



**Certified Tech
Developer**
The Ultimate Degree

DigitalHouse >
Coding School

Reporte Final de Testing

Digitalbooking

Integrantes:

Ayelén Andrade

Brenda María Belén Acosta

Lara Amaya

Diego Fernando

Jesus Eduardo Campo Alvarez

Andrea Orozco Ramírez



Contenido

Introducción	3
Resumen de las actividades de prueba	3
Alcance	3
Dentro del Alcance	3
Fuera de Alcance	3
Tipos de Pruebas Ejecutadas	3
Enfoque de la Prueba	4
Exit Criteria	4
Resumen de Resultados	5
Diseño de Pruebas	5
Ejecución de Pruebas	6
Ejecución Manual	6
Ejecución Automática	6
Reporte de Defectos	6
Todos los defectos	6
Defectos por prioridad	7
Defectos por Severidad	7
Defecto por Estado	7
Defectos Creados vs Resueltos	8
Defectos Abiertos	9
Lecciones Aprendidas / Conclusión	9



Introducción

Este documento es el Informe Final de Pruebas del sistema Rentcards. El propósito de este documento es proveer evidencia de que el Exit Criteria para el proceso de Testing se cumplió y por lo tanto, se concluye la fase de pruebas y puede cerrarse. Se demuestra que los Issues de GitLab relacionados con testing fueron implementados desde los Sprint 1 a 4. Este documento va a ser utilizado como entrada para la revisión general de las actividades de prueba y para tomar la decisión si el sistema cumple con las expectativas.

Resumen de las actividades de prueba

El sistema que desarrollamos consiste en una aplicación que ofrece el servicio de alquiler de vehículos, ahora tendrás la oportunidad de realizar una reserva a través de esta aplicación Rentcards que permite alquilar vehículos de cuatro diferentes categorías las cuales son: deportivo, lujoso, económico y eléctrico. La reserva se puede llevar a cabo seleccionando fechas en dónde el auto se encuentre disponible, y se necesita realizar con un usuario autenticado con un previo registro en la página. También, existe un usuario de tipo administrador que podrá agregar autos para alquilar.

URL: <https://dev.digibook.link/>

<i>Sprint</i>	<i>Nombre de Issues de Testing</i>	<i>Estado</i>
Sprint 1	Planificación y ejecución de los tests	Completa
	Planear casos de test basados en las historias de usuarios que existen en el sprint, y luego ejecutarlos.	Completa
	Realizar también una prueba exploratoria sobre lo que se desarrolló durante el Sprint.	Completa
	Testear las APIs	Completa
	Validar el funcionamiento de la API Rest creada por el Back End durante el Sprint. Utilicen Postman para las pruebas y cree los scripts de prueba necesarios.	Completa
Sprint 2	Implementar tests automatizados	Completa
	Utilizando JEST, crear los unit tests automatizados necesarios para cubrir lo que fue desarrollado durante los sprints 1 y 2.	Completa



	Crear test unitarios y definir el porcentaje de cobertura.	Completa
	Desarrollar los script de pruebas automatizados utilizando Postman para las APIs desarrolladas en este sprint.	Completa
	Pruebas manuales de usuario, suite de regresión y suite de smoke.	Completa
	Prueba manual exploratoria.	Completa
Sprint 3	Crear y ejecutar las pruebas unitarias (unit testing) de las páginas de reserva y confirmación de reserva a través de JEST framework.	Completa
	Añadir las pruebas necesarias para la API de productos, reserva y usuarios a la colección creada previamente en Postman (scripts de pruebas).	Completa
	Pruebas manuales de usuario, suite de regresión y suite de smoke.	Completa
	Prueba manual exploratoria.	Completa
Sprint 4	Implementar testing automatizado	Completa
	Implementar testing manual	Completa
	Realizar y registrar el testing exploratorio sobre las funcionalidades de la página de mis reservas y sobre la página formulario de producto.	Completa
	Agregar casos de prueba Manuales para ambas páginas y clasificar los mismos en Smoke y Regression Test	Completa Completa
	Confeccionar el reporte final de testing	Completa
	Cantidad de casos de prueba creados por funcionalidad/historia de usuario tanto manuales como automáticos.	Completa
	Cantidad de casos de prueba ejecutados, pasados y fallados.	Completa
	Cantidad de defectos por estado y prioridad/severidad	Completa



Alcance

Dentro del Alcance

Funcionalidades del sistema dentro del alcance	Manual	Automatizado
Usuario logueado mostrar el nombre de usuario, avatar con las iniciales del nombre del usuario y un link de cierre de sesión	X	X (JEST)
Login de usuario	X	
Crear cuenta usuario	X	X (POSTMAN, JUNIT)
Usuario al ingresar al dar click en “ver detalles” pueda visualizar los detalles del producto	X	
Bloque buscador página principal	X	
Filtrar por categoría página principal	X	X (POSTMAN)
Filtrar por ciudad página principal	X	X (POSTMAN)
Filtrar por fechas página principal	X	X (POSTMAN)
Calendario página principal seleccionar rango de fechas	X	X (POSTMAN)
Funcionalidad de botón “iniciar sesión”	X	
Funcionalidad de botón “crear cuenta”	X	
Funcionalidad del logo al dar clic regrese a pagina principal	X	
Producto con flecha de volver a página anterior	X	



