible detectar defectos tempranamente, optimizar procesos internos y elevar la estabilidad del producto final.

La identificación y priorización de los flujos más utilizados por los usuarios permitió concentrar los esfuerzos en aquellos módulos cuyo funcionamiento resulta esencial para la operación diaria del comercio electrónico. Este enfoque estratégico permitió aprovechar de forma eficiente el tiempo, los recursos humanos y las herramientas disponibles. A través del diseño y documentación de casos de prueba estructurados y del desarrollo de scripts automatizados reutilizables, se construyó un repositorio sólido de validaciones que facilita futuras iteraciones de mantenimiento, actualizaciones o refactorización del sistema.

El análisis del rendimiento, la seguridad y la consistencia de la interfaz permitió observar el comportamiento del sistema en condiciones reales y bajo distintos niveles de carga. Las pruebas de estrés confirmaron tiempos de respuesta aceptables, mientras que el análisis de seguridad reveló vulnerabilidades que pudieron ser abordadas oportunamente. Esto contribuyó a mejorar la experiencia de navegación del usuario final, así como la confiabilidad del sistema frente a riesgos operativos y técnicos. Del mismo modo, la

evaluación de la interfaz permitió corregir inconsistencias y mejorar la coherencia visual y funcional del sistema.

El proceso de estimación de esfuerzos también aportó aprendizajes relevantes. La distribución de casos con más de 200 horas proyectadas y cerca de 18.5 días laborales de ejecución para el equipo mostró la magnitud del trabajo que implica garantizar calidad en un sistema en producción. Esta planificación permitió establecer tiempos realistas, gestionar el trabajo por etapas y anticipar contingencias relacionadas con ajustes técnicos, correcciones posteriores o re-ejecuciones necesarias para consolidar evidencia. La experiencia comprobó que la calidad no depende únicamente de ejecutar pruebas, sino de planificar cuidadosamente, estimar con precisión y ajustar los esfuerzos de acuerdo con la complejidad de cada flujo.

Finalmente, el análisis consolidado de resultados, defectos, métricas y evidencias permitió generar documentación útil para la toma de decisiones futuras. La sistematización de resultados no solo mostró el estado actual del sistema, sino que también dejó un conjunto de recomendaciones que servirán como base para fortalecer el mantenimiento, la evolución funcional y las prácticas de aseguramiento de calidad en versiones posteriores. En conjunto, el proyecto evidenció la importancia de integrar el testing como un proceso continuo, estructurado y sostenible, que contribuye directamente a la confianza del usuario final y a la estabilidad del sistema a largo plazo.

10. Carta de Salida

Proyecto: Plataforma CuscaTec – Sistema de Comercio Electrónico

Fecha: 27/11/2025

Responsable: Equipo de QA

I. Objetivo

La presente carta tiene como finalidad declarar la conclusión del proceso de pruebas ejecutado sobre la plataforma CuscaTec, dejando constancia del alcance cubierto, los resultados obtenidos y las condiciones bajo las cuales se recomienda avanzar a la siguiente fase operativa.

II. Alcance cubierto

De acuerdo con la matriz de seguimiento final, se evaluaron los módulos críticos del sistema:

- Login
- Products
- Orders
- Inventory
- Customers
- Users
- Category
- Purchase

Se aplicaron pruebas:

- Funcionales
- No funcionales
- Regresión por módulo
- Pruebas ejecutadas manualmente y mediante Selenium/Laravel Dusk/PHPUnit/OWAZ ZAP
- Pruebas automatizadas mediante despliegue continuo.
- Validaciones de UI, seguridad y consistencia de datos

III. Resultados

1. Métricas de ejecución

- Total de casos diseñados: 77

- Casos aprobados: 57
- Casos ejecutados con hallazgos: 11
- Casos fallidos: 9

2. Distribución porcentual real

- Aprobados: 74%
- Con hallazgos: 14%
- Fallidos: 12%

3. Estado cualitativo

- No se reportan defectos críticos.
- Los fallos registrados corresponden a comportamientos funcionales corregibles.
- Los hallazgos documentados se consideran de bajo impacto en la operación principal.
- La totalidad de los casos fueron ejecutados, documentados y trazables.

4. Limitaciones

- Durante las pruebas se identificaron las siguientes limitaciones:
- Algunos casos con dependencias visuales o de entorno no pudieron automatizarse.
- Persisten hallazgos menores relacionados con interfaz y consistencia de mensajes.
- Los módulos no críticos se mantuvieron fuera del alcance (según plan original).

IV. Conclusión

De acuerdo con los resultados registrados en la matriz de control, el sistema demuestra un comportamiento estable en sus módulos centrales, sin fallas críticas que comprometan los flujos primarios de negocio.

Los fallos detectados son de severidad media o baja, y pueden ser abordados en próximas iteraciones sin impedir la operación.

El sistema cumple con los criterios de salida establecidos, por lo que se recomienda avanzar a la siguiente fase operativa, manteniendo seguimiento a los casos fallidos y hallazgos documentados.

11. Referencias

IEEE. (2013). *ISO/IEC/IEEE 29119-2:2013 – Software and systems engineering — Software testing — Part 2: Test processes*. IEEE.

Migration. (2025, 26 agosto). *Cómo Implementar Laravel 8 en Hostinger - Centro de ayuda de Hostinger*. Centro de Ayuda de Hostinger. <https://www.hostinger.com/es/support/6152127-como-implementar-laravel-8-en-hostinger/>

ZAP – *Getting started*. (s. f.). <https://www.zaproxy.org/getting-started/>

ISTQB Glossary. (s. f.). *All terms*. [istqb.org. https://glossary.istqb.org/en_US/search?term=&exact_matches_first=true&page=4](https://glossary.istqb.org/en_US/search?term=&exact_matches_first=true&page=4)

LaRavel Dusk - Laravel 12.X - The PHP framework for web artisans. (s. f.). <https://laravel.com/docs/12.x/dusk>

12. Anexos

12.1. Anexo 1: Resultados del análisis estático de código

El análisis estático realizado sobre el código fuente generó un conjunto de hallazgos clasificados por tipo, severidad y número de ocurrencias. Estos resultados permiten identificar bugs, code smells y vulnerabilidades que afectan la calidad, mantenibilidad y seguridad del sistema. La tabla resume los principales problemas detectados, organizados en categorías que reflejan:

- **Errores de implementación (bugs):** situaciones donde el comportamiento del código puede ser incorrecto o no cumplir con los estándares definidos.
- **Problemas de calidad (code smells):** prácticas que no generan fallos inmediatos, pero comprometen la legibilidad, mantenibilidad y escalabilidad del software.
- **Vulnerabilidades:** riesgos de seguridad que pueden ser explotados y requieren atención prioritaria.

| Name | Type | Severity | Number |
|--|------|----------|--------|
| "<!DOCTYPE>" declarations should appear before "<html>" tags | BUG | MAJOR | 1 |
| "<html>" element should have a language attribute | BUG | MAJOR | 5 |
| Tables should have headers | BUG | MAJOR | 20 |
| "<th>" tags should have "id" or "scope" attributes | BUG | MAJOR | 69 |
| Elements deprecated in HTML5 should not be used | BUG | MAJOR | 2 |
| Font declarations should contain at least one generic font family | BUG | MAJOR | 13 |
| Properties should not be duplicated | BUG | MAJOR | 5 |
| Member names should not be duplicated within a class or object literal | BUG | MAJOR | 2 |
| Objects should not be created to be dropped immediately without being used | BUG | MAJOR | 3 |
| Non-empty statements should change control flow or have at least one side-effect | BUG | MAJOR | 2 |

Tabla 153. Issues List – SonarQube (Parte I).

| Name | Type | Severity | Number |
|---|------------|----------|--------|
| All code should be reachable | BUG | MAJOR | 7 |
| Mouse events should have corresponding keyboard events | BUG | MINOR | 158 |
| "require_once" and "include_once" should be used instead of "require" and "include" | BUG | MINOR | 7 |
| Variables should be declared explicitly | CODE_SMELL | BLOCKER | 9 |
| Variable variables should not be used | CODE_SMELL | BLOCKER | 5 |
| Constructor names should start with an upper case letter | CODE_SMELL | CRITICAL | 5 |
| Variables should be declared with "let" or "const" | CODE_SMELL | CRITICAL | 279 |
| Cognitive Complexity of functions should not be too high | CODE_SMELL | CRITICAL | 1 |
| String literals should not be duplicated | CODE_SMELL | CRITICAL | 64 |
| Control structures should use curly braces | CODE_SMELL | CRITICAL | 12 |
| "switch" statements should have "default" clauses | CODE_SMELL | CRITICAL | 2 |
| Cognitive Complexity of functions should not be too high | CODE_SMELL | CRITICAL | 6 |
| A conditionally executed single line should be denoted by indentation | CODE_SMELL | CRITICAL | 22 |
| Unnecessary parentheses should not be used for constructs | CODE_SMELL | CRITICAL | 19 |
| String literals should not be duplicated | CODE_SMELL | CRITICAL | 2 |
| Track uses of "TODO" tags | CODE_SMELL | INFO | 1 |
| Track uses of "TODO" tags | CODE_SMELL | INFO | 1 |
| Attributes deprecated in HTML5 should not be used | CODE_SMELL | MAJOR | 25 |
| HTML "<table>" should not be used for layout purposes | CODE_SMELL | MAJOR | 4 |

Tabla 154. Issues List – SonarQube (Parte II).

| Name | Type | Severity | Number |
|--|------------|----------|--------|
| DOM elements with ARIA role should only have supported properties | CODE_SMELL | MAJOR | 42 |
| Prefer tag over ARIA role | CODE_SMELL | MAJOR | 274 |
| DOM elements with ARIA roles should have a valid non-abstract role | CODE_SMELL | MAJOR | 3 |
| Focusable elements should not have "aria-hidden" attribute | CODE_SMELL | MAJOR | 1 |
| Non-interactive DOM elements should not have interactive ARIA roles | CODE_SMELL | MAJOR | 8 |
| Anchor tags should not be used as buttons | CODE_SMELL | MAJOR | 8 |
| Non-interactive DOM elements should not have the `tabIndex` property | CODE_SMELL | MAJOR | 2 |
| Heading elements should have accessible content | CODE_SMELL | MAJOR | 1 |
| Images should have a non-redundant alternate description | CODE_SMELL | MAJOR | 82 |
| Label elements should have a text label and an associated control | CODE_SMELL | MAJOR | 63 |
| Sections of code should not be commented out | CODE_SMELL | MAJOR | 162 |
| Selectors should not be duplicated | CODE_SMELL | MAJOR | 12 |
| CSS files should not be empty | CODE_SMELL | MAJOR | 1 |
| Double quote to prevent globbing and word splitting | CODE_SMELL | MAJOR | 20 |
| Sections of code should not be commented out | CODE_SMELL | MAJOR | 2 |
| Unused assignments should be removed | CODE_SMELL | MAJOR | 17 |
| Variables should be used in the blocks where they are declared | CODE_SMELL | MAJOR | 5 |
| Variables and functions should not be redeclared | CODE_SMELL | MAJOR | 93 |
| Ternary operators should not be nested | CODE_SMELL | MAJOR | 6 |

Tabla 155. Issues List – SonarQube (Parte III).

| Name | Type | Severity | Number |
|--|------------|----------|--------|
| Sparse arrays should not be created with extra commas | CODE_SMELL | MAJOR | 2 |
| If statements should not be the only statement in else blocks | CODE_SMELL | MAJOR | 1 |
| Comma operator should not be used | CODE_SMELL | MAJOR | 2 |
| Generic exceptions <code>ErrorException</code> , <code>RuntimeException</code> and <code>Exception</code> should not be thrown | CODE_SMELL | MAJOR | 6 |
| Unused function parameters should be removed | CODE_SMELL | MAJOR | 2 |
| Sections of code should not be commented out | CODE_SMELL | MAJOR | 18 |
| Functions should not have too many lines of code | CODE_SMELL | MAJOR | 2 |
| Classes should not have too many methods | CODE_SMELL | MAJOR | 1 |
| Unused assignments should be removed | CODE_SMELL | MAJOR | 4 |
| Use of namespaces should be preferred to "include" or "require" functions | CODE_SMELL | MAJOR | 8 |
| The "print" statement should not be used | CODE_SMELL | MAJOR | 1 |
| Sections of code should not be commented out | CODE_SMELL | MAJOR | 2 |
| Image, area and button with image elements should have an "alt" attribute | CODE_SMELL | MINOR | 10 |
| Anchors should contain accessible content | CODE_SMELL | MINOR | 2 |
| Extra semicolons should be removed | CODE_SMELL | MINOR | 1 |
| Arguments in long RUN instructions should be sorted | CODE_SMELL | MINOR | 8 |
| Too long RUN instruction should be split into multiple lines | CODE_SMELL | MINOR | 4 |
| Reduce the amount of consecutive RUN instructions | CODE_SMELL | MINOR | 4 |
| Unused local variables and functions should be removed | CODE_SMELL | MINOR | 18 |
| "for of" should be used with Iterables | CODE_SMELL | MINOR | 2 |

Tabla 156. Issues List – SonarQube (Parte IV).

| Name | Type | Severity | Number |
|---|---------------|----------|--------|
| Empty statements should be removed | CODE_SMELL | MINOR | 3 |
| Boolean literals should not be redundant | CODE_SMELL | MINOR | 1 |
| Files should end with a newline | CODE_SMELL | MINOR | 87 |
| Lines should not end with trailing whitespaces | CODE_SMELL | MINOR | 122 |
| Field names should comply with a naming convention | CODE_SMELL | MINOR | 1 |
| Unused local variables should be removed | CODE_SMELL | MINOR | 12 |
| More than one property should not be declared per statement | CODE_SMELL | MINOR | 1 |
| Closing tag ">" should be omitted on files containing only PHP | CODE_SMELL | MINOR | 7 |
| "elseif" keyword should be used in place of "else if" keywords | CODE_SMELL | MINOR | 2 |
| Regular expression quantifiers and character classes should be used concisely | CODE_SMELL | MINOR | 2 |
| Google API keys should not be disclosed | VULNERABILITY | BLOCKER | 1 |

Tabla 157. Issues List – SonarQube (Parte V).

