



**Certified Tech
Developer**
The Ultimate Degree

DigitalHouse>
Coding School



Reporte Final de Testing

Integrantes:

- Lina María Fernández
- Tatiana Muñoz
- María Niveiros
- Leticia Pistarelli
- Leandro Rodríguez



Contenido

Introducción	3
Resumen de las actividades de prueba	3
Alcance	3
Dentro del Alcance	4
Fuera de Alcance	4
Tipos de Pruebas Ejecutadas	4
Enfoque de la Prueba	5
Exit Criteria	6
Resumen de Resultados	6
Diseño de Pruebas	6
Ejecución de Pruebas	7
Ejecución Manual	7
Ejecución Automática	8
Reporte de Defectos	9
Todos los defectos	10
Defectos por prioridad	10
Defectos por Severidad	11
Defecto por Estado	11
Defectos Creados vs Resueltos	12
Defectos Abiertos	12
Lecciones Aprendidas / Conclusión	12

Introducción

Este documento es el Informe Final de Pruebas del sistema Digital Car. 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

Digital Car es un sitio de renta de automóviles temporarios que brinda servicios en ciudades y aeropuertos de la República Argentina. El usuario del sitio puede filtrar su búsqueda por categorías, ciudades y fechas. Al registrarse en el sitio puede, a partir de esas búsquedas, reservar el vehículo que desee y consultar su historial de reservas. El administrador del sitio, puede gestionar los vehículos, agregando nuevos y modificando sus políticas de producto.

Se puede acceder al mismo a través de <http://www.digitalcar.ml/>

Planificación y ejecución de los tests #13 · created 1 month ago by Ezequiel Cortés	🕒 Sprint 1	En progreso	Testing / QA	CLOSED	👁 1	🗨 0	updated 1 month ago			
Testear las APIs #14 · created 1 month ago by Ezequiel Cortés	🕒 Sprint 1	Testing / QA		CLOSED	👁 1	🗨 0	updated 1 month ago			
Implementar tests automatizados #32 · created 1 month ago by Ezequiel Cortés	🕒 Sprint 2	A verificar	En progreso	Obligatoria	Testing / QA		CLOSED	🌐 0	🗨 0	updated 3 minutes ago
Testing Manual #34 · created 1 month ago by Daniel Tejerina	🕒 Sprint 2	Obligatoria	Testing / QA				CLOSED	👁 0	🗨 0	updated 4 weeks ago
Implementar tests manuales #55 · created 3 weeks ago by Ezequiel Cortés	🕒 Sprint 3	En progreso	Obligatoria	Testing / QA			CLOSED	👁 0	🗨 0	updated 2 weeks ago
Implementar tests automatizados #56 · created 3 weeks ago by Ezequiel Cortés	🕒 Sprint 3	En progreso	Obligatoria	Testing / QA			CLOSED	🌐 0	🗨 0	updated 2 minutes ago
Implementar testing automatizado #62 · created 1 week ago by Ezequiel Cortés	🕒 Sprint 4	Obligatoria	Testing / QA				CLOSED	🌐 0	🗨 0	updated 2 minutes ago
Implementar testing manual #63 · created 1 week ago by Ezequiel Cortés	🕒 Sprint 4	Obligatoria	Testing / QA				CLOSED	👁 0	🗨 0	updated 5 hours ago
Confeccionar el reporte final de testing #64 · created 1 week ago by Ezequiel Cortés	🕒 Sprint 4	Obligatoria	Testing / QA				CLOSED	👁 0	🗨 0	updated 5 hours ago

Alcance

Dentro del Alcance

Las pruebas ejecutadas durante los 4 Sprints, abarcaron las siguientes funcionalidades:

- Registro de usuario
- Login de usuario
- Registro de administrador
- Login de administrador
- Búsqueda de vehículos (filtros: por categorías, por fechas, por ciudades y por fechas y ciudades)
- Reserva de vehículos a partir de las búsquedas
- Consulta de historial de reservas por usuario
- Alta de vehículos
- Modificación de políticas de producto

Fuera de Alcance

Quedaron fuera del alcance de las pruebas ejecutadas el funcionamiento del mapa de ubicación del vehículo, las recomendaciones y puntuaciones de los usuarios.