Funcionalidad al botón de reserva en página de detalle de producto	X	X (POSTMAN, JUNIT)
Bloque de imágenes con botón de “Ver Más”, el cual, permitirá acceder a un componente de galería de imágenes	X	
Galería con auto deslizamiento cada 3 segundos	X	
Producto calendario con fechas disponibles	X	X (POSTMAN)
Página de reserva Icono que permita regresar	X	
Formulario de reserva	X	
Horario de llegada en reserva	X	
Calendario Home Seleccionar múltiples fechas	X	
Calendario en reserva permite mover en distintos meses	X	
Selección de producto	X	
Reserva de producto		X (POSTMAN, JUNIT)
Crear un producto como administrador		X (POSTMAN)
Agregar, listar, eliminar y actualizar categoría		X (POSTMAN, JUNIT)
Encontrar categoría por ID de producto		X (POSTMAN)
Agregar, listar ,eliminar y actualizar ciudades		X (POSTMAN, JUNIT)
Encontrar ciudad por ID de producto		X (POSTMAN)
Agregar, listar y encontrar producto por ID, eliminar y		X (POSTMAN)



actualizar producto		
Encontrar productos por ciudad y por categoría	X	X (POSTMAN)
Buscar producto por ciudad y fecha	X	X (POSTMAN)
Agregar, listar, eliminar y actualizar características	X	X (POSTMAN, JUNIT)
Listar imágenes por ID y agregar imagen de producto	X	X (POSTMAN, JUNIT)
Actualizar puntuaciones		X (POSTMAN, JUNIT)
Agregar usuarios	X	X (POSTMAN)
Login como administrador con JWT	X	X (POSTMAN)
Login como usuario cliente con JWT		X (POSTMAN)
Encontrar reserva por id de producto y encontrar reserva por ID de usuario	X	X (POSTMAN)

Fuera de Alcance

<i>Funcionalidades del sistema fuera del alcance</i>
Calendario página principal
Calendario producto
Calendario reservas
Iconos
Librería de imágenes



Tipos de Pruebas Ejecutadas

	<i>Sprint 1</i>	<i>Sprint 2</i>	<i>Sprint 3</i>	<i>Sprint 4</i>
<i>Prueba Estática</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>
<i>Prueba Exploratoria</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>
<i>Prueba de Sistema</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>SI (Cypress)</i>
<i>Prueba de Humo</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>
<i>Prueba de Regresión</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>
<i>Prueba de Componente / Unidad</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>
<i>Prueba de Integración (Postman)</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>	<i>SI</i>

Enfoque de la Prueba

Durante todos los sprints se realizó una planificación de los tests y su posterior ejecución para validar los resultados. Se crearon test funcionales tanto positivos como negativos, basados en las historias de usuario, incluyendo las pruebas correspondientes a las nuevas funcionalidades creadas, clasificándolas en de pruebas de suite de smoke y suite de regresión. Adicionalmente, antes de finalizar los sprint se ejecutaron pruebas exploratorias.

En el transcurso del proyecto se crearon y ejecutaron pruebas unitarias a diferentes componentes de la aplicación. Para los componentes del desarrollo del frontend, se realizaron tests unitarios con la librería 'react-testing-library' y en el backend se utilizó JUnit para testear la funcionalidad de los métodos creados. También se utilizó POSTMAN para



hacer pruebas de las APIs, por medio del cual obtuvimos diferentes tipos de respuesta que posteriormente fueron validados.

Luego de ejecutar los casos de prueba en cada sprint, incluyendo las pruebas dentro de las suites de regresión y de smoke, y al final de cada prueba exploratoria se reportaron los defectos encontrados en la planilla de defectos. A estos se les asignó un estado y su priorización, ya que cuando el defecto se reporta eficientemente las probabilidades de que sea solucionado rápidamente es mayor.

Link Planilla de Casos de Prueba y Link de plantilla de defectos :

https://docs.google.com/spreadsheets/d/1HarYtPpiVODAo_eFG1pZQwWhKqU2b4ygvEUL7m-0gyY/edit#gid=1685899497

Exit Criteria

Se definió los siguientes criterios de aceptación para finalizar las pruebas:

- No se debe tener defectos en estado abierto de severidad crítica y/o bloqueante
- No se debe tener defectos mayor en estado abierto
- La ejecución de los test en postman deben tener un 100% *de past rate*.