12.2. Anexo 2: Análisis de caja negra

El sistema fue desarrollado y es mantenido por un único programador independiente, sin procesos formales de aseguramiento de calidad. Aunque está en producción y funciona, presenta baja cobertura de pruebas, limitadas a ejecuciones manuales sin documentación, automatización ni métricas, lo que incrementa el riesgo operativo ante cambios o crecimiento. Para mitigar esta situación se propone aplicar el método de caja negra, evaluando entradas y salidas sin acceder al código, lo que permite validar funcionalidad, rendimiento y experiencia del usuario aun sin documentación técnica disponible

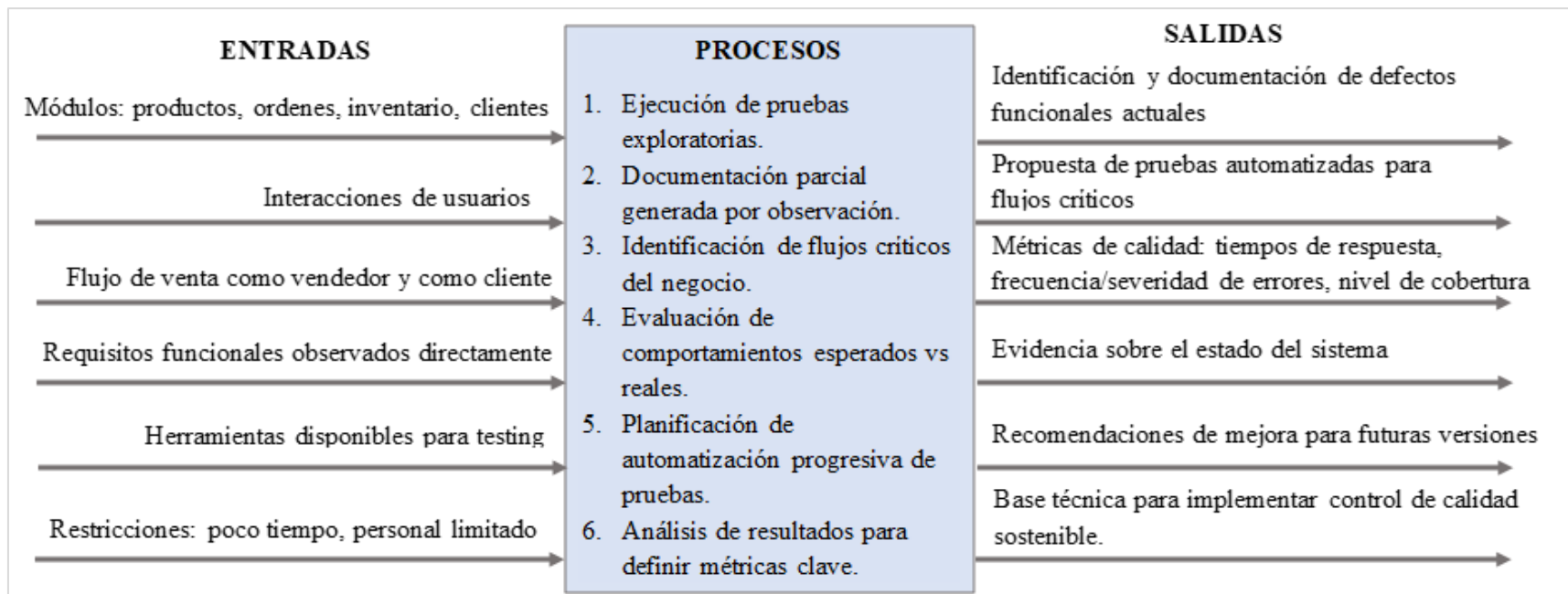


Ilustración 26. Diagrama de la caja negra.

12.3. Anexo 3: Diagramas y flujos del sistema

El siguiente diagrama de componentes ilustra la arquitectura fundamental de un sistema, detallando la interacción entre sus elementos clave. Este diseño representa una aproximación común para aplicaciones web robustas, y se visualiza cómo se interactúa con la interfaz. Tras realizar un análisis del sistema, se constató que el cliente no utiliza pasarelas de pago. En su lugar, el flujo de pago de las órdenes se gestiona mediante los estados Pending (selección del tipo de pago pendiente), Check (cheque) y Cash (efectivo). En cuanto a la factura, esta no es enviada de manera automática al realizar una orden y es generado mediante CSS y HTML para luego realizar su descarga.

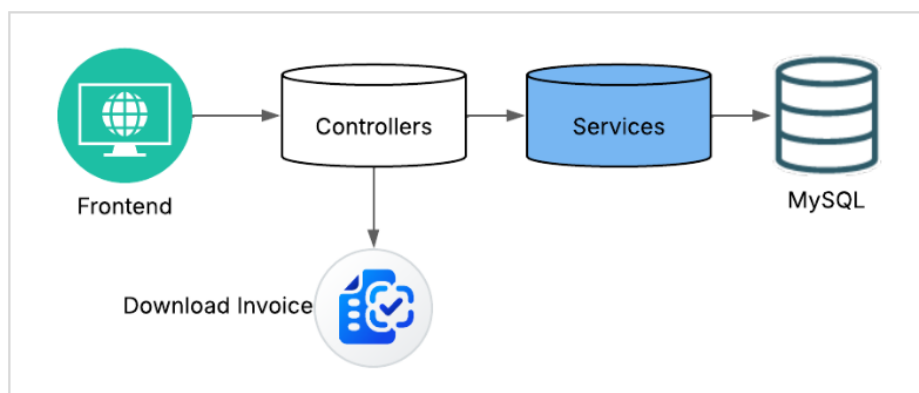


Ilustración 27. Diagrama de la arquitectura del sistema.

El siguiente diagrama presenta la estructura modular del sistema de comercio electrónico, organizando sus funcionalidades consideradas clave en componentes esenciales. El diseño refleja los procesos centrales que permiten gestionar productos, pedidos, inventario, usuarios, categoría y ordenes, garantizando una operación fluida de la plataforma.

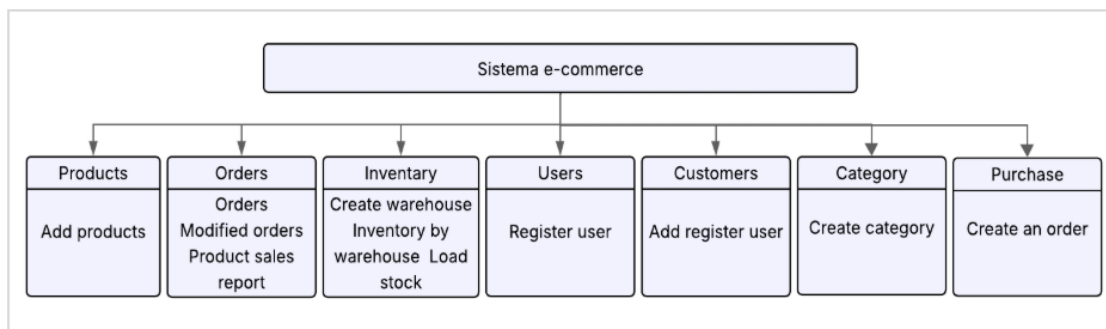


Ilustración 28. Diagrama de los módulos del sistema

- Flujo de vista Customers:** Se describe cómo el sistema gestiona a los clientes, permitiendo filtrarlos mediante búsqueda o estado, así como crear nuevos registros desde un formulario dedicado. Además, ofrece acciones para editar la información de un cliente o desactivarlo en la base de datos, asegurando un control organizado y actualizado de la información.

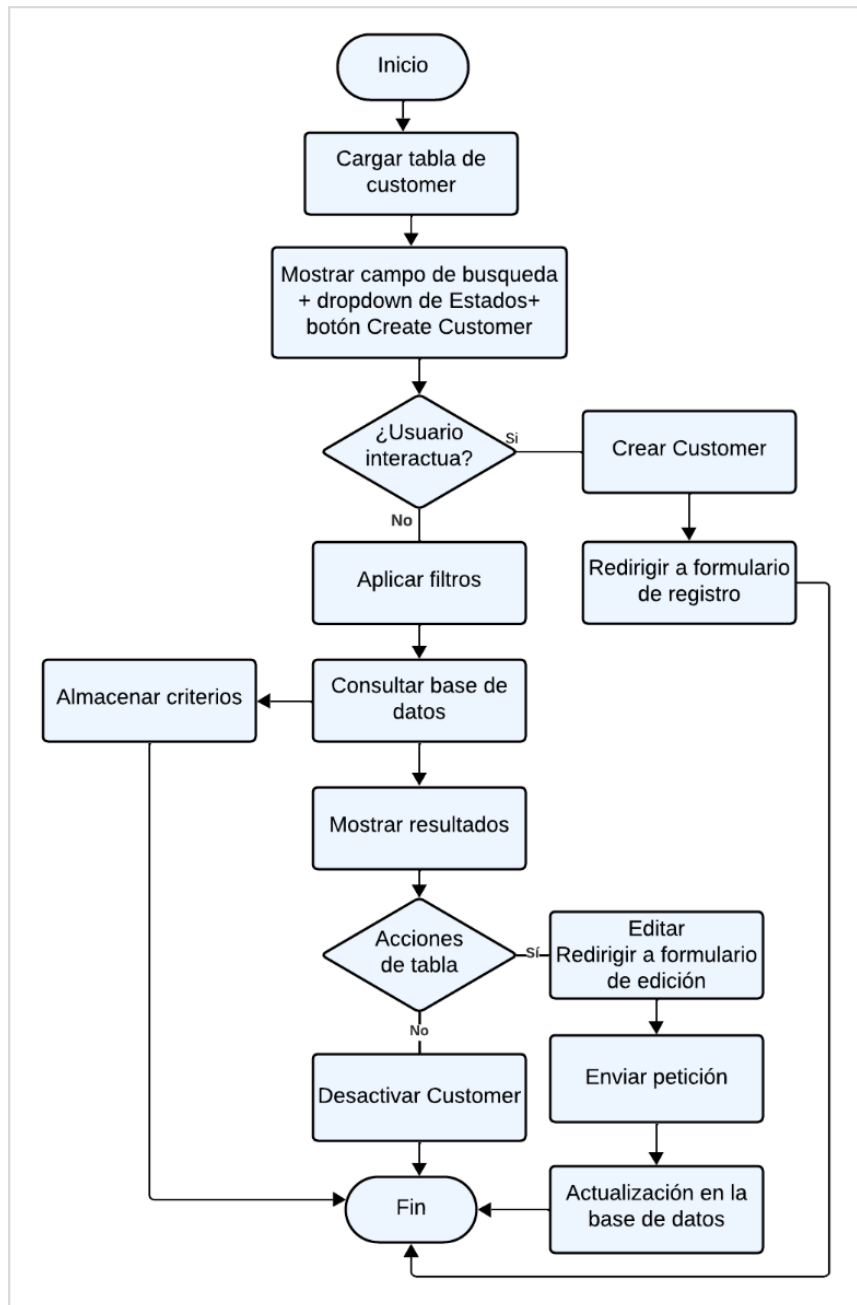


Ilustración 29. Flujo del módulo Customers

- Flujo de modulo Products:** Se describe cómo el sistema administra los productos mediante la pestaña Active y la pestaña Trashed. Permite realizar búsquedas, agregar nuevos productos, editar información, actualizar precios y enviar productos a la papelera, así como restaurarlos en caso necesario. De esta forma se asegura que la información de inventario se mantenga organizada y siempre actualizada.

