



# -VintageCar-

## Reporte Final de Testing

### Grupo 7

#### Integrantes:

- Katyana Prieto
- Nicolás Gambino
- Joaquín Mármol
- Patricia Del Castillo
- Karen Céspedes Rivas
- Fernando Pajón

## **-Introducción-**

El presente documento es el informe final de las actividades de testing llevadas a cabo durante el desarrollo de *Vintage Car*, el proyecto integrador final del primer año de la carrera Certified Tech Developer (CTD) de Digital House, llevado a cabo por el grupo siete.

El propósito final de este instrumento es mostrar las actividades realizadas desde el primer sprint y poner de manifiesto que los criterios de aceptación (exit criteria) fueron cumplidos y, por lo tanto, esta fase puede cerrarse. A su vez, el objetivo ulterior es también servir como herramienta para determinar si el sistema cumple con las expectativas requeridas.

### Resumen de las actividades de prueba

¿Qué es *Vintage Car*? Se trata de una plataforma que permite alquilar coches de estilo.

Para el backend y la base de datos se realizaron pruebas unitarias de integración a través de postman. Para el frontend, se ejecutaron tests manuales, exploratorios y se utilizó Selenium para validar el funcionamiento de algunos componentes.

<b>Nro</b>	<b>ISSUE</b>	<b>SPRINT</b>	<b>ESTADO</b>
12	Planificación y ejecución de los test.	1	Cerrado
13	Testear la API.	1	Cerrado
46	Testeo automatizado.	2	Cerrado
48	Testeo con Postman.	2	Cerrado
49	Testeo exploratorio, manual, estático.	2	Cerrado
75	Implementación tests manuales	3	Cerrado
76	Implementar tests automatizados	3	Cerrado
87	Implementar testing automatizado	4	Cerrado
88	Implementar testing manual	4	Cerrado

## **-Alcance-**

### Dentro del alcance

Con testeo manual se probó la correcta implementación y gestión de ciudades, características del producto, el formulario de reserva, los inputs de usuario y contraseña, la visualización de las distintas cards, el footer y las tablas de la base de datos (aunque también se automatizó parte de estas pruebas con postman).

Con testeo automatizado se probó y verificó el CRUD de la API, el Security y la integración del sistema. También se probaron las funciones referidas a la creación de usuario, al inicio de sesión erróneo e incorrecto, el calendario, la existencia del template reserva, la correcta reserva, el formulario de producto, el bloque buscador, el enrutador al home, eventos de login, responsive calendario, la visualización de productos y descripciones, entre otros.

### Fuera de alcance

Quedaron fuera de alcance los test relativos a la confirmación de cuenta de usuario, a la latitud y longitud del mapa, y lo relativo a pruebas de seguridad y estrés.

## Tipos de pruebas ejecutadas

	SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 4
Prueba estática	✓	✓	✓	✓
Prueba exploratoria	✓	✓	✓	✓
Prueba de humo	✓	✓	✓	✓
Prueba de regresión	X	✓	✓	✓
Prueba de componente	✓	✓	✓	✓

### **-Enfoque de la prueba-**

Inicialmente se leía detenidamente el backlog de cada sprint y nos las dudas que pudiéramos tener las despejábamos con el product owner durante la planning.

A continuación, en una planilla de hoja de cálculo, disponíamos las issues y la dividíamos equitativamente, pero sabiendo que en cualquier momento podíamos rotar en nuestras tareas si era necesario para asegurar que el proyecto avance.

Desde el área de testing, realizamos test negativos y positivos a lo largo de todo el desarrollo. Creamos una rama para subir al repositorio todo lo relativo al testing. Durante el transcurso de primer sprint, nos dedicamos al testeo puramente manual porque nos encontrábamos todavía en el momento de diseño de la página, por lo que no había regresión para testear o automatizar.

Durante el segundo sprint se dió la posibilidad de usar jest. Aunque nuestro equipo optó al principio por implementarlo en el desarrollo de nuestra página, y logramos usarlo para testear componentes unitarios como algunas card, filtros y demás, luego lo descartamos por problemas de configuración, puesto que jest entraba en conflicto con algunas dependencias (packaje.json) y herramientas relacionadas a react.

En este sprint, todo se probaba en localhost, llevando a cabo testeo manual y automático de las funcionalidades más relevantes, quienes eran examinadas positiva y negativamente para asegurar la mayor calidad y descartar errores “ocultos” o difíciles de encontrar si solo hacíamos happy path.

El tercer sprint fue probablemente el más desafiante y el que más requirió de nuestro esfuerzo. El uso de postman

para testar la API luego de que el back añadiera seguridad fue fundamental para asegurar la calidad exigida.

El front end siguió testeándose fundamentalmente de manera manual y con testeo de regresión a través de Selenium Web Driver. En estos sprints, iba incrementándose obviamente la cantidad de test de regresión y con un incremento en la cantidad de casos de prueba enfocados en las funcionalidad primordiales logramos abarcar en un alto porcentaje las issues requeridas, quedando solo algunos carry overs para el siguiente sprint.

El último sprint encontró a los responsables de testing, front end y back end con una comunicación más fluida y con mayor gimnasia a la hora de desarrollar y asegurar calidad. Así, cada vez que una build era entregada se testeaba extensamente y se realizan sesiones de testeo exploratorio donde se dejaba constancia de los resultados o los posibles defectos que pudieran haberse encontrado.