Resumen de Resultados

Diseño de Pruebas

Funcionalidades del sistema dentro del alcance	<i>Test Manuales</i>	<i>Test Automáticos</i>	<i>Test de Integración (Postman)</i>	<i>Test Total</i>
Usuario logueado mostrar el nombre de usuario, avatar con las iniciales del nombre del usuario y un link de cierre de sesión	1	1(Jest)	0	2
Login de usuario	2	0	0	2



Crear cuenta usuario	2	0	1	3
Usuario al ingresar al dar click en “ver más” pueda visualizar los detalles del producto	23	0	0	23
Bloque buscador página principal	1	0	0	1
Filtrar por categoría página principal	1	0	1	1
Filtrar por ciudad página principal	1	0	2	3
Filtrar por fechas página principal	1	0	1	2
Calendario página principal seleccionar rango de fechas	1	0	0	1
Funcionalidad de botón “iniciar sesión”	1	0	0	1
Funcionalidad de botón “crear cuenta”	1	0	1	2
Funcionalidad del logo al dar clic regrese a pagina principal	1	0	0	1
Producto con flecha de volver a página anterior	1	0	0	1
Funcionalidad al botón de reserva en página de detalle de producto	1	0	1	2
Bloque de imágenes con botón de “Ver Más”, el cual, permitirá acceder a un componente de galería de imágenes	1	0	0	1
Galería con auto	1	0	0	1



deslizamiento cada 3 segundos				
Producto calendario con fechas disponibles	1	0	1	2
Página de reserva Icono que permita regresar	1	0	0	1
Formulario de reserva	2	0	0	1
Horario de llegada en reserva	1	0	0	1
Calendario Home Seleccionar múltiples fechas	1	0	0	1
Calendario en reserva permite mover en distintos meses	1	0	0	1
Selección de producto	1	0	0	1
Reserva de producto	0	0	1	1
Crear un producto como administrador	0	0	1	1
Agregar, listar y encontrar categoría por ID de producto, eliminar y actualizar categoría	0	0	1	1
Agregar, listar y encontrar ciudad por ID de producto, eliminar y actualizar ciudades	0	0	1	1
Agregar, listar y encontrar producto por ID, eliminar y actualizar producto	0	0	1	1
Encontrar productos por ciudad y por categoría	1	0	2	3



Buscar producto por ciudad y fecha				
Agregar, listar, eliminar y actualizar características	1	0	4	5
Listar imágenes por ID de producto	3	0	1	4
Agregar imagen de producto	0	1(JUnit)	1	2
Actualizar puntuaciones	0	1(JUnit)	1	2
Agregar usuarios	1	0	1	2
Login como administrador con JWT	2	0	1	3
Login como usuario cliente con JWT	2	0	1	3
Encontrar reserva por id de producto, y encontrar reserva por ID de usuario	0	0	2	2

Fuera de Alcance

	<i>Test Manuales</i>	<i>Test Automáticos</i>	<i>Test de Integración (Postman)</i>	<i>Test Total</i>
Calendario página principal seleccionar rango de fechas	1	0	0	1
librería de imágenes con auto deslizamiento	1	0	0	1



Libreria de imagenes que pasen cada 3 segundos	1	0	0	1
Calendario reservas se puede elegir rango de fechas válidas	3	0	0	3
Iconos se pueden visualizar	2	0	0	2

Ejecución de Pruebas

Ejecución Manual del sprint 4

Funcionalidades del sistema dentro del alcance	<i>Test Pasado</i>	<i>Test Fallados</i>	<i>Test no ejecutados</i>	<i>Test Total</i>
Usuario logueado mostrar el nombre de usuario, avatar con las iniciales del nombre del usuario y un link de cierre de sesión	2	0	0	2
Login de usuario	2	0	0	2
Crear cuenta usuario	3	0	0	3
Usuario al ingresar al dar click en "ver más" pueda visualizar los detalles del producto	23	0	0	23
Bloque buscador página principal	1	0	0	1
Filtrar por categoría página principal	1	0	0	1



Filtrar por ciudad página principal	3	0	0	3
Filtrar por fechas página principal	2	0	0	2
Calendario página principal seleccionar rango de fechas	1	0	0	1
Funcionalidad de botón “iniciar sesión”	1	0	0	1
Funcionalidad de botón “crear cuenta”	2	0	0	2
Funcionalidad del logo al dar clic regrese a pagina principal	1	0	0	1
Producto con flecha de volver a página anterior	1	0	0	1
Funcionalidad al botón de reserva en página de detalle de producto	2	0	0	2
Bloque de imágenes con botón de “Ver Más”, el cual, permitirá acceder a un componente de galería de imágenes	1	0	0	1
Galería con auto deslizamiento cada 3 segundos	1	0	0	1
Producto calendario con fechas disponibles	2	0	0	2
Página de reserva Icono que permita regresar	1	0	0	1
Formulario de reserva	1	0	0	1



Horario de llegada en reserva	1	0	0	1
Calendario Home Seleccionar múltiples fechas	1	0	0	1
Calendario en reserva permite mover en distintos meses	1	0	0	1
Selección de producto	1	0	0	1
Encontrar productos por ciudad y por categoría Buscar producto por ciudad y fecha	1	0	0	1
Agregar, listar, eliminar y actualizar características	1	0	0	1
Listar imágenes por ID de producto	1	0	0	1
Agregar usuarios	1	0	0	1
Login como administrador con JWT	1	0	0	1
Login como usuario cliente con JWT	3	0	0	3

Ejecución Automática del Sprint 4

Tests unitarios en JEST

Cubrimos más del 40% de componentes testeados.