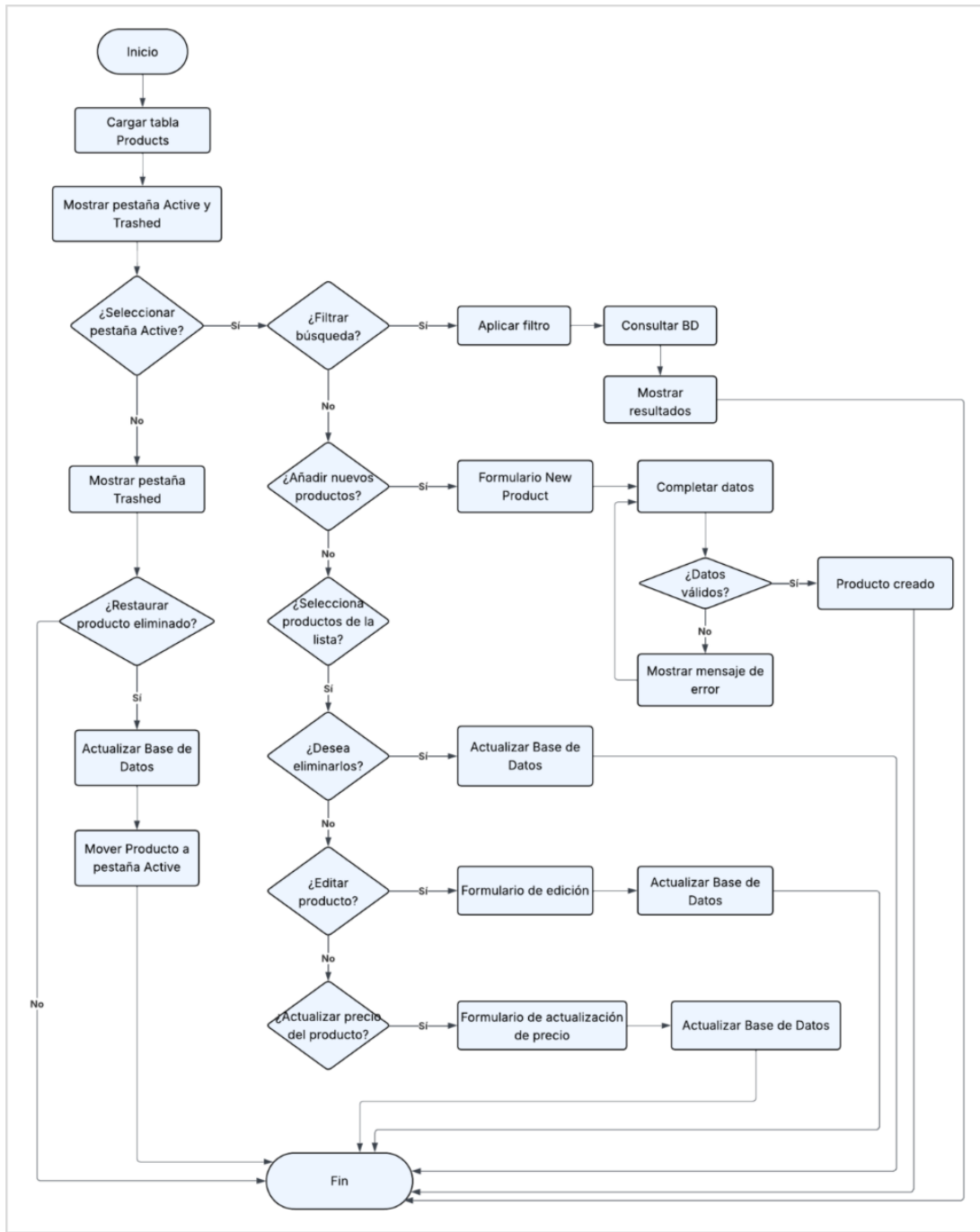


Ilustración 30. Flujo del módulo Products

- **Flujo de vista Users:** Se describe cómo el sistema permite administrar a los usuarios registrados, abarcando desde la carga inicial de la tabla hasta las acciones de creación, edición y retorno a la vista principal. Al ingresar al módulo, el sistema presenta una tabla con los usuarios existentes, permitiendo al usuario realizar operaciones sobre ellos de forma controlada y validada.

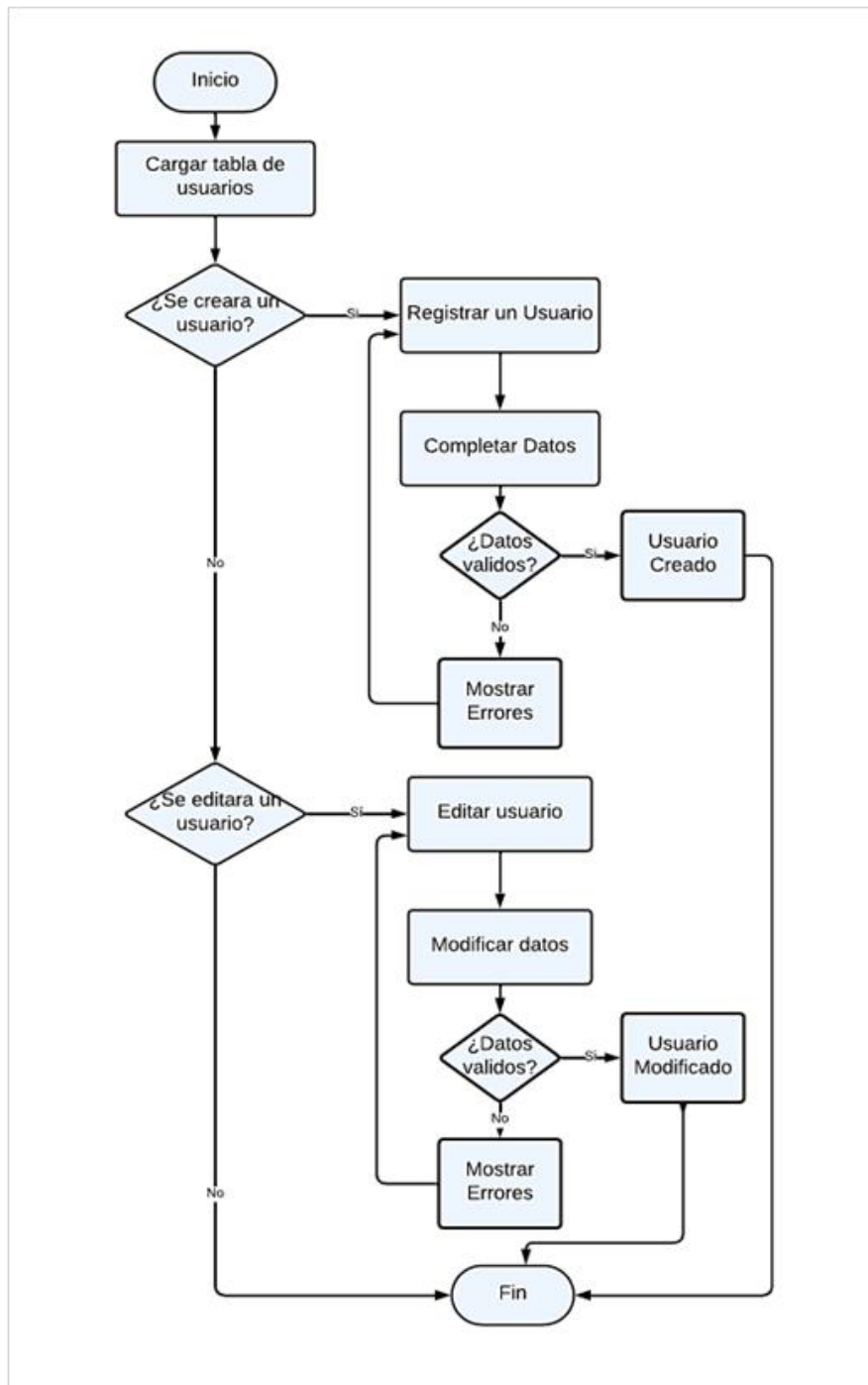


Ilustración 31. Flujo del módulo Users

- Flujo de vista Orders:** Se describe cómo el sistema administra las órdenes desde su carga inicial en la tabla hasta las diferentes acciones que el usuario puede realizar. Permite aplicar filtros, consultar el detalle de una orden, editar información, envíos y estados, generar la orden en PDF o desactivarla.

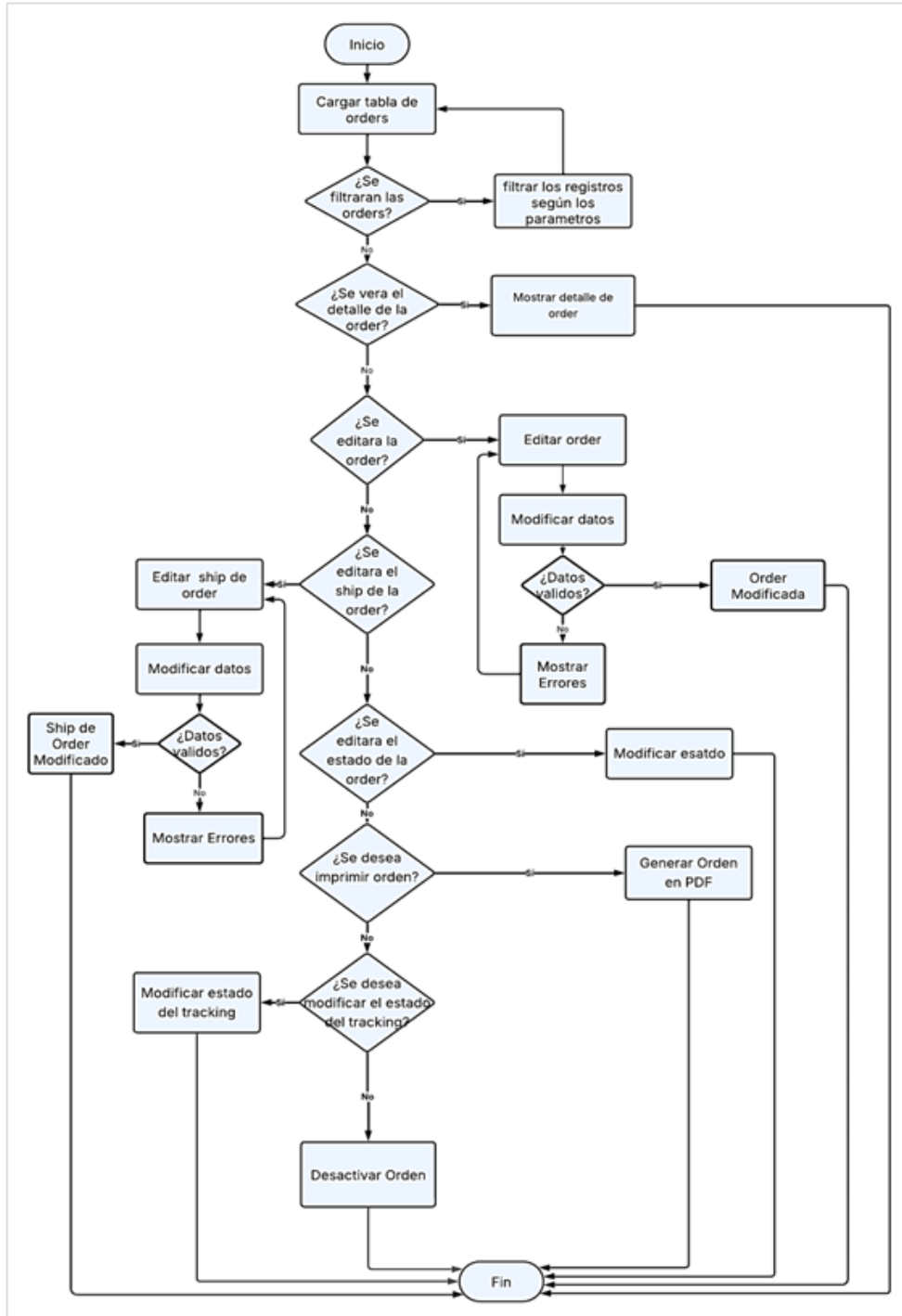


Ilustración 32. Flujo del módulo Orsers

- Flujo de vista Inventory:** Se describe cómo el sistema gestiona el inventario y los almacenes (warehouses), permitiendo crear nuevos almacenes, validar su información y administrar las existencias. Además, ofrece la posibilidad de filtrar y exportar el inventario en Excel, consultar el kardex de productos en un periodo específico y actualizar existencias mediante carga de archivos.