Issues						
Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda						
100% 123 Predetermi... 9 B I S A						
A1	A	B	C	D	E	F
1		Titulo	Issues	sprint	Estimación	Realizada
2	sprint 1	Implementar template general responsive	Issue 1	sprint 1	L	✓
3		Crear la tabla de Categorías	Issue 2	sprint 1	S	✓
4		CRUD Categorías	Issue 3	sprint 1	M	✓
5		Crear API	Issue 4	sprint 1	M	✓
6		API: Agregar endpoint de categorías	Issue 5	sprint 1	M	✓
7		Implementación template bloque buscador	Issue 6	sprint 1	L	✓
8		Implementar el template del bloque Listado	Issue 7	sprint 1	M	✓
9		Implementar el template del bloque Categorías	Issue 8	sprint 1	M	✓
10		Implementar el template de Login y registro	Issue 9	sprint 1	L	✓
11		Evento de login de usuario	Issue 10	sprint 1	M	✓
12		Implementar Evento de Header	Issue 11	sprint 1	M	✓
13		Planificación y ejecución de los tests	Issue 12	sprint 1	M	✓
14		Testear la API	Issue 13	sprint 1	M	✓
15		Diseño de la red	Issue 14	sprint 1	M	✓
16	sprint 2	Generar enrutador para home y productos	Issue 15	sprint 1	S	✓
17		Crear controlador de productos	Issue 16	sprint 1	M	✓
18		Completar información del frontend de producto con datos de la API	Issue 17	sprint 1	M	✓
19	necesita apis	Implementar template responsive de producto	Issue 18	sprint 2	XL	✓
20	sin redondeado	Agregar bloque de imágenes y galería a template producto	Issue 19	sprint 2	L	✓
21		Crear tabla "productos" en la base de datos	Issue 20	sprint 2	S	✓
22		Agregar las acciones de crear, listar y buscar por id al controlador de productos	Issue 21	sprint 2	M	✓

## -Exit Criteria-

Fueron establecidos como criterios de salida, los siguientes:

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

- No debe tener defectos en estado abierto de severidad crítica y/o bloqueante.
- No se debe tener defectos en estado abierto de prioridad crítica.
- La ejecución de las pruebas de regresión deben tener mínimo un 80% de éxito.
- Los tests de postman de issues esenciales requieren 100% de éxito.



**-Resumen de resultados-**

	Test Manuales	Test automáticos	Test de integración con Postman
Login usuario	3	2	3
Creación de cuenta	4	2	3
Seleccionar Producto	2	0	6
Reserva	2	3	9
Formulario Crear Producto	3	2	5
Filtros productos	2	2	4
Responsive	6	0	0
Calendario	2	1	0
Botones	1	0	0
Tablas Bases de Datos	8		12
Seguridad API	4	1	6
Form reserva	5	4	3

Postman:

Autenticación global				<a href="#">Fork</a>   0   <a href="#">Save</a>   <a href="#">Share</a>   <a href="#">...</a>			
	VARIABLE	TYPE ①	INITIAL VALUE ①	CURRENT VALUE ①	...	Persist All	Reset All
<input checked="" type="checkbox"/>	BASE_URL	default ▾	http://3.14.246.73:8080/vint...	http://3.14.246.73:8080/vintage			
<input checked="" type="checkbox"/>	AUTH_URL	default ▾	{{BASE_URL}}/authenticate	{{BASE_URL}}/authenticate			
<input checked="" type="checkbox"/>	JWT	default ▾		eyJhbGciOiJIUzI1NiJ9.eyJ2ZXJpZmlyY...			
	Add a new variable						

Projecto integrador - grupo 7 / Authentication / Post authenticathe

Save

POST

{{AUTH\_URL}}

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

1 pm.environment.set("JWT", pm.response.json().jwt)

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts ↗

Snippets

Get an environment variable

Get a global variable

Body

Cookies

Headers (14)

Test Results

200 OK 488 ms 737 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1 "jwt": "eyJhbGciOiJIUzI1NiJ9.

2 eyJ2ZXZpZmllYWRvIjp0cnVlCjZdWII0iJmZXJuYW5kb3Bham9uMDAZ21haWwuY29tIiwibmFtZSI6IkZlcml5bmRvIiwibGZhdGUiOiJpbnVlUGFqb24iLCJpZCI6ImZIsImV4cCI6MTY3MDQ1OTgxNSwiaWF0IjoxNjcwNDUyMjE1LCJyb2wiOiJ0nsiaWQiOjEsIm5vbWJyZSI6IlJPTEVFQURNSU4ifX0.yV5pIATcgU0JMfB\_msZCYVjWGRz U3d4u3D7GNBYnw"

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	2s 479ms	2	304 ms

All Tests Passed (2) Failed (0) Skipped (0) [View Summary](#)

Iteration 1

**POST** Post authenticathe {{AUTH\_URL}} / Authentication / Post authenticathe 200 OK 327 ms 737 B

Pass Status code is 200

**POST** Post authenticathe (USUARIO SIN PERMISOS) {{AUTH\_URL}} / Authentication / Post aut... 403 Forbidden 281 ms 551 B

Pass Status code is 403

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	1s 757ms	4	241 ms

All Tests Passed (4) Failed (0) Skipped (0) [View Summary](#)

Iteration 1

**GET** Get características {{BASE\_URL}}/caracteristica / características / Get características 200 OK 363 ms 1.116 KB

Pass Status code is 200

**POST** Post características {{BASE\_URL}}/caracteristica / características / Post características 200 OK 191 ms 464 B

Pass Status code is 200

**PUT** Put características