All files

86.66% Statements 39/45 **72.72%** Branches 16/22 **88.23%** Functions 15/17 **86.36%** Lines 38/44

File		Statements		Branches		Functions		Lines	
Avatar	<div><div></div></div>	55.55%	5/9	71.42%	5/7	50%	1/2	55.55%	5/9
AvatarInitials	<div><div></div></div>	90.9%	10/11	83.33%	5/6	100%	3/3	90%	9/10
Boton	<div><div></div></div>	100%	5/5	80%	4/5	100%	1/1	100%	5/5
BotonesHeader	<div><div></div></div>	100%	4/4	50%	2/4	100%	2/2	100%	4/4
Botonvermas	<div><div></div></div>	100%	2/2	100%	0/0	100%	1/1	100%	2/2
Categoria	<div><div></div></div>	66.66%	2/3	100%	0/0	50%	1/2	66.66%	2/3
Contexto	<div><div></div></div>	100%	1/1	100%	0/0	100%	0/0	100%	1/1
Estrellas	<div><div></div></div>	100%	1/1	100%	0/0	100%	1/1	100%	1/1
Footer	<div><div></div></div>	100%	2/2	100%	0/0	100%	1/1	100%	2/2
Logo	<div><div></div></div>	100%	2/2	100%	0/0	100%	1/1	100%	2/2
PuntuacionProducto	<div><div></div></div>	100%	1/1	100%	0/0	100%	1/1	100%	1/1
ReservaTitulo	<div><div></div></div>	100%	3/3	100%	0/0	100%	1/1	100%	3/3
Ubicacion	<div><div></div></div>	100%	1/1	100%	0/0	100%	1/1	100%	1/1

Code coverage generated by Istanbul at Mon Dec 06 2021 20:19:03 GMT-0300 (hora estándar de Argentina)

Tests de integración en POSTMAN

API testada	Cantidad de tests	Resultado
Usuarios	3	Passed
Categorías	5	Passed
Ciudades	5	Passed
Reservas	4	Passed
Productos	9	Passed
Características	5	Passed
Imágenes	2	Passed
Puntuaciones	1	Passed

En el ANEXO 1 se encuentra el detalle de los tests realizados en Postman.