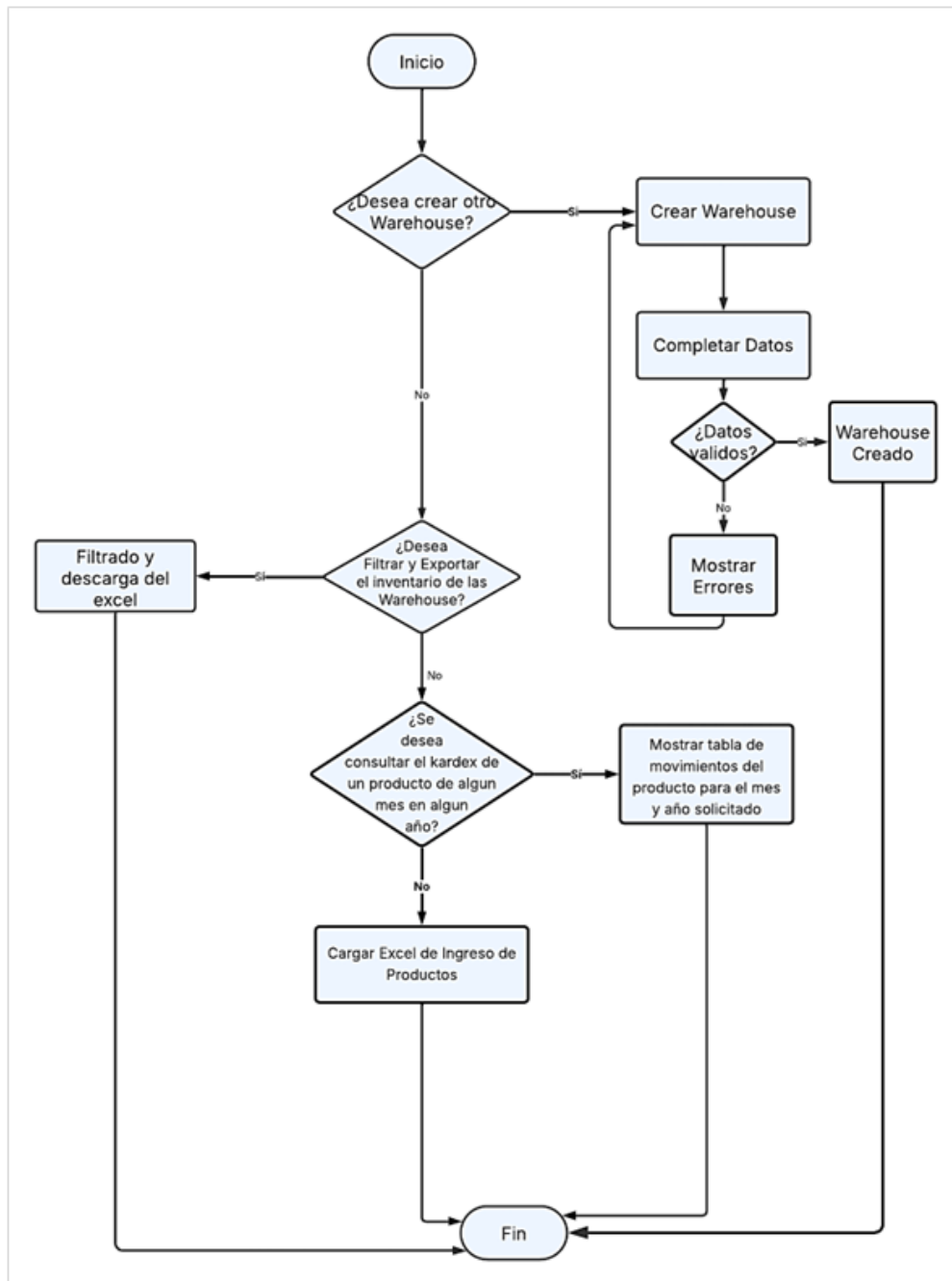


Ilustración 33. Flujo del módulo Inventory.

- Flujo de vista Category:** Se describe cómo el sistema permite gestionar las categorías de productos, ofreciendo funcionalidades para crear, modificar y eliminar registros de forma controlada. El proceso inicia con la visualización de opciones disponibles, seguido por decisiones del usuario que determinan el tipo de acción a realizar. Al crear una categoría, se valida la integridad de los datos ingresados.

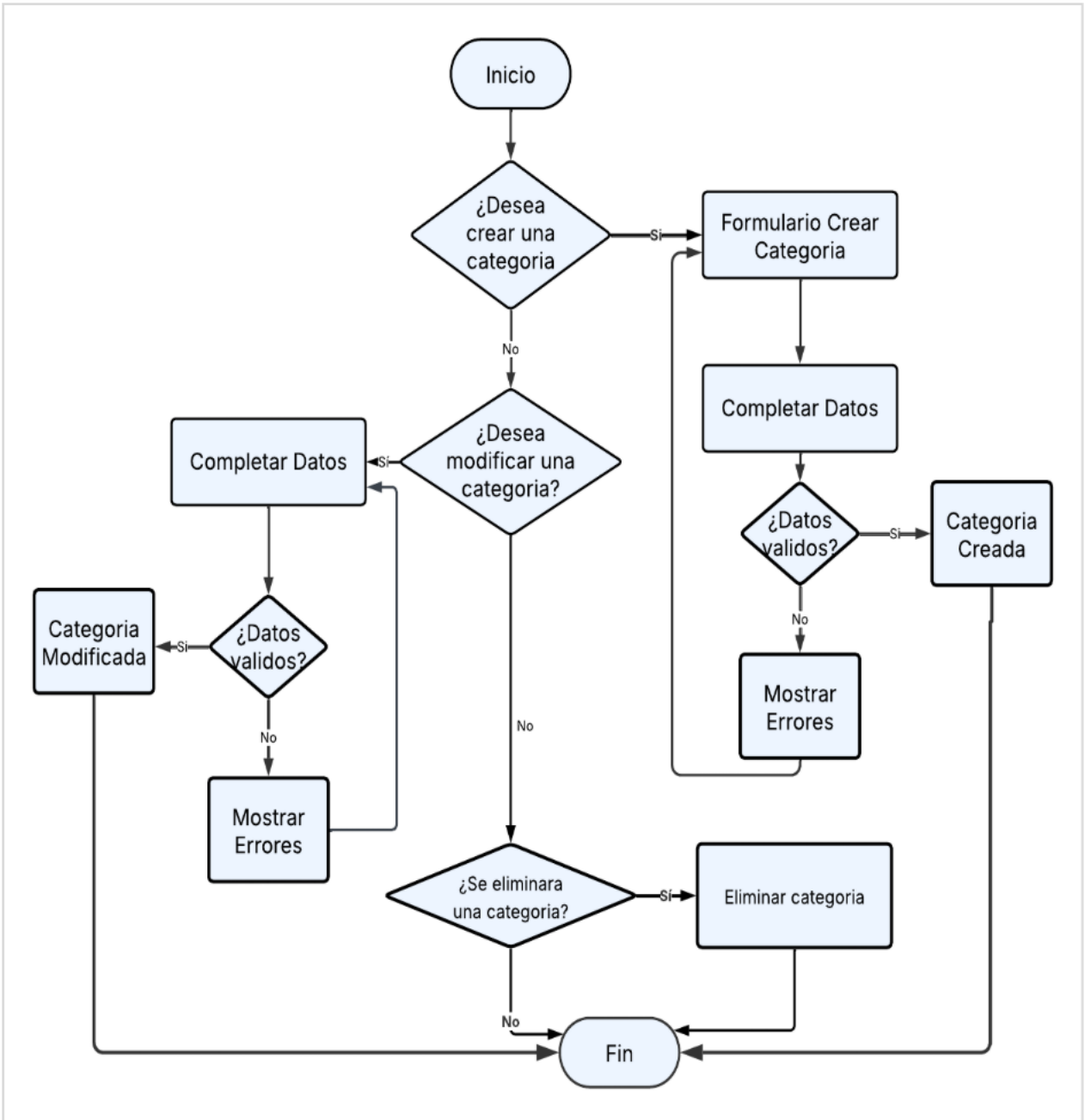


Ilustración 34. Flujo del módulo Category.

- **Flujo de vista Purchase:** Se describe cómo el sistema permite al usuario realizar una compra en el sistema, comenzando con la visualización de categorías y productos, seguido por la selección de artículos y el registro de los datos necesarios para generar una orden. Durante el proceso, se validan los datos ingresados para garantizar la integridad de la información.

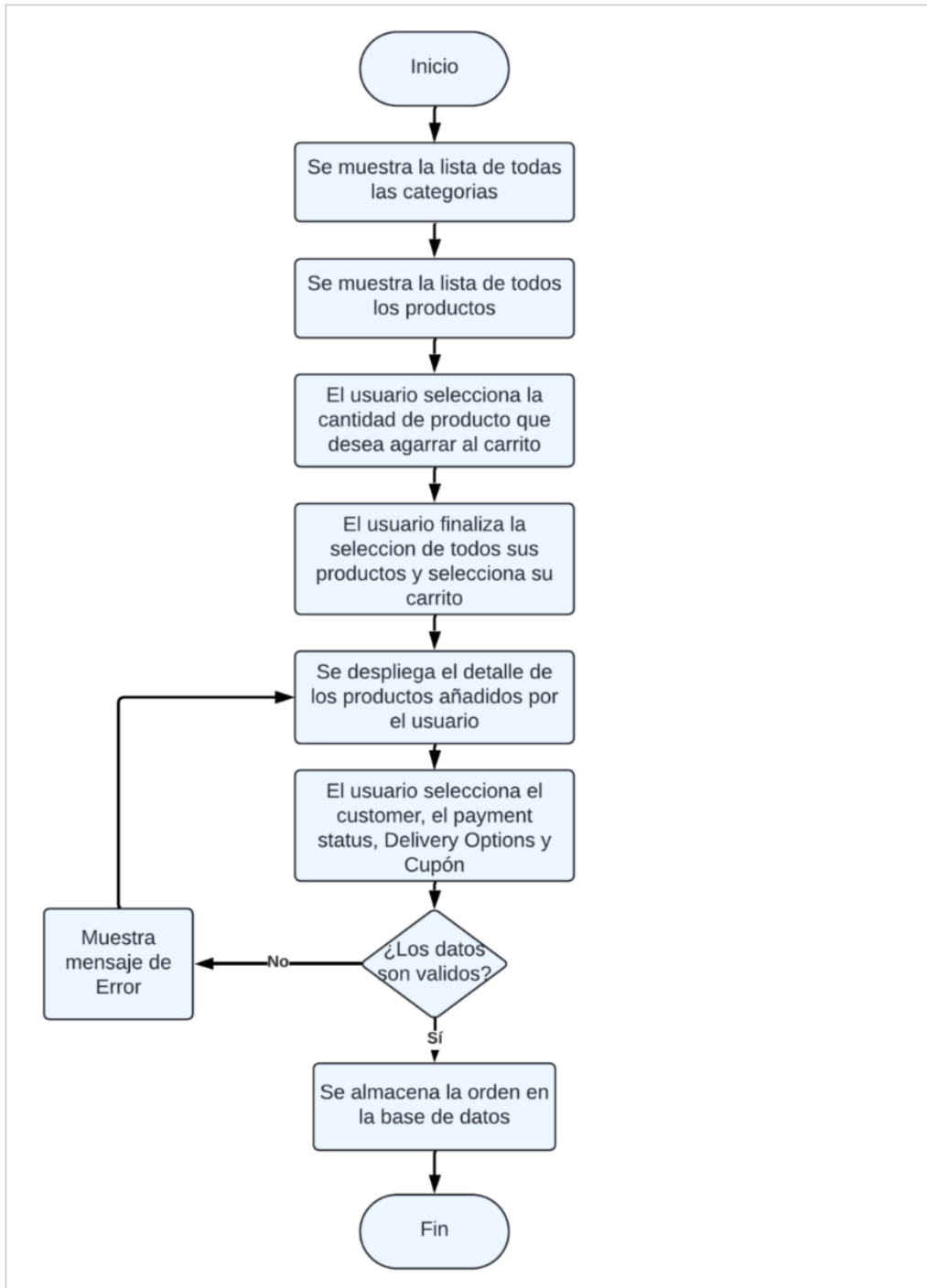


Ilustración 35. Flujo del módulo crear una orden.

12.4. Anexo 4: Requerimientos, casos de prueba y resultados

Se muestra a continuación dos tablas complementarias: una con los requerimientos identificados y otra con los resultados de las pruebas:

| ID | Módulo | Sub módulo | Descripción | Rol |
|-----|----------|------------|--|----------------|
| L01 | Login | - | Formulario con campos “Email” y “Password”; ambos obligatorios; email válido | Todos |
| L02 | Login | - | Botón “Iniciar Sesión” habilitado solo cuando ambos campos estén completos | Todos |
| L03 | Login | - | Mensaje de error “Email o contraseña incorrectos”; se muestra en rojo sin recargar página | Todos |
| L04 | Login | - | Redirección según rol: Admin → /login, Seller → /ecommerce/seller/login, Customer → /customer-ecommerce/seller/login | Todos |
| P01 | Products | - | Visualización de productos en tabla con columnas: ID, Name, Category, Stock, Last update. Debe incluir paginación | Admin |
| P02 | Products | - | Botón “Add New Product” redirecciona a página de creación con campos: Name, Description, Category, Image. Validación de campos e imagen. | Admin |
| P03 | Products | - | Cada fila incluye acciones: Edit, Move to Trash, Manage Prices. Edit y Manage Prices redireccionan; Trash confirma eliminación. | Admin |
| P04 | Products | - | Barra de búsqueda con texto libre para filtrar productos. Búsqueda insensible a mayúsculas. | Admin |
| P05 | Products | - | Botón “Save” valida que todos los campos estén completos y correctos. Confirmación rápida tras guardar. | Admin |
| P06 | Products | - | “Move to Trash” mueve producto a pestaña “Trashed”. Producto desaparece de “Active” y aparece en “Trashed”. | Admin |
| O01 | Orders | Orders | Permite filtros simultáneos: Customer, Seller, Date, Payment Status, Delivery Status, Origin, Product. | Admin y Seller |

Tabla 158. Requerimientos del sistema (Parte I).