Tipos de Pruebas Ejecutadas

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Prueba Estática	✗	✓	✗	✗
Prueba Exploratoria	✓	✓	✓	✓
Prueba de Sistema	✗	✗	✓	✓
Prueba de Humo	✓	✓	✓	✓
Prueba de Regresión	✗	✓	✓	✓



Prueba de Componente	✓	✓	✓	✓
Prueba de Integración (Postman)	✓	✓	✓	✓
Pruebas de compatibilidad	✓	✓	✓	✓
Pruebas de usabilidad	✓	✓	✓	✓
Pruebas de confirmación	✓	✓	✓	✓
Pruebas de casos de uso	✓	✓	✓	✓
Pruebas funcionales	✓	✓	✓	✓

Enfoque de la Prueba

Al inicio de cada sprint se crearon los casos de prueba en base a los requerimientos especificados en las user stories, haciendo foco en crear casos de prueba positivos y negativos por cada funcionalidad.

Ante cada build recibido, se ejecutaban las pruebas previstas y se realizaba el informe de defectos correspondiente. Se ejecutaron las pruebas de integración utilizando Postman y se reportaron los defectos detectados. Posteriormente, se realizaron pruebas de confirmación para verificar que los defectos reportados como solucionados por el equipo de desarrollo pasen exitosamente las pruebas correspondientes.

Se desarrollaron pruebas unitarias para cada funcionalidad, en el backend testeando los services y los controllers con JUnit, y en el frontend se testearon los diferentes componentes utilizando Jest.

Antes de finalizar cada sprint, se ejecutó la prueba exploratoria correspondiente, quedando los resultados finales plasmados en los informes entregados y los defectos encontrados en la planilla correspondiente. A su vez se ejecutaron pruebas de compatibilidad con los exploradores Microsoft Edge, Google Chrome, Mozilla Firefox y Opera, en las versiones desktop, mobile y tablet.

Se ejecutaron pruebas de UI con Selenium Webdriver, para las funcionalidades de registro, login y reserva.

Link Planilla de Casos de Prueba:

[Panilla de casos de prugeba DigitalCar](#)

Link Planilla de Defectos:

[Bug Reports DigitalCar](#)



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.
- La ejecución de los unit tests con Jest debe tener un 40% de tasa de tests pasados.
- La ejecución de los unit tests con Junit debe tener un 40% de tasa de tests pasados.
- La ejecución de los tests de integración con Postman debe tener un 100% de tasa de tests pasados.
- La ejecución de los tests manuales deben tener un 90% de tasa de tests pasados.

Resumen de Resultados

Diseño de Pruebas

	Tests Manuales	Tests Automáticos Junit / Jest	Tests de integración - Postman	Tests de U Selenium Webdriverl	Tests Totales
Login de usuario	72	8	19	18	117
Registro de usuario	60	10	19	21	110
Búsqueda de producto	93	21	125	0	239
Realizar Reserva	60	18	8	1	87
Consultar historial de reservas	3	15	34	1	53
Administrar Producto	6	14	0	0	20
Modificar políticas	6	5	23	0	34
Sitio Responsive	8	0	0	0	8
Elementos y estilos	354	20	0	17	391
TOTALES	662	111	228	58	1059



Ejecución de Pruebas

Ejecución Manual

	Test Pasado	Test Fallados	Test no ejecutados	Test Total
Login de usuario	6	0	0	6
Registro de usuario	6	0	0	6
Búsqueda de producto	33	9	0	33
Realizar Reserva	24	9	0	24
Consultar historial de reservas	3	0	0	3
Administrar Producto	6	0	0	6
Modificar políticas	6	0	0	6
Sitio Responsive	2	0	0	2
	86	0	0	86