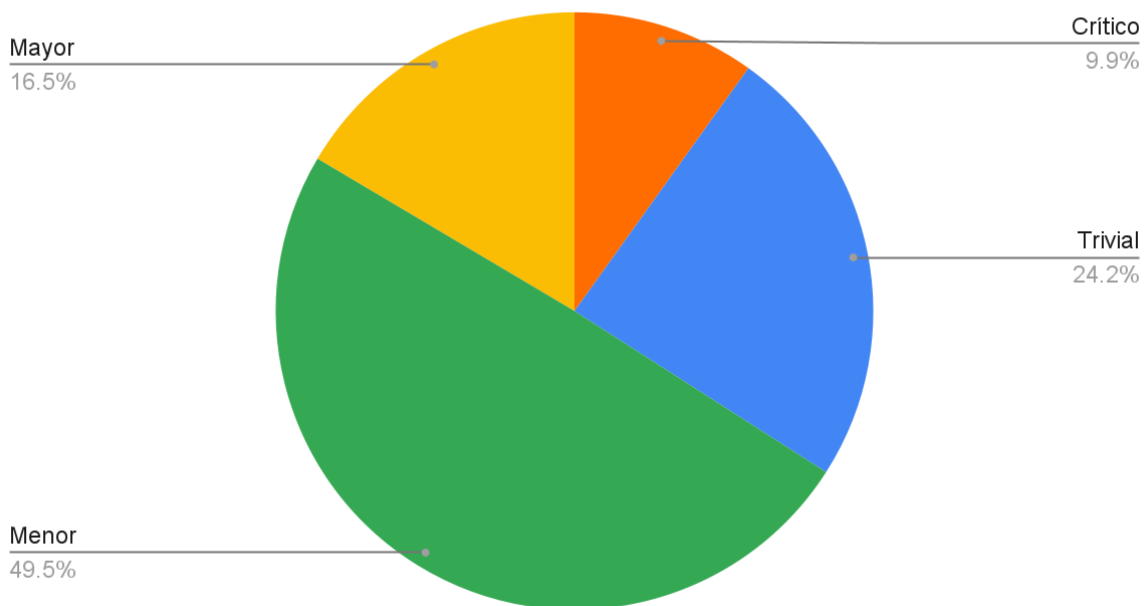


Reporte de Defectos

Todos los defectos

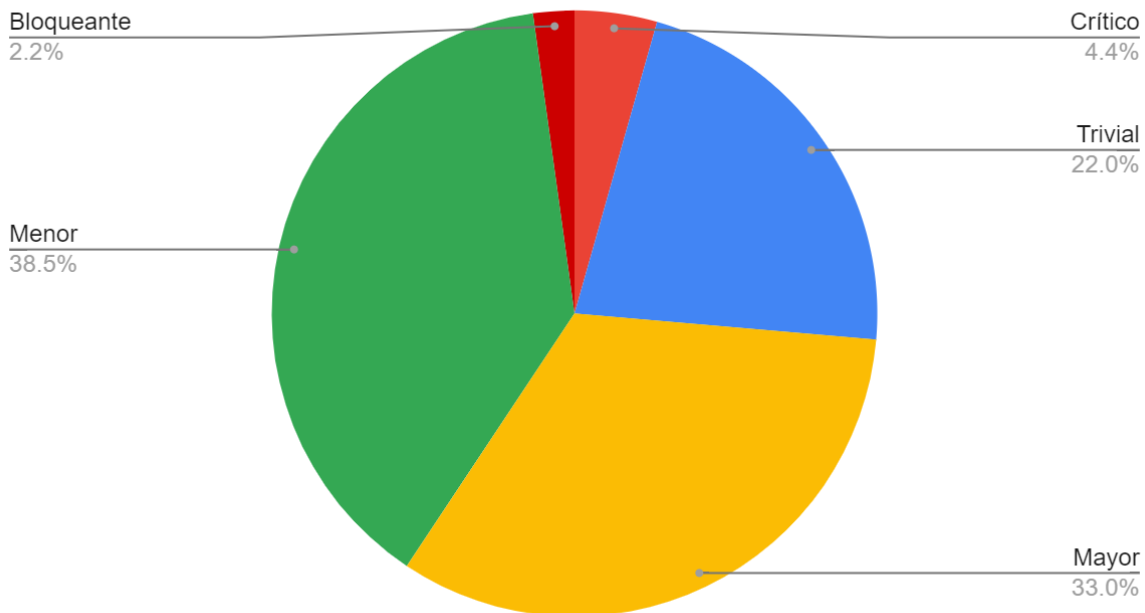
La siguiente sección muestra información con respecto al número total de defectos que se han presentado durante la duración de la fase de prueba.