| ID | Módulo | Sub módulo | Descripción | Rol |
|------|-----------|------------------------|---|----------------|
| O02 | Orders | Orders | Tabla dinámica con columnas: ID, Customer, Total, Origen, Fecha, Estado y Tracking. Incluye paginación y carga óptima al aplicar filtros. | Admin y Seller |
| O03 | Orders | Orders | Acciones por fila: Detail, Edit Order, Edit & Ship, Update Order Status, Print, Tracking, Trash. Algunas abren modal o redireccionan. Impresión genera PDF. | Admin y Seller |
| O04 | Orders | Orders | Botones globales: Filter, Clear, Export a Excel con datos visibles. | Admin y Seller |
| O05 | Orders | Orders | Solo de lectura. Filtros y detalle de orden. | Customer |
| MO01 | Orders | Modified Orders | Permite filtros simultáneos: Customer, Seller, rango de fechas, Payment, Delivery Status, Origin y Product. Funcionalidad idéntica a Ordenes. | Admin y Seller |
| MO02 | Orders | Modified Orders | Acciones por fila: “Order Detail” y “Print Order”. Impresión genera PDF con formato y datos completos. | Admin y Seller |
| SR01 | Orders | Product Sales Report | Implementación de filtros: Select User, Start Date, End Date. Fecha en formato dd/mm/yyyy. | Admin |
| SR02 | Orders | Product Sales Report | Mostrar tabla de ventas. Total de productos vendidos debe mostrarse correctamente. | Admin |
| SR03 | Orders | Product Sales Report | Botones: Filter, Export to Excel. Exportación genera archivo .xlsx con datos visibles. | Admin |
| I01 | Inventory | Warehouse | Botón “Create Warehouse” con campos: Name, Address, State. Todos los campos son obligatorios. | Admin |
| I02 | Inventory | Warehouse | Tabla con listado de bodegas registradas. Muestra: ID, Name, Address, State. No se podrá editar la información. | Admin |
| I03 | Inventory | Warehouse | Botones globales: “Filter” y “Clear” en la vista. “Clear” limpia todos los filtros aplicados. | Admin |
| I04 | Inventory | Inventory by Warehouse | Tabla de inventario por almacenes. Incluir paginación. | Admin |
| I05 | Inventory | Inventory by Warehouse | Botones: Filter, download Excel. Exportación genera archivo .xlsx con datos visibles. | Admin |

Tabla 159. Requerimientos del sistema (Parte II).

| ID | Módulo | Sub módulo | Descripción | Rol |
|-------------|-----------|------------|--|----------------|
| I06 | Inventory | Kardex | Filtros: Product, Month, Year. Validación estricta: mes entre 1 y 12; año en formato YYYY. | Admin |
| I07 | Inventory | Kardex | Tabla de historial de entradas y salidas de inventario. Retorna datos precisos de movimientos y totales. | Admin |
| I08 | Inventory | Load Stock | Selección de warehouse y carga de archivo Excel. Incluir columnas: warehouse, product, cantidad y date. | Admin |
| I09 | Inventory | Load Stock | Botones: Cargar, Descargar Plantilla Excel. Descarga genera archivo .xlsx con encabezados de productos cargados. | Admin |
| C01 | Customer | - | Registro de cliente con los campos: Name, Phone, Address, Email, Password, State. Todos los campos son requeridos. | Admin y Seller |
| C02 | Customer | - | Tabla con clientes del mismo estado. Acciones para Editar, cambiar contraseña, deshabilitar. | Admin y Seller |
| CO3 | Customer | - | Envío automático de credenciales por email tras el registro exitoso del cliente. | Admin y Seller |
| CO4 | Customer | - | Generación automática de contraseña segura al registrar un nuevo cliente. | Admin y Seller |
| CO5 | Customer | - | Filtro de clientes por Estado geográfico y búsqueda combinada por nombre + Estado. | Admin y Seller |
| U01 | Users | - | Botón “Crear User” con campos obligatorios: Name, Email, Password, Role. Todos los campos son requeridos. | Admin |
| U02 | Users | - | Campo “Password” con funcionalidades: generación automática de contraseña y opción para ocultar caracteres. | Admin |
| U03 | Users | - | Tabla de usuarios. Muestra: ID, Name, Email, Role. Incluye acción “Edit” por fila. | Admin |
| U04 | Users | - | Campo “Role” con opciones limitadas. Solo permite seleccionar “Administrador” o “Bodeguero”. | Admin |
| CA01 | Category | - | Botón “Crear Category” con campos obligatorios: Name, Description. Validar que sean obligatorios y únicos. | Admin |
| CA02 | Category | - | Permitir subir imagen representativa de la categoría. Validar formato (JPG, JPEG, PNG, GIF). | Admin |

Tabla 160. Requerimientos del sistema (Parte III).

| ID | Módulo | Sub módulo | Descripción | Rol |
|------|----------|------------|--|-------------------|
| CA03 | Category | - | Botón “Create Category” registra la nueva categoría. Muestra ID, Name, Email, Role. Incluye acción “Edit” por fila. | Admin |
| CA04 | Category | - | Botón “Back to List”. Validar que al regresar no se pierda integridad de datos en la lista. | Admin |
| P01 | Purchase | - | Al iniciar sesión, se muestra la pantalla de categorías. Validar que la vista inicial corresponda a categorías. | Seller / Customer |
| P02 | Purchase | - | Al seleccionar una categoría, se muestran los productos asociados. | Seller / Customer |
| P03 | Purchase | - | Se puede agregar o remover productos del carrito. Validar que el stock permita agregar y remover productos | Seller / Customer |
| P04 | Purchase | - | Validar que los cálculos sean precisos. | Seller / Customer |
| P05 | Purchase | - | Desde la vista del Seller, se selecciona el estado de pago. Validar que se actualice y sea consistente en facturación. | Seller |

Tabla 161. Requerimientos del sistema (Parte IV).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|--------|---------------|--------------------------------------|--|---|---|---|-------------------------|
| CP01 | Login | L01, L02 | Acceso exitoso | Verificar que el sistema permita acceso con credenciales válidas | El sistema debe permitir el acceso únicamente cuando se ingresen credenciales válidas registradas en la base de datos | El sistema permitió el ingreso exitoso del usuario con las credenciales válidas registradas en la base de datos | El sistema debería de tener un solo login independientemente el tipo de usuario | Aprobado |
| CP02 | Login | L03 | Manejo de credenciales incorrectas | Verificar que el sistema muestre error con credenciales inválidas | El sistema debe rechazar el intento de acceso, mostrando un mensaje de error que indique credenciales inválidas. El usuario no debe ingresar al sistema ni acceder a la interfaz principal. | Se muestra un mensaje de que las credenciales no tienen un match en los registros y no ingresa al sistema. Cumple con lo esperado | N/A | Aprobado |
| CP03 | Login | L01 | Campos obligatorios vacíos del login | Validar comportamiento con campos vacíos | Validaciones frontend bloquean el envío | El formulario se envía y carga nuevamente la vista de login con un mensaje de error de campos requeridos | El frontend no valida que los campos se encuentren debidamente llenos antes del envío, por lo que la request de campos vacíos se envía al servidor y luego regresa la request y muestra el mensaje de error | Ejecutado con hallazgos |
| CP04 | Login | L01, Reglas 4 | Bloqueo por intentos fallidos | Verifica que la cuenta sea bloqueada luego de varios intentos fallidos | El sistema debe bloquear la cuenta del usuario después de varios intentos fallidos consecutivos, impidiendo nuevos accesos | La cuenta no se bloquea por más intentos fallidos de inicio de sesión con un usuario válido | Crear un espacio donde se registren los intentos fallidos con un usuario registrado en la base de datos y al pasar de 3 bloquee la cuenta | Fallido |
| CP05 | Login | L04 | Redirección por rol | Validar enrutamiento según tipo de usuario | El sistema debe redirigir al usuario autenticado hacia la URL o vista definida para su rol específico, garantizando que cada tipo de usuario acceda únicamente a las funcionalidades que le corresponden. | Redirecciona a la página principal esperada para cada tipo de usuario | El sistema redirigió correctamente al usuario hacia la página principal esperada para su rol, confirmando que el enrutamiento por rol funciona de manera adecuada y consistente. No se presentaron errores ni accesos indebidos a vistas de otros roles | Aprobado |

Tabla 162. Matriz de resultados (Parte I).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|----------|---------------|---------------------------------------|--|---|--|--|-------------------------|
| CP06 | Login | L04 | Redirección por logueo | Una vez logueado en el sistema no debe poder ver ningún login hasta cerrar sesión | El sistema debe redirigir automáticamente al usuario hacia la página de inicio correspondiente a su rol, impidiendo el acceso a la vista de login mientras la sesión esté activa | El usuario administrador si cumple con esta condición, sin embargo, los usuarios tipo seller y customer si pueden regresar al login sin haber cerrado su sesión. | Se debería tener un solo login no importando el tipo de usuario que ingrese. | Ejecutado con hallazgos |
| CP07 | Products | P02, P05 | Crear producto válido | Validar que el sistema permita registrar un producto con todos los campos obligatorios | El sistema debe registrar correctamente un producto cuando todos los campos obligatorios han sido completados, mostrando un mensaje de confirmación. | El sistema permitió guardar el producto de manera satisfactoria, desplegando el mensaje de confirmación y reflejando el registro en la tabla de productos. | | Aprobado |
| CP08 | Products | P02 | Agregar Imagen en diferentes formatos | Validar formatos permitidos en carga de imagen | Imagen aceptada o sin imagen, con formato valido | Se aceptan archivos únicamente en formato de imagen, de lo contrario, arroja mensaje de error. | | Aprobado |
| CP09 | Products | P03 | Editar producto existente | Validar edición de producto previamente registrado | El sistema debe permitir la edición de un producto existente, validando que todos los campos obligatorios se completen antes de guardar. Al finalizar, el producto debe actualizarse en la base de datos y reflejarse en tabla. | El sistema procesó la edición del producto de manera satisfactoria, y actualizando la información en la lista de productos. | N/A | Aprobado |

Tabla 163. Matriz de resultados (Parte II).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|----------|---------------|--------------------------------------|---|---|--|---|---------------------|
| CP10 | Products | P04 | Búsqueda de productos | Evaluar búsqueda con texto parcial o incompleto | El sistema debe mostrar en la lista los productos relacionados con el texto parcial o incompleto ingresado en la barra de búsqueda. La funcionalidad debe responder de manera ágil y sin errores. | Se muestran los productos que incluyen los caracteres ingresados en la búsqueda. No se presentaron fallos ni inconsistencias durante la ejecución. | N/A | Aprobado |
| CP11 | Products | P06 | Envío de producto a estado "Trashed" | Verificar que al seleccionar "Move to Trash" el producto se desactive lógicamente | El producto seleccionado es movido a estado "Trashed". El mensaje de confirmación de acción se muestra y la operación se completa sin errores. | Producto movido a trash confirmado en base de datos. Acción ejecutada correctamente. | N/A | Aprobado |
| CP12 | Products | P06 | Eliminación en cascada | Permitir desactivar varios productos a la vez | El sistema debe permitir la desactivación simultánea de múltiples productos, reflejando el cambio de estado en todos ellos. Los productos seleccionados deben desaparecer de la lista activa y mostrarse en la pestaña correspondiente. | Los productos seleccionados se mueven a trash y quedan visibles en la papelera, manteniendo los registros en la base de datos | Se sugiere renombrar la ruta massDestroy a bulk-delete para mejorar la claridad semántica y alinearse con convenciones RESTful. | Aprobado |

Tabla 164. Matriz de resultados (Parte III).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|----------|---------------|--|---|--|---|--|---------------------|
| CP13 | Products | P06 | Restauración de producto individual | Verificar que producto “trashed” se restaure individualmente | El sistema debe permitir la restauración individual de un producto previamente enviado a la papelera, mostrando nuevamente el registro en la tabla de productos activos. | El sistema restauró correctamente el producto seleccionado desde la papelera, mostrándolo nuevamente en la tabla de productos activos. Además, desplegó un mensaje de éxito. | N/A | Aprobado |
| CP14 | Products | P06 | Restauración masiva de productos | Validar restauración masiva desde pestaña “Trashed” | El sistema debe permitir la restauración masiva de productos previamente enviados a la papelera, mostrando todos los registros seleccionados en la lista de productos activos. | El sistema restauró correctamente los productos seleccionados desde la pestaña “Trashed”, mostrándolos nuevamente en la tabla de productos activos. La funcionalidad se ejecutó de manera consistente | N/A | Aprobado |
| CP15 | Products | P02, P05 | Guardar con campos obligatorios vacíos | Validar que el formulario no envíe datos incompletos | El sistema debe impedir el envío del formulario cuando existan campos obligatorios vacíos, mostrando un mensaje de validación claro al usuario | No permite ingresar un producto sin nombre ya que ese campo es obligatorio y muestra un mensaje de advertencia que el campo debe ser llenado. | N/A | Aprobado |
| CP16 | Products | P02 | Manejo de valores numéricos negativos | Validar que el sistema rechace valores numéricos negativos en el campo de Precio. | El sistema debe mostrar únicamente las órdenes registradas dentro del rango de fechas ingresado, excluyendo aquellas que no cumplen con el criterio | El sistema permite guardar el precio negativo sin mostrar ningún mensaje de error y finaliza la operación normalmente, redirigiendo a la lista de productos. | El error en la prueba indica que el mensaje de validación nunca apareció, lo que confirma que no existe la restricción en la aplicación. Se recomienda implementar validación en backend y opcionalmente validar también en frontend para mejorar la experiencia de usuario. | Fallido |