Ejecución Automática

DigitalCar - Grupo 5 Local, 7 mins ago

RUN SUMMARY

			1
▶	POST	Categories - Add new category	9 0
▶	GET	Categories - Get all categories	7 0
▶	PUT	Categories - Put category by id	9 0
▶	DELETE	Categories - Delete category	9 0
▶	GET	Categories - Check deleted category	7 0
▶	POST	Cities - New city	9 0
▶	POST	Cities - New city with duplicated values	3 0
▶	GET	Cities - Get all cities	23 0
▶	POST	Features - Add new feature	9 0
▶	GET	Features - Get all features	17 0
▶	POST	Cars - Add new car	9 0
▶	GET	Cars - Get all cars	25 0
▶	GET	Cars - Get car by id	25 0
▶	GET	Cars - Filter car by city name	25 0
▶	GET	Cars - Filter car by category name	25 0
▶	GET	Cars - Filter cars by start date	25 0
▶	GET	Images - Get all images	19 0
▶	POST	Users - Register new user	19 0
▶	POST	Users - Login registered user	19 0
▶	POST	Booking - Add new booking	8 0
▶	GET	Booking - Find by user	17 0
▶	GET	Booking - Find by car	17 0
▶	PUT	Politics - Update Policy	9 0
▶	GET	Politics - Get Politics	14 0



All files

44.78% Statements 388/670 32.57% Branches 114/358 36.68% Functions 84/229 45.45% Lines 388/668

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File		Statements	Branches	Functions	Lines	
src	<div><div></div></div>	38.1%	8/21	25%	3/12	25%
src/components/AddFeatures	<div><div></div></div>	54.55%	18/33	37.5%	3/8	50%
src/components/AddImages	<div><div></div></div>	31.58%	18/57	11.11%	2/18	22.22%
src/components/Administration	<div><div></div></div>	43.9%	18/41	21.43%	6/28	41.67%
src/components/Button	<div><div></div></div>	100%	1/1	100%	2/2	100%
src/components/Calendar	<div><div></div></div>	38.46%	5/13	63.64%	7/11	20%
src/components/CalendarReservation	<div><div></div></div>	41.38%	12/29	42.11%	8/19	37.5%
src/components/CardCategory	<div><div></div></div>	80%	4/5	100%	0/0	50%
src/components/CardDetailReservation	<div><div></div></div>	27.27%	6/22	7.14%	1/14	16.67%
src/components/CardProduct	<div><div></div></div>	91.67%	11/12	75%	6/8	83.33%
src/components/CategoryPicker	<div><div></div></div>	72.73%	8/11	100%	0/0	42.86%
src/components/FavoriteHeart	<div><div></div></div>	84.62%	11/13	83.33%	5/6	100%
src/components/Footer	<div><div></div></div>	100%	1/1	100%	0/0	100%
src/components/Gallery	<div><div></div></div>	37.5%	3/8	0%	0/2	20%