Defectos por severidad

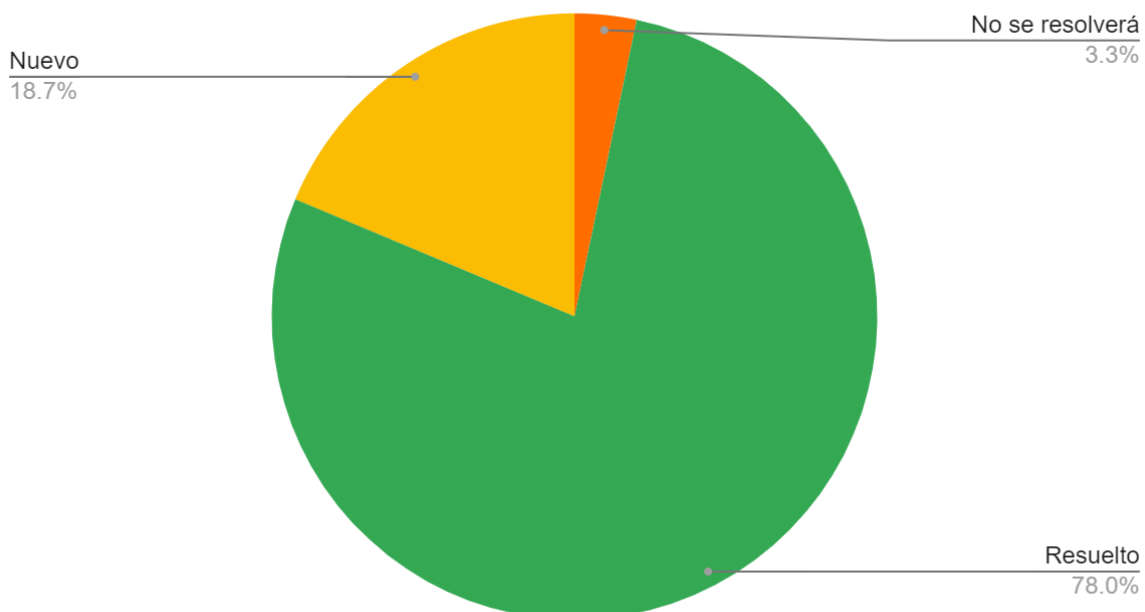




Defectos según Prioridad



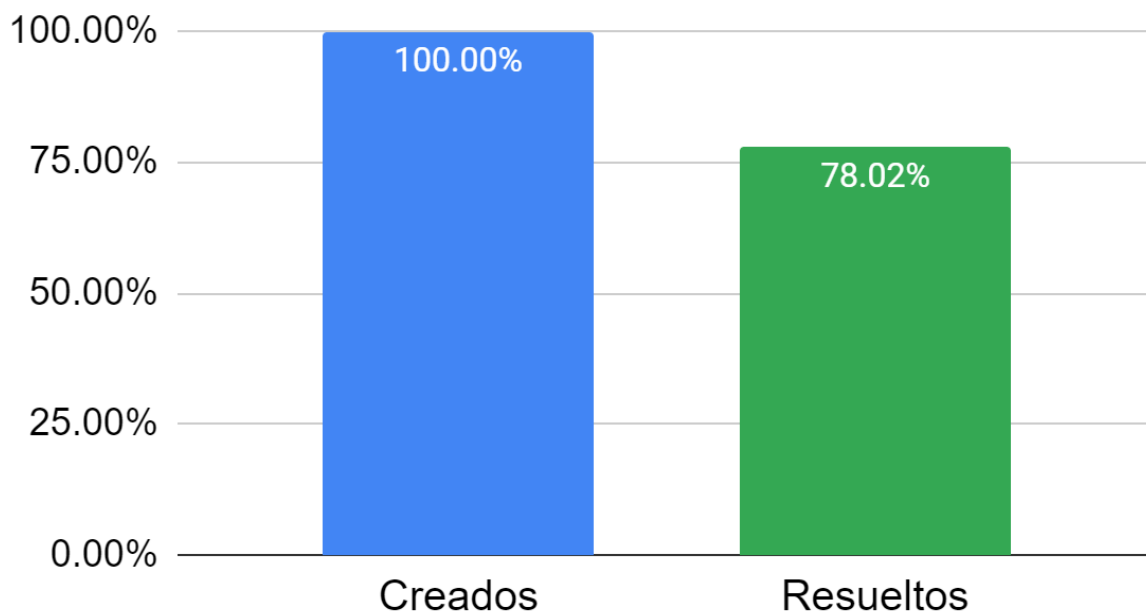
Estado de defectos





Defectos Creados vs Resueltos

Porcentaje de tests resueltos vs creados

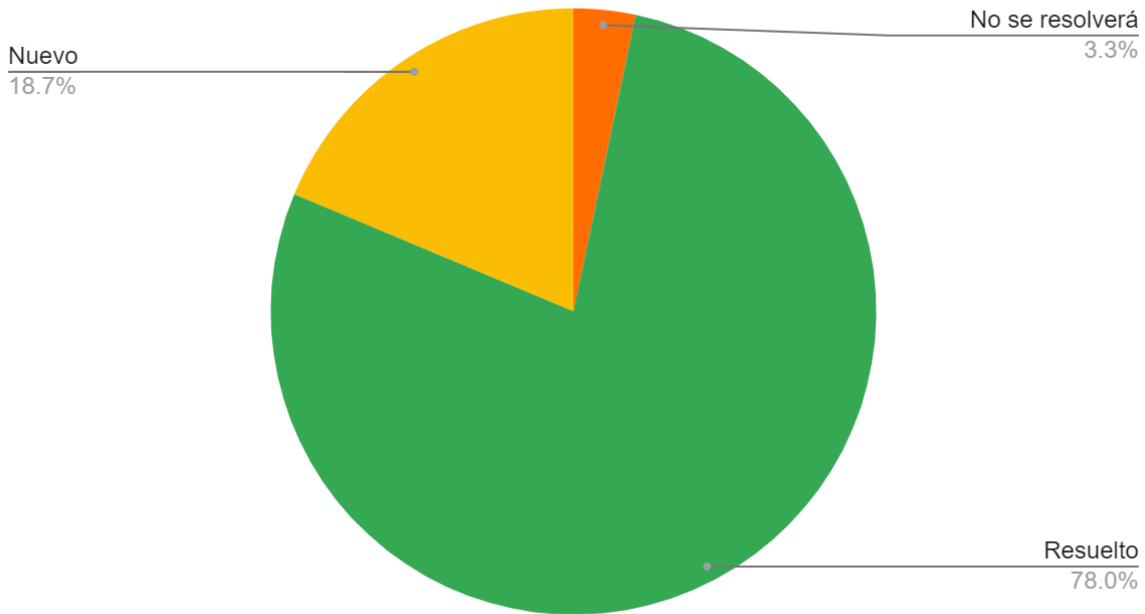


Defectos Abiertos

La siguiente sección muestra información con respecto al número total de defectos que permanecen abiertos al final de la fase de pruebas.



Estado de defectos



de acuerdo a la gráfica la cantidad de defectos que No se encuentran solucionados son 18.7%, cumpliendo con el exit criteria es decir no existen defectos abiertos de severidad y/o bloqueante

Lecciones Aprendidas / Conclusión

- Concluimos que el presente proyecto nos permitió aplicar todos los conocimientos adquiridos de testing , identificamos los 7 principios de testing a lo largo del proyecto, donde pudimos evidenciar que la prueba temprana ahorra tiempo y dinero y por esta razón es muy importante la continua detección de defectos.
- Las pruebas funcionales fueron fundamentales en nuestro proyecto porque así pudimos identificar las características críticas, la funcionalidad y la usabilidad; y así garantizar que estas características y funcionalidades se comporten según lo esperado sin ningún problema.
- Las pruebas unitarias fueron indispensables para probar unidades individuales de nuestro sistema con la finalidad de testear un comportamiento y obtener un resultado esperado.