Tabla 165. Matriz de resultados (Parte IV).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|--------|---------------|--|---|--|--|---|---------------------|
| CP17 | Orders | O01 | Filtrar órdenes por rango de fechas | Validar que filtros por fechas funcionen correctamente | Resultados filtrados correctamente | Resultados filtrados correctamente | N/A | Aprobado |
| CP18 | Orders | O03 | Visualización de detalle de orden | Verificar que detalle de orden muestre información completa | La página de detalle de la orden se carga correctamente. Se visualiza el nombre del cliente y el vendedor. El monto total de la orden coincide con lo esperado El estado de seguimiento es visible | La página de detalle de la orden se carga correctamente. La información del vendedor, cliente, monto, fecha y estado de seguimiento es visible. | N/A | Aprobado |
| CP19 | Orders | O03 | Actualización de envío y número de seguimiento | Validar modificación el estado de envío de una orden y el número de envío | El sistema edito correctamente la orden existente El total calculado correctamente actualizado. Número y estado de envío actualizado en la orden | Ambos tests pasaron. Los datos de la orden fueron actualizados exitosamente. | Mantener la lógica de reutilización de tracking number para evitar errores por campos obligatorios. | Aprobado |
| CP20 | Orders | O03 | Modificación de detalles de orden | Editar cantidad, precios y descuentos | El sistema edito correctamente la orden existente El total calculado correctamente actualizado. Número y estado de envío actualizado en la orden | Los cambios en cantidad, precio, descuento y cliente se reflejan correctamente en la orden y en el total calculado | N/A | Aprobado |
| CP21 | Orders | O03 | Actualización de estado de pago | Cambiar tipo de pago asociado a una orden | El estado del método de pago se actualiza correctamente en la base de datos y se refleja en el sistema | El método de pago fue actualizado de manera exitosa. El estado fue actualizado de C a D (Cheque a Efectivo) | N/A | Aprobado |

Tabla 166. Matriz de resultados (Parte V).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|--|--|---|---|--|-------------------------|
| CP22 | Orders | O04 | Exportación de órdenes a Excel | Validar generación de archivo Excel con datos filtrados | Excel descargado con encabezados. Los registros deben coincidir con los filtros aplicados. | Excel descargado con encabezados. Los registros deben coincidir con los filtros aplicados. | N/A | Aprobado |
| CP23 | Orders | O03 | Impresión de detalle de orden | Verificar generación correcta de PDF con detalle de orden | El PDF debe mostrar el detalle completo de la orden, con formato correcto, productos listados, totales calculados y sin errores de visualización. | El PDF se generó correctamente, con todos los datos visibles, formato limpio, cálculos precisos y formato amigable. | Dado que el sistema no ofrece opción para enviar factura por correo electrónico u otro canal, contar con una opción para enviarla serviría como respaldo operativo. | Aprobado |
| CP24 | Orders | SR01, SR02 | Generación de reporte de ventas por producto | Generar reporte con productos más vendidos filtrados por fecha | El sistema debe aplicar los filtros correctamente y generar un archivo Excel con las órdenes correspondientes. | Se aplicaron los filtros correctamente, se descargó el archivo Excel y los datos mostrados corresponden. | N/A | Aprobado |
| CP25 | Orders | O03 | Eliminación de orden | Verificar eliminación lógica de orden | La orden del cliente fue eliminada correctamente. Eliminación confirmada en el sistema y base de datos. | Orden fue eliminada correctamente Confirmación en base de datos | N/A | Aprobado |
| CP26 | Inventory | I01 | Registro exitoso de bodega | Registrar nueva bodega con datos completos | La bodega se crea correctamente en la base de datos, el registro contiene: nombre, dirección, estado. | La Bodega/Warehouse se registra correctamente, sin embargo, el state_id no corresponde al guardado lo que sugiere una sobrescritura en el controlador. Además, debido a la paginación limitada registros, las bodegas recién creadas pueden no ser visibles en la primera página. | La bodega se registra correctamente, pero el state_id no se reflejaba en la interfaz por una sobrescritura fija en el controlador (state_id = 1). La vista está limitada a 10 registros por paginación, lo que puede ocultar bodegas nuevas. Se recomienda agregar navegación visible y revisar filtros para asegurar trazabilidad visual. | Ejecutado con hallazgos |

Tabla 167. Matriz de resultados (Parte VI).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|---------------------------------------|--|--|--|--|---------------------|
| CP27 | Inventory | I02, I03 | Filtro por bodega en inventario | Mostrar productos según bodega seleccionada | El sistema debe filtrar correctamente los productos del inventario según la bodega seleccionada, mostrando únicamente aquellos que corresponden a dicha bodega. La operación debe ejecutarse sin errores | El sistema aplicó el filtro de manera satisfactoria, desplegando únicamente los productos asociados a la bodega seleccionada | N/A | Aprobado |
| CP28 | Inventory | I05 | Exportación de inventario a Excel | Generar archivo Excel con reporte visible | El sistema debe generar un archivo Excel con los datos del inventario. La operación debe ejecutarse sin errores | El sistema exportó correctamente el archivo Excel. La funcionalidad se ejecutó de manera satisfactoria, sin incidencias ni inconsistencias | N/A | Aprobado |
| CP29 | Inventory | I06, I07 | Consulta de historial Kardex mensual | Mostrar historial de movimientos filtrado por producto y fecha | El sistema debe mostrar el historial mensual del producto seleccionado, con cálculos de entradas y salidas correctos y totales precisos | El sistema muestra el historial del producto seleccionado en el mes indicado, incluyendo los movimientos y cálculos correctos | Se sugiere agregar la opción de ver el historial solo seleccionando el producto sin necesidad de seleccionar un mes en específico, ya que la búsqueda se vuelve tediosa para el usuario final. | Aprobado |
| CP30 | Inventory | I07 | Verificación del saldo por movimiento | Validar cálculo de saldo en Kardex | El sistema debe calcular el saldo de cada movimiento como: saldo anterior + entradas – salidas, reflejando valores exactos en cada registro y en el total acumulado | El sistema muestra correctamente el saldo por movimiento aplicando la fórmula saldo anterior + entradas – salidas, con totales precisos en el Kardex | N/A | Aprobado |

Tabla 168. Matriz de resultados (Parte VII).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|-----------------------------------|---|---|--|--|---------------------|
| CP31 | Inventory | I09, I10 | Carga masiva de stock desde Excel | Verificar que el módulo de carga de inventario funcione correctamente mediante la carga de un archivo Excel | El archivo se carga correctamente. Mostrar mensaje de confirmación. | Archivo cargado correctamente. Mensaje "Archivo importado correctamente" | N/A | Aprobado |
| CP32 | Customer | C01 | Registrar nuevo cliente válido | Validar que el sistema registre correctamente un nuevo cliente con sus datos personales, estado geográfico, usuario asociado, | El sistema registrara un nuevo cliente al sistema de forma exitosa | Se ejecutó exitosamente con todos los requisitos cumplidos. El sistema registró correctamente un nuevo cliente en la base de datos, y con contraseña hashheada | Se identificó que el tiempo de ejecución fue de 9.46 segundos. Este incremento se atribuye al envío real del email a través de SMTP durante la prueba. | Aprobado |
| CP33 | Customer | C02 | Editar cliente existente | Validar que el sistema permita editar los datos de un cliente existente y que los cambios se reflejen correctamente en la base de datos. En caso de modificar las credenciales, que este envíe el correo de notificación. | Campos phone, address, email, y password actualizados. Correo enviado a nuevo email | Se ejecuto exitosamente. El sistema edito correctamente al cliente existente y se envió el correo con la información requerida. | N/A | Aprobado |
| CP34 | Customer | C02 | Desactivar (eliminar) cliente | Verificar que un cliente puede ser eliminado desde la interfaz de administración, asegurando que ya no aparece en la lista de clientes. | El cliente seleccionado debe ser eliminado/desactivado y ya no debe aparecer en la lista de clientes. | El cliente seleccionado debe ser eliminado/desactivado y ya no debe aparecer en la lista de clientes. | N/A | Aprobado |
| CP35 | Customer | C02 | Búsqueda de cliente por nombre | Validar que el sistema pueda buscar clientes por nombre completo o parcial de manera eficiente, mostrando resultados correctos en menos de 2 segundos. | Cada cliente buscado debe aparecer en los resultados, y el tiempo de respuesta debe ser menor a 2 segundos. | Todos los clientes seleccionados fueron encontrados correctamente, con tiempos de respuesta entre 0.32s y 0.98s, cumpliendo el criterio de <2s | N/A | Aprobado |

Tabla 169. Matriz de resultados (Parte VIII).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|----------|---------------|--|---|---|--|---|-------------------------|
| CP36 | Customer | CO3 | Envío de credenciales por email | Validar que el sistema envíe correos electrónicos con contenido responsivo, seguro, que incluya las credenciales de acceso para nuevos clientes. | El correo debe enviarse exitosamente, contener los datos correctos, mostrarse con diseño responsivo. | El correo fue enviado y recibido en cliente de correo. El diseño se adaptó correctamente. Las credenciales fueron enviadas como texto visible. No se detectaron enlaces inseguros. | El correo envía la contraseña del usuario en texto plano, siendo un riesgo crítico al poder ser interceptadas. Se recomienda reestructurar la lógica para enviar un enlace de restablecimiento de contraseña para mantenerla segura | Aprobado |
| CP37 | Customer | CO4 | Generación automática de contraseña | Validar que el campo password en el formulario de creación de cliente se autocompleta con una contraseña segura que incluya letras, números y símbolos. | El campo password se autocompleta con una contraseña de al menos 6 caracteres, incluyendo letras, números y símbolos. | Contraseña generada: q8sT1vaPro Falló la validación de símbolo (no se incluyó). Contraseña generada: Z%hDp\$4on! Paso la validación (número y símbolo incluido) | El generador actual no garantiza la inclusión de mayúsculas, símbolos o números. Se recomienda ajustar el script para asegurar que la contraseña autogenerada cumpla con los criterios de seguridad | Ejecutado con hallazgos |
| CP38 | Customer | CO5 | Búsqueda de cliente según Estado y combinada | Verificar que la búsqueda de clientes por estado y la búsqueda combinada (Nombre + Estado) funcione correctamente. | El sistema debe mostrar clientes por Estado. Las combinaciones válidas de nombre + Estado deben devolver resultados. Las búsquedas sin datos deben excluirse éticamente sin error | Búsqueda por estado correcta. Búsquedas combinadas correctas | N/A | Aprobado |

Tabla 170. Matriz de resultados (Parte IX).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|--------|---------------|--|---|--|--|--|-------------------------|
| CP39 | Users | U01 | Creación de usuario Administrador o Seller | Registrar usuario con rol válido | El sistema debe permitir la creación de un nuevo usuario con rol válido (Administrador o Seller), asignando correctamente el rol seleccionado en el formulario. La operación debe completarse sin errores | El sistema creó satisfactoriamente el usuario con el rol asignado, validando que la funcionalidad de registro se ejecuta de manera consistente. El nuevo usuario quedó disponible en la lista de usuarios. | | Aprobado |
| CP40 | Users | U01 | Validación de email único | No permitir crear usuario con email duplicado | El sistema debe impedir la creación de un nuevo usuario cuando el correo electrónico ingresado ya existe en la base de datos. La validación debe ejecutarse correctamente, mostrando un mensaje de error y garantizando la unicidad del campo email. | No permite crear o actualizar un usuario con correo repetido. | | Aprobado |
| CP41 | Users | U02 | Generación de contraseña de usuario | Validar que la contraseña de usuario pueda generarse correctamente mediante el botón "Generate password" y que cumpla con los criterios de complejidad. | Campo de contraseña inicialmente vacío. Contraseña generada correctamente al hacer clic en el botón y que cumpla con los criterios mínimos de complejidad | Contraseña generada mediante botón, campo inicialmente vacío. Los criterios de complejidad de contraseña no siempre son aplicados | El generador de contraseña actual no garantiza que siempre se cumplan todos los criterios de complejidad (mayúsculas, números y símbolos). Se recomienda mejorar la función para asegurar que cumpla con los requisitos de seguridad | Ejecutado con hallazgos |

Tabla 171. Matriz de resultados (Parte X).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|--------|---------------|--------------------------------------|---|---|---|------------------------------|---------------------|
| CP42 | Users | U03 | Edición de rol de usuario | Cambiar rol de usuario solo por Admin | El sistema debe permitir que un usuario autenticado como Admin edite el rol de otros usuarios, reflejando el cambio de manera inmediata en la base de datos y en la interfaz. La operación debe completarse sin errores. | El sistema actualizó correctamente el rol del usuario seleccionado, confirmando que la funcionalidad de edición se ejecuta de forma correcta. El cambio quedó reflejado en la tabla de usuarios y no se presentaron incidencias | | Aprobado |
| CP43 | Users | U05 | Validación de opciones del campo Rol | El dropdown solo debe mostrar roles válidos | El sistema debe mostrar únicamente las opciones válidas en el campo Rol, limitando el dropdown a “Administrador” y “Bodeguero”. La validación debe garantizar que no aparezcan roles no autorizados y que la selección se realice sin errores | El sistema desplegó correctamente las opciones esperadas en el dropdown, mostrando únicamente “Administrador” y “Bodeguero”. La funcionalidad se ejecutó de manera correcta, sin incidencias ni roles adicionales visibles | | Aprobado |
| CP44 | Users | U01 | Validación de Campos Vacíos | Verificar que el sistema no permita crear un usuario si los campos obligatorios están vacíos. | El sistema debe validar que ningún usuario pueda ser creado si existen campos obligatorios vacíos, mostrando mensajes de error específicos como “El nombre es obligatorio”. El formulario debe permanecer en la misma página, sin registrar información incompleta. | El sistema mostró mensajes de error por cada campo que se dejó vacío y no permitió crear el usuario con los campos obligatorios vacíos. | | Aprobado |