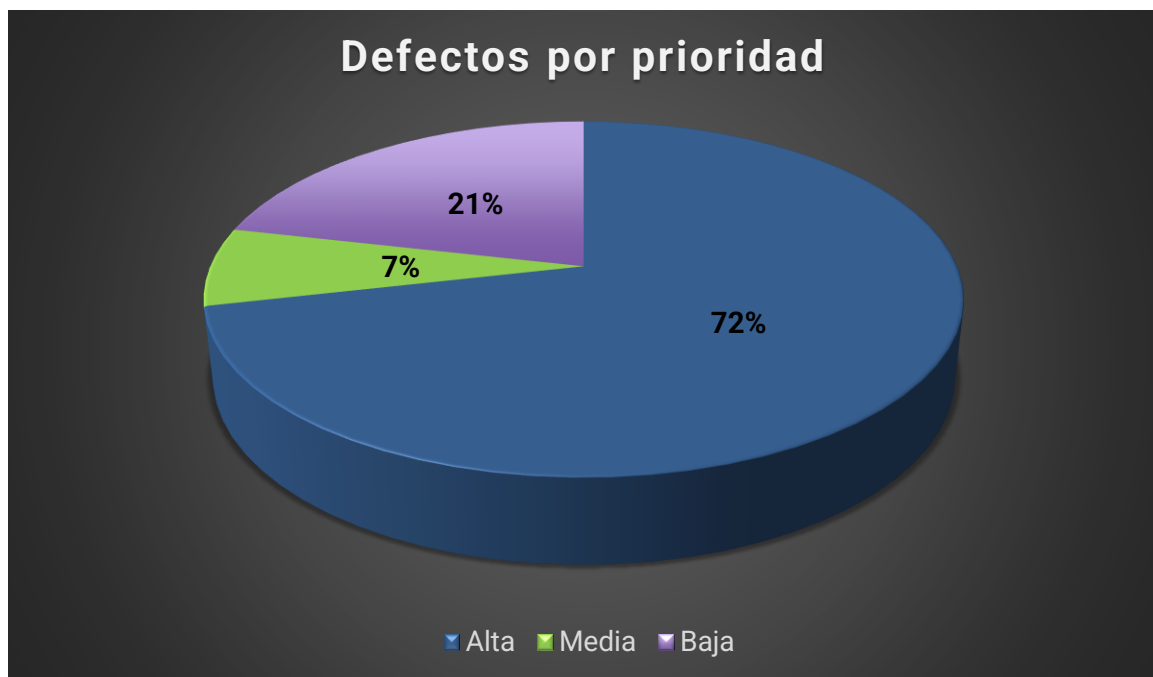


Reporte de Defectos

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 prioridad





Defectos por Severidad

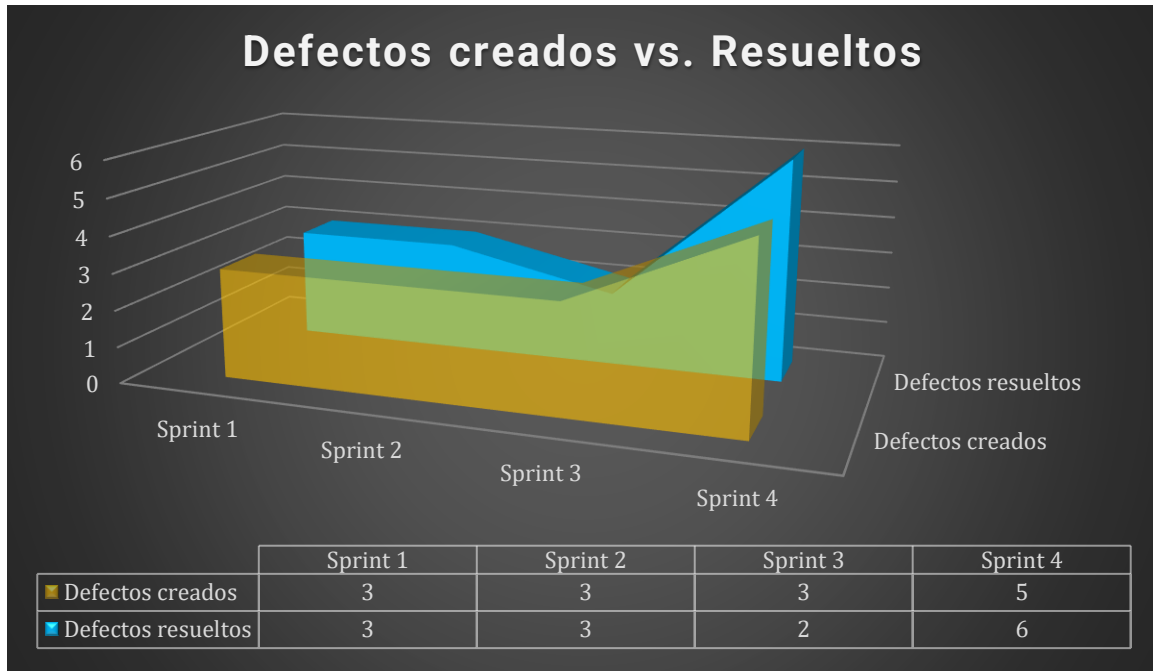


Defectos por Estado





Defectos Creados vs Resueltos



Defectos Abiertos

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

No se registran defectos abiertos al final de la fase de pruebas.

Lecciones Aprendidas / Conclusión

Durante este proceso de aprendizaje e implementación de los conocimientos adquiridos a lo largo de la cursada, pudimos conocer más en profundidad las tareas que realiza un QA.

Ampliamos nuestro dominio de las herramientas que habíamos utilizado anteriormente como Postman para realizar el Api Testing y Jest para React (la cual resultó un desafío muy importante). Mejoramos la forma de documentar nuestras planillas de casos de prueba, reportes de defectos e informes de pruebas exploratorias.

Toda esta experiencia nos permitió poner a prueba nuestro sistema, hacerlo fallar, buscar defectos y solucionarlos, haciendo de nuestro producto un sitio mucho más estable y confiable. Aún sabiendo que la prueba exhaustiva es imposible, procuramos abarcar la mayor cantidad de casos, a fin de minimizar los defectos existentes.