- Fue fundamental realizar la clasificación de Suite de Regresión para verificar la continuidad y el buen funcionamiento de las funcionalidades preexistentes, posterior a la implementación de otra nueva, para mantener la integridad del sistema aunque hubieran cambios.
- Las pruebas exploratorias ayudan a elevar la calidad del Software ya que al final de cada una de ellas se genera un reporte de defectos, para luego clasificarlos en orden de prioridad y severidad y en base a eso resolverlos.
- Durante los cuatro sprints estuvimos investigando las metodologías para aplicar los test manuales de la forma más eficiente posible y las tecnologías disponibles para los test automatizados, lo cual requirió de bastante tiempo de estudio y dedicación para llevarse a cabo.

ANEXO 1

Detalle de tests realizados en Postman con su correspondiente Pass Status Code.

Test realizado	Evidencia
Agregar Usuario	<p>POST addUsers http://localhost:8085/usuarios / Usuarios / addUsers 201 Created</p> <p>Pass Status code is 201</p>
Añadir usuario mediante login	<p>POST login admin jwt http://localhost:8085/usuarios/login / Usuarios / login admin jwt 200 OK</p> <p>Pass Status code is 200</p>
Añadir usuario administrador mediante login	<p>POST login user jwt http://localhost:8085/usuarios/login / Usuarios / login user jwt 200 OK</p> <p>Pass Status code is 200</p>
Back/ Agregar categoría	<p>POST addCategory http://localhost:8085/categorias / Categorias / addCategory 200 OK 763 ms 577 B</p> <p>Pass Status code is 200</p> <p> Test Results 919 ms Hiber CategoriaServiceTest 919 ms 2021- crear_y_listar() 919 ms Hiber </p>
Back/ Listar todas las categorías	<p>GET getCategories http://localhost:8085/categorias / Categorias / getCategories 200 OK</p> <p>Pass Status code is 200</p> <p> Test Results 919 ms Hiber CategoriaServiceTest 919 ms 2021- crear_y_listar() 919 ms Hiber </p>



Buscar categoría por id	<div><div>GET findCategoryById http://localhost:8085/categorias/1 / Categorias / findCategoryById 200 OK</div><div>Pass Status code is 200</div><div><div>Test Results 517 ms Hibernate</div><div>CategoriaServiceTest 517 ms</div><div>✓ buscar_por_id() 517 ms</div></div></div>
Actualizar categoría	<div><div>PUT updateCategory http://localhost:8085/categorias / Categorias / updateCategory 200 OK</div><div>Pass Status code is 200</div><div><div>Test Results 490 ms Hibernate</div><div>CategoriaServiceTest 490 ms</div><div>✓ editar() 490 ms</div></div></div>
Eliminar categoría	<div><div>DELETE deleteCategory http://localhost:8085/categorias/5 / Categorias / deleteCategory 204 No Content</div><div>Pass Status code is 204 no content</div><div>Pass Status code name has string</div></div>
Agregar una ciudad	<div><div>POST addCity http://localhost:8085/ciudades / Ciudades / addCity 200 OK</div><div>Pass Status code is 200</div><div><div>Test Results 1 sec 8 ms Hibernate</div><div>CiudadServiceTest 1 sec 8 ms</div><div>✓ crear_ciudad() 1 sec 8 ms</div></div></div>
Listar todas las ciudades	<div><div>GET getCities http://localhost:8085/ciudades / Ciudades / getCities 200 OK</div><div>Pass Status code is 200</div><div><div>Test Results 553 ms 2021-12-</div><div>CiudadServiceTest 553 ms Hibernate</div><div>✓ listar_todas() 553 ms</div></div></div>
Buscar ciudad por id	<div><div>GET findCityById http://localhost:8085/ciudades/1 / Ciudades / findCityById 200 OK</div><div>Pass Status code is 200</div><div><div>Test Results 608 ms Hib</div><div>CiudadServiceTest 608 ms Hib</div><div>✓ buscar_por_id() 608 ms</div></div></div>
Actualizar ciudad	<div><div>PUT updateCity http://localhost:8085/ciudades / Ciudades / updateCity 200 OK</div><div>Pass Status code is 200</div><div><div>Test Results 505 ms Hibernate</div><div>CiudadServiceTest 505 ms</div><div>✓ editar_ciudad() 505 ms</div></div></div>