Tabla 172. Matriz de resultados (Parte XI).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|-----------------------|--|--|--|---|-------------------------|
| CP45 | Seguridad | NF06 | Inyección código SQL | Validar que el sistema rechaza intentos de inyección SQL en el formulario de login, sin ejecutar consultas maliciosas ni exponer errores técnicos. | El sistema debe rechazar entradas maliciosas. No debe ejecutar consultas SQL no autorizadas. Debe mostrar mensajes de error genéricos o validar el formato antes de enviar. No debe reflejar contenido sin codificación. | La aplicación no muestra errores SQL, indicando una configuración adecuada. No se evidencian vulnerabilidades de SQL Injection por diferencias en las respuestas. Todos los payloads fueron aceptados por el backend con código HTTP 302 Found, sin evidencia de ejecución maliciosa. No se detectó acceso no autorizado ni fuga de datos. | El código 302 sugiere redirección, probablemente al mismo formulario de login para fallos de autenticación. | Aprobado |
| CP46 | Seguridad | NF10 | Acceder a vista Admin | Bloquear acceso a endpoints Admin para usuarios básicos | El sistema debe mostrar una pantalla informando que el usuario no tiene permisos para visualizar el contenido solicitado, bloqueando completamente el acceso a la vista de administrador | El sistema mostró una pantalla informando que el usuario no cuenta con los permisos necesarios para visualizar el contenido solicitado | Se recomienda implementar un redireccionamiento automático hacia la pantalla de inicio correspondiente al tipo de usuario que inicia sesión, para mejorar la experiencia, no se registra ningún log de seguridad | Ejecutado con hallazgos |
| CP47 | Seguridad | NF04 | Headers HTTP | Validar la presencia, ausencia y calidad de los headers HTTP en las respuestas del sistema. | Headers definidos desde el código deben estar presentes y configurados correctamente. Headers inseguros o ausentes deben ser identificados para corrección. | Header set-cookie con HttpOnly, Secure, SameSite=Lax, Content-Type, Cache-Control, Content-Security-Policy y Referrer son correctos. Faltan algunos headers de seguridad importantes. El header X-Powered-By está presente. | X-Powered-By expone la tecnología del backend innecesariamente. Sin embargo, esto no afecta el funcionamiento del sistema. Se recomienda su corrección desde el código fuente por razones de seguridad y buenas prácticas. Adicional, agregar los headers X-Content-Type-Options, X-Frame-Options, Referrer-Policy y Permissions-Policy para reforzar la seguridad del sistema. | Fallido |

Tabla 173. Matriz de resultados (Parte XII).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|--------------|---------------|-------------------------------|--|---|---|---|-------------------------|
| CP48 | Rendimiento | NF05 | 100 usuarios concurrentes | Simular 100 usuarios haciendo búsquedas simultáneas | El sistema debe responder de manera estable, sin caídas ni errores en las solicitudes concurrentes, manteniendo la disponibilidad y tiempos de respuesta | Se tuvo un 1.05% de Error en cuanto a caídas de las 100 requests concurrentes | Se desvía por un margen menor a 2% en cuanto al objetivo esperado de la prueba, de ser necesario esta concurrencia o una un poco superior se debería ver la manera de ampliar los recursos del servidor | Ejecutado con hallazgos |
| CP49 | Rendimiento | NF05 | Saturación del sistema | Simular saturación con 500+ usuarios | El sistema debe manejar la saturación recuperándose automáticamente, manteniendo la disponibilidad y evitando bloqueos críticos | El servidor rechazó las solicitudes cuando se superaron las 100 peticiones provenientes de una misma dirección IP | El servidor bloquea cuando son más demasiadas solicitudes de una misma dirección IP, esto es excelente en cuanto a seguridad, pero complica pruebas de saturación al sistema | Ejecutado con hallazgos |
| CP50 | Concurrencia | P09 | Edición del mismo producto | Validar edición simultánea sin conflictos | El sistema debería permitir que dos usuarios editen el mismo producto al mismo tiempo sin que se pierdan cambios, manteniendo la información correcta y completa | Al guardar las ediciones, el sistema reemplaza los valores y solo conserva la última modificación, lo que provoca que se pierdan los cambios anteriores | Se deberían actualizar únicamente los campos que sean diferentes al contenido actual en la base de datos | Fallido |
| CP51 | Seguridad | NF03 | Gestión de sesión – time-out | Validar que la sesión se cierre tras detectar inactividad | Sesión expira automáticamente y se redirige al login | Después de 40 min de inactividad, lo cual significa que no se está aplicando el timeout por inactividad | Esto representa un riesgo medio. OWASP lo clasifica como una debilidad en el manejo de sesiones. | Fallido |
| CP52 | Seguridad | NF10 | Restricción de acceso por rol | Verificar que usuarios no autorizados no puedan acceder a funciones restringidas | El sistema debe denegar el acceso y mostrar un mensaje indicando que el usuario no cuenta con los permisos necesarios. No debe exponerse información sensible ni permitir acceso parcial a la vista restringida | El sistema denegó el acceso y mostró un mensaje informando que el usuario no cuenta con los permisos requeridos. No se expuso información sensible ni se permitió acceso al contenido restringido | Se debería redireccionar luego de informar que no se tiene acceso. | Aprobado |

Tabla 174. Matriz de resultados (Parte XII).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|-----------------------------------|--|--|--|---|---------------------|
| CP53 | UI/UX | NF02 | Compatibilidad con navegadores | Validar que el sistema funciona correctamente en los diferentes navegadores en sus versiones recientes | Login y dashboard cargado. Formulario abierto y cargado sin errores. Recursos (CSS, JS, imágenes) cargados. | Prueba ejecutada correctamente en Chrome, Edge y Firefox. Datos enviado y CSS, JS e imágenes cargadas correctamente. | N/A | Aprobado |
| CP54 | Seguridad | NF07 | Revisión de código seguro | Confirmar que se aplican prácticas seguras de programación, minimizando vulnerabilidades comunes | El reporte final no debe mostrar vulnerabilidades críticas y el código debe cumplir con los estándares de seguridad definidos, garantizando estabilidad y mantenibilidad | El reporte final muestra 412 issues críticos | Es urgente la refactorización de código el análisis del código revela un sistema cuya estructura interna presenta serios problemas que afectan directamente su calidad, estabilidad y capacidad de evolucionar de forma segura. | Fallido |
| CP55 | Seguridad | NF08 | Validación de certificado SSL/TLS | Verificar que el certificado SSL/TLS del sitio web sea válido, esté correctamente configurado y utilice protocolos seguros | El certificado es válido, está vigente, la conexión utiliza protocolos y cifrados seguros sin advertencias | El certificado es válido, está vigente, la conexión utiliza protocolos y cifrados seguros sin advertencias | N/A | Aprobado |
| CP56 | Seguridad | NF09 | Transporte seguro de credenciales | Verificar que el envío de usuario y contraseña se haga bajo canal cifrado (HTTPS/TLS) | El tráfico debe estar cifrado mediante, garantizando que las credenciales no se envíen en texto plano y que no puedan ser interceptadas por terceros | Las credenciales se enviaron en texto plano, lo que representa una vulnerabilidad crítica en el sistema | Modificar la forma de envío de las credenciales ya que si alguien logra interceptarlas quedan grandes vulnerabilidades en el sistema | Fallido |

Tabla 175. Matriz de resultados (Parte XIV).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|--|--|---|---|--|---------------------|
| CP57 | Seguridad | NF01 | Protección contra fuerza bruta | Verificar que el módulo de login implemente mecanismos efectivos contra ataques de fuerza bruta y presencia de CAPTCHA o mecanismos similares. | El sistema debe activar mecanismos de defensa ante múltiples intentos fallidos: bloqueo temporal, CAPTCHA, rate limit, etc. | Pruebas manuales se mostraron bloqueo temporal al superar el umbral de intentos con el mensaje: "Too many login attempts. Please try again in X seconds." pero el tiempo de espera no escala progresivamente. Pruebas automáticas con Fuzzer no activaron defensa visible. Todas las respuestas HTTP fueron 302 Found. No se detectó CAPTCHA en ninguno de los intentos. Mensaje de error mostrado y capturado en el body HTTP: These credentials do not match our records. | La defensa contra fuerza bruta está parcialmente implementada. Utiliza un bloqueo temporal fijo (el tiempo varía sin un patrón progresivo acumulativo). Funciona en sesiones manuales, pero no se activa ante ataques automatizados sin estado. Dado que el sistema solo bloquea por sesión/cookie, podría ser evadido con ataques sin estado (no mantienen cookies, tokens, variables de sesión). Se sugiere fortalecer la protección con mecanismos como backoff progresivo, bloqueo por IP/usuario y CAPTCHA o 2FA tras múltiples intentos. | Fallido |
| CP58 | Seguridad | NF09 | Detección de credenciales predeterminadas /débiles | Intentar crear usuario con credenciales comunes o vacías. | Acceso denegado, mensaje genérico | El sistema permite la creación de usuarios con contraseñas que no cumplen con estándares mínimos de seguridad, como longitud, complejidad o uso de caracteres especiales. Esto representa una vulnerabilidad que puede facilitar ataques de fuerza bruta o acceso no autorizado. | El servidor respondió con redirección (302 Found), lo cual indica que las contraseñas fueron aceptadas sin validación. | Fallido |

Tabla 176. Matriz de resultados (Parte XV).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|-----------------------------|--|--|---|---|-------------------------|
| CP59 | Seguridad | NF10 | Enumeración de cuentas | Verificar si el sistema revela la existencia de cuentas de usuario mediante mensajes de error diferenciados durante el proceso de autenticación. | Mensaje genérico ante credenciales inválidas que no indique existencia del usuario. Códigos HTTP consistentes para fallos de autenticación (no devolver 404 específico para usuario no existente ni otros códigos diferenciadores). No diferencias observables en headers/cookies que permitan inferir existencia. | Mensaje de error mostrado y capturado en el body HTTP: These credentials do not match our records. Resultado del Fuzzer: todas las peticiones devolvieron HTTP 302 Found y el tamaño del cuerpo de respuesta fue consistente (386 bytes) para todos los intentos. Prueba con campo password vacío retornó HTTP 200 OK y mostró validación de campo: The password field is required. No se detectaron diferencias en cabeceras relevantes ni emisión de cookie de sesión en respuestas de fallo que permitan identificar cuentas. | Asegurar mecanismos contra fuerza bruta (bloqueo por IP/usuario, CAPTCHA o 2FA). | Aprobado |
| CP60 | Seguridad | NF01 | Validación de "Remember Me" | Evaluar si la opción "recordarme" conserva sesiones de forma segura | La sesión debe mantenerse activa únicamente si el usuario seleccionó la opción "Remember Me" | Si mantiene la sesión activa durante horas, aunque se cierre el navegador, pero es independiente sobre si se presiona el botón "Remember me" | Independientemente si se presiona o no el botón la sesión queda guardada por un lapso de unas horas | Ejecutado con hallazgos |
| CP61 | Seguridad | NF01 | Gestión segura de sesiones | Verificar atributos de cookies, expiración y control de sesiones concurrentes | El sistema debe generar cookies seguras, expirar la sesión correctamente tras el tiempo definido y controlar la concurrencia | El sistema debe generar cookies seguras, expirar la sesión correctamente tras el tiempo definido y controlar la concurrencia | N/A | Aprobado |
| CP62 | Seguridad | NF12 | Bypass de autenticación | Intentar saltarse el login o manipular parámetros. | No permite saltarse el login. | El servidor respondió con 302 Found, redireccionando al login, lo cual indica que bloqueó el acceso por falta de sesión. | | Aprobado |

Tabla 177. Matriz de resultados (Parte XVI).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|-----------|---------------|--|---|---|--|------------------------------|---------------------|
| CP63 | Seguridad | NF10 | Respuestas ante errores y mensajes | Evaluar si los mensajes de error revelan información sensible. | El sistema debía mostrar mensajes de error simples, sin incluir información sensible ni detalles internos en la respuesta o en el código HTML | Durante la ejecución de la prueba, los mensajes de error se presentaron correctamente y no revelaron información sensible en la respuesta ni en el HTML | N/A | Aprobado |
| CP64 | UI/UX | NF11 | Pruebas de UI/UX y resiliencia | Verificar funcionamiento de CAPTCHA, enlaces, validaciones, y responsive. | La interfaz debía mostrarse correctamente, con mensajes claros, enlaces funcionando sin errores y un diseño adaptable a distintos dispositivos | La interfaz se mostró de forma correcta y los enlaces funcionaron adecuadamente | N/A | Aprobado |
| CP65 | Purchase | P01 | Vista inicial de categorías | Validar que al iniciar sesión el usuario sea redirigido a la pantalla de categorías | El sistema debe redirigir al usuario autenticado directamente a la pantalla de categorías, mostrando de manera clara y organizada todas las categorías disponibles | El sistema redirigió correctamente al usuario a la pantalla de categorías, mostrando todas las categorías disponibles de forma completa y sin errores de visualización. El comportamiento observado coincide con lo esperado | N/A | Aprobado |
| CP66 | Purchase | P02 | Visualización de productos por categoría | Verificar que al seleccionar una categoría se muestren solo los productos asociados | El sistema debe mostrar los productos asociados a la categoría seleccionada, sin errores de visualización | El sistema mostró correctamente los productos correspondientes a la categoría seleccionada, cumpliendo con la funcionalidad esperada y sin presentar inconsistencias en la visualización | N/A | Aprobado |
| CP67 | Purchase | P03 | Gestión de productos en carrito | Validar agregar y remover productos desde la categoría seleccionada | El sistema debe permitir agregar y remover productos del carrito de manera inmediata, reflejando los cambios en la vista del carrito y manteniendo la consistencia de los datos | El sistema permitió agregar y remover productos del carrito de manera correcta, reflejando los cambios en tiempo real y cumpliendo con la funcionalidad esperada. No se presentaron errores ni inconsistencias | N/A | Aprobado |

Tabla 178. Matriz de resultados (Parte XVII).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|----------|---------------|---|--|--|--|---|---------------------|
| CP68 | Purchase | P04 | Cálculo de totales | Validar que se muestren los cálculos correctos (subtotal, descuentos, shipping, total final) | El sistema debía mostrar correctamente los cálculos de subtotal, descuentos y total final, reflejando valores precisos en cada etapa | Durante la prueba, los cálculos se mostraron de manera correcta y los valores de subtotal, descuentos y total final coincidieron con lo esperado | N/A | Aprobado |
| CP69 | Purchase | P05 | Actualización de estado de pago (Admin) | Validar que desde la vista Admin se pueda cambiar estado de pago y se refleje en facturación | El sistema debía permitir que el administrador cambiara el estado de pago de una orden y que este cambio se reflejara correctamente en la base de datos y en la facturación | Durante la prueba, el estado de pago se actualizó de forma correcta y el cambio quedó reflejado tanto en la base de datos como en la facturación | N/A | Aprobado |
| CP70 | Category | CA01 | Validación de campos obligatorios | Verificar que el formulario de crear categorías valide correctamente los campos, asegurando que se muestren errores cuando los datos sean inválidos. | Si los campos obligatorios están vacíos, se muestran mensajes de error y el formulario permanece abierto. Si los campos obligatorios se completan correctamente, el registro se guarda y el sistema redirige a la lista de categorías. | Coincide con los resultados esperados: - Escenarios con campos vacíos muestran errores inmediatamente. - El campo "Name" es obligatorio, ya que su ausencia impide el guardado del formulario, independientemente de la presencia o ausencia de los campos "Description" e "Image", los cuales son opcionales. | Se sugiere mostrar mensajes y señales visuales claras para facilitar la validación y mejorar la trazabilidad del formulario, contribuyendo a una mejor experiencia del usuario. | Aprobado |
| CP71 | Category | CA02 | Validación de carga de imagen | Verificar que el sistema acepte solo formatos válidos de imagen y rechace los no permitidos | Categoría guardada correctamente junto con su imagen. Formatos válidos: mensaje de éxito. Formatos no válidos: mensaje de error. | Categoría guardada correctamente junto con su imagen. Formatos válidos: mensaje de éxito. Formatos no válidos: mensaje de error. | Formato del tipo webp no aceptada | Aprobado |

Tabla 179. Matriz de resultados (Parte XVIII).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|----------|---------------|---------------------------------------|--|--|--|------------------------------|---------------------|
| CP72 | Category | CA03 | Registro exitoso de categoría | Verificar que al crear una categoría con datos válidos se registre correctamente en la base de datos y aparezca en la lista de categorías. | La categoría se guarda en la base de datos. La categoría aparece en la lista de categorías. El campo name, description e imagen se muestran y guardan correctamente. | La categoría fue registrada correctamente con nombre, descripción e imagen visibles en la lista. | N/A | Aprobado |
| CP73 | Category | CA04 | Retorno a lista desde Create Category | Verificar que al presionar el botón “Back to List” desde el formulario de creación de categoría, el sistema regrese correctamente a la vista de lista de categorías. | El sistema regresa a la lista de categorías. Los datos existentes en la lista se muestran correctamente. No se presentan mensajes de error. | Funciona correctamente, la lista se muestra correctamente tras presionar Back to List. | N/A | Aprobado |
| CP74 | Category | CA05 | Eliminar categoría | Validar que se pueda eliminar una categoría existente de la lista correctamente y que los cambios se reflejen en el sistema. | La categoría seleccionada se elimina correctamente. La lista de categorías se actualiza y no muestra la categoría eliminada. | La categoría fue eliminada exitosamente. BD actualizado | N/A | Aprobado |
| CP75 | Category | CA05 | Editar una categoría | Validar que se pueda editar una categoría existente y que los cambios se reflejen correctamente en la BD y en la vista de lista. | La categoría se actualiza correctamente y se muestran los cambios en la lista, incluyendo la imagen. | La categoría fue editada correctamente. Los cambios se reflejan correctamente. | N/A | Aprobado |

Tabla 180. Matriz de resultados (Parte XIX).

| ID | Módulo | Req. asociado | Título del caso | Descripción | Resultado esperado | Resultado obtenido | Observaciones/ Recomendación | Estado de la prueba |
|------|--------|---------------|--|--|--|---|---|-------------------------|
| CP76 | Orders | O03 | Agregar un producto a la orden | Validar que el sistema permita agregar dinámicamente un producto existente a una orden real, recalculando totales y aplicando descuentos si corresponde. | Todos los pasos ejecutados correctamente. Producto agregado, totales recalculados. El sistema no permite agregar producto, si la orden está vacía. | Todos los pasos ejecutados correctamente. Producto agregado, totales recalculados. Si la orden se encuentra vacía, el sistema no permite agregar nuevos producto generándose un error no crítico | El sistema debe permitir regenerar dinámicamente order_details al agregar nuevos productos, evitando bloqueos cuando la orden está vacía. | Ejecutado con hallazgos |
| CP77 | Orders | O03 | Remover producto /productos de una orden | Validar que el sistema permite eliminar un producto específico de una orden sin afectar la integridad de los datos ni generar errores silenciosos. | Productos seleccionados son eliminados. El total se actualiza correctamente. No se generan errores y la orden permanece activa y editable. | Productos fueron seleccionados y eliminados correctamente. Error SQL al eliminar un producto duplicado (ej. si hay 2 productos con el mismo nombre, no permite eliminar el primero ingresado). Al eliminar el ultimo registro duplicado, se eliminan los demás. Si la orden queda vacía, el sistema no permite agregar nuevos productos | Se recomienda revisar la lógica de eliminación para que cada instancia de producto sea tratada de forma independiente y no agrupada. Además, el sistema debe permitir regenerar dinámicamente order_details al agregar nuevos productos, evitando bloqueos cuando la orden queda vacía. No es un “fallo” técnico porque el script se ejecuta sin romperse, pero tampoco es un “paso” funcional porque el comportamiento observado no cumple con lo esperado | Ejecutado con hallazgos |

Tabla 181. Matriz de resultados (Parte XX).

12.5. Anexo 5: Presentación de acta de aceptación de matriz de pruebas y resultado

El presente documento formaliza la aceptación de la matriz de pruebas y sus resultados como parte del proceso de validación del sistema de comercio electrónico en producción. Resume los entregables evaluados, los hallazgos obtenidos durante la ejecución de pruebas funcionales, y autoriza la continuidad del proyecto en el marco de la tesis.

| |
|--|
| <p style="text-align: center;">ACTA DE ACEPTACIÓN DE MATRIZ DE PRUEBAS Y RESULTADOS</p> <p>Proyecto: Diseño e implementación de un plan integral de pruebas manuales y automatizadas para asegurar la calidad de un sistema de comercio electrónico en producción.</p> <p>Fecha: 01 de diciembre del 2025</p> <p>Usuario solicitante: Dennis Orellana</p> <p>Responsables de pruebas:</p> <ul style="list-style-type: none">• Jennifer Cruz• Mauricio Hernández• Nathaly Rodríguez <p>1. Objetivo del documento</p> <p>El presente documento tiene como propósito formalizar la aceptación, por parte del usuario solicitante, de la matriz de pruebas ejecutada durante el proceso de validación del sistema ecommerce, así como de los resultados obtenidos de cada uno de los casos de prueba.</p> <p>Esta acta certifica que el usuario ha revisado la información correspondiente, valida que los resultados reflejan el comportamiento real del sistema y autoriza la continuidad del proceso establecido para el proyecto de tesina.</p> <p>2. Alcance</p> <p>Esta acta abarca los entregables generados durante la fase de pruebas del sistema, los cuales son:</p> <ul style="list-style-type: none">• Matriz de pruebas funcionales del sistema.• Resultados de la ejecución de pruebas.• Hoja de control de seguimiento.• Evidencias documentadas en capturas de pantalla.• Incidencias registradas y sus respectivas observaciones. <p>3. Resumen general de resultados</p> <p>Durante la ejecución del plan de pruebas, se obtuvieron los siguientes resultados:</p> <ul style="list-style-type: none">• Total de casos ejecutados: 77• Casos aprobados: 58• Casos fallidos: 9• Casos ejecutados con hallazgos: 10 |
|--|

Ilustración 36. Acta de Aceptación de Pruebas y Resultados - Parte I.

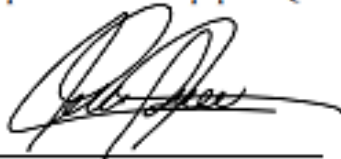
4. Declaración de aceptación

Yo, Dennis Ernesto Orellana Ramírez, en calidad de usuario solicitante, declaro que:

1. He revisado la matriz de pruebas y los resultados obtenidos.
2. Confirmando que la información presentada refleja de manera fiel la ejecución del proceso de pruebas del sistema.
3. Acepto los entregables correspondientes a la fase de pruebas funcionales documentada en este proyecto.
4. Autorizo el cierre de esta fase o la continuación hacia las siguientes actividades definidas para el proyecto.



Jennifer Cruz
Representante del equipo de QA



Dennis Orellana
Usuario aceptante

Ilustración 37. Acta de Aceptación de Pruebas y Resultados - Parte II.

13. Glosario

El presente glosario reúne los principales términos técnicos y metodológicos utilizados en el dicho capítulo. Su finalidad es facilitar la comprensión del contenido, asegurando que los conceptos empleados se interpreten de manera uniforme.

| Término | Definición |
|--------------------------------|---|
| Caso de prueba | Conjunto de pasos diseñados para verificar el correcto funcionamiento de una funcionalidad específica. |
| Defecto | Inconsistencia o error detectado durante las pruebas que afecta el comportamiento esperado del sistema. |
| Evidencia | Captura, registro o documento que respalda el resultado obtenido en la ejecución de un caso de prueba. |
| Cobertura de pruebas | Medida que indica el grado en que los casos de prueba abarcan los requisitos o funcionalidades del sistema. |
| Pruebas funcionales | Validaciones que confirman que cada módulo cumple con los requerimientos definidos. |
| Pruebas de integración | Evaluaciones que verifican la correcta comunicación y consistencia de datos entre distintos módulos. |
| Pruebas de rendimiento | Validaciones que miden tiempos de respuesta, estabilidad y capacidad del sistema bajo carga. |
| Pruebas de seguridad | Actividades destinadas a comprobar la protección de datos, autenticación y resistencia ante ataques. |
| Pruebas de concurrencia | Validaciones que aseguran el correcto funcionamiento con múltiples usuarios simultáneos. |
| Ambiente productivo | Entorno real donde el sistema se encuentra desplegado y en operación para los usuarios finales. |
| Ambiente de pruebas | Entorno controlado donde se ejecutan validaciones antes de liberar cambios al ambiente productivo. |
| Ciclo de pruebas | Conjunto de actividades planificadas que incluyen diseño, ejecución, registro y análisis de pruebas. |
| Plan de pruebas | Documento que define el alcance, objetivos, recursos y cronograma de las pruebas. |
| Matriz de trazabilidad | Herramienta que relaciona requisitos con casos de prueba para asegurar cobertura completa. |

| Término | Definición |
|-----------------------------------|---|
| Caso de prueba (Test Case) | Conjunto de condiciones, entradas y pasos definidos para verificar una funcionalidad específica del sistema. |
| Sprint | Periodo de tiempo determinado en el que se planifican y ejecutan un conjunto de tareas dentro de la metodología ágil Scrum. |
| Prueba funcional | Evaluación de las funciones del sistema para garantizar que cumplan con los requerimientos especificados. |
| Trazabilidad | Capacidad de vincular requerimientos, casos de prueba, resultados y evidencias de forma estructurada y verificable |
| Evidencia de prueba | Registro documentado de los resultados obtenidos durante la ejecución de un caso de prueba, incluyendo capturas o reportes. |
| Automatización de pruebas | Uso de herramientas y scripts para ejecutar pruebas de manera automática, reduciendo el tiempo y los errores humanos. |
| Bug / Defecto | Cualquier comportamiento del sistema que no cumple con los requisitos o expectativas definidos. |
| Matriz de trazabilidad | Herramienta que permite relacionar cada caso de prueba con los requisitos del sistema, asegurando que todos estén cubiertos. |
| Plan de pruebas | Documento que define el alcance, objetivos, recursos, cronograma y riesgos asociados al proceso de testing. |
| QA (Quality Assurance) | Actividades planificadas y sistemáticas para garantizar que el software cumple con los estándares de calidad requeridos. |
| Riesgo | Evento potencial que puede afectar negativamente la ejecución del plan de pruebas, la calidad del sistema o la trazabilidad de los resultados |