Eliminar ciudad	<div><div>DELETE</div><div>deleteCity</div><div>http://localhost:8085/ciudades/5</div><div>/ Ciudades / deleteCity</div><div>204 No Content</div></div> <div><div>Pass</div><div>Status code is 204 no content</div></div> <div><div>Pass</div><div>Status code name has string</div></div>
Agregar reserva	<div><div>POST</div><div>addReserves</div><div>http://localhost:8085/reservas</div><div>/ Reservas / addReserves</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Listar todas las reservas	<div><div>GET</div><div>getReserves</div><div>http://localhost:8085/reservas</div><div>/ Reservas / getReserves</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar reserva por id de producto	<div><div>GET</div><div>findReservesByProductId</div><div>http://localhost:8085/reservas/buscarPorIdProducto/1</div><div>/ Reservas / findReservesByProductId</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar reserva por id de usuario	<div><div>GET</div><div>findReservesByUserId</div><div>http://localhost:8085/reservas/buscarPorIdUsuario/2</div><div>/ Reservas / findReservesByUserId</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Agregar producto	<div><div>POST</div><div>addProduct</div><div>http://localhost:8085/productos</div><div>/ Productos / addProduct</div><div>201 Created</div></div> <div><div>Pass</div><div>Status code is 201</div></div>
Listar todos los productos	<div><div>GET</div><div>getProducts</div><div>http://localhost:8085/productos</div><div>/ Productos / getProducts</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar productos por id	<div><div>GET</div><div>findProductById</div><div>http://localhost:8085/productos/1</div><div>/ Productos / findProductById</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar productos por ciudad	<div><div>GET</div><div>findProductsByCity</div><div>http://localhost:8085/productos/buscarPorCiudad/1</div><div>/ Productos / findProductsByCity</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar productos por categoría	<div><div>GET</div><div>findProductsByCategory</div><div>http://localhost:8085/productos/buscarPorCategoria/1</div><div>/ Productos / findProductsByCateg...</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar productos por fechas	<div><div>POST</div><div>findProductByDates</div><div>http://localhost:8085/productos/buscarPorFechas</div><div>/ Productos / findProductByDates</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Buscar productos por ciudad y fechas	<div><div>POST</div><div>findByCityAndDates</div><div>http://localhost:8085/productos/buscarPorFechasYCiudad/1</div><div>/ Productos / findByCityAndDates</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>
Actualizar producto	<div><div>PUT</div><div>updateProduct</div><div>http://localhost:8085/productos</div><div>/ Productos / updateProduct</div><div>200 OK</div></div> <div><div>Pass</div><div>Status code is 200</div></div>



Eliminar producto	<div><div>DELETE</div><div>deleteProduct http://localhost:8085/productos/9 / Productos / deleteProduct</div><div>204 No Content</div><div><div>Pass</div><div>Status code is 204 no content</div></div><div><div>Pass</div><div>Status code name has string</div></div></div>
Agregar característica	<div><div>POST</div><div>addFeature http://localhost:8085/caracteristicas / Características / addFeature</div><div>200 OK</div><div><div>Pass</div><div>Status code is 200</div></div><div><div>JUnit Jupiter</div><div>435 ms</div></div><div><div>CaracteristicaServiceTest</div><div>435 ms</div></div><div><div>nueva_puntuacion()</div><div>435 ms</div></div></div>
Listar todas las características	<div><div>GET</div><div>getFeatures http://localhost:8085/caracteristicas / Características / getFeatures</div><div>200 OK</div><div><div>Pass</div><div>Status code is 200</div></div></div>
Buscar característica por id	<div><div>GET</div><div>getFeatureById http://localhost:8085/caracteristicas/9 / Características / getFeatureById</div><div>200 OK</div><div><div>Pass</div><div>Status code is 200</div></div></div>
Actualizar característica	<div><div>PUT</div><div>updateFeature http://localhost:8085/caracteristicas/9 / Características / updateFeature</div><div>200 OK</div><div><div>Pass</div><div>Status code is 200</div></div></div>
Eliminar característica	<div><div>DELETE</div><div>deleteFeature http://localhost:8085/caracteristicas/9 / Características / deleteFeature</div><div>204 No Content</div><div><div>Pass</div><div>Status code is 204 no content</div></div><div><div>Pass</div><div>Status code name has string</div></div><div><div>JUnit Jupiter</div><div>528 ms</div></div><div><div>CaracteristicaServiceTest</div><div>528 ms</div></div><div><div>eliminar_puntuacion()</div><div>528 ms</div></div></div>
Agregar imagen desde producto	<div><div>POST</div><div>addProductImage http://localhost:8085/productos / Imagenes / addProductImage</div><div>201 Created</div><div><div>Pass</div><div>Status code is 201</div></div></div>
Listar imágenes por id	<div><div>GET</div><div>ListarImagenesPorId http://localhost:8085/productos/1/imagenes / Imagenes / ListarImagenesPorId</div><div>200 OK</div><div><div>Pass</div><div>Status code is 200</div></div></div>
Actualizar puntuaciones	<div><div>PUT</div><div>updateScore http://localhost:8085/productos / Puntuaciones / updateScore</div><div>200 OK</div><div><div>Pass</div><div>Status code is 200</div></div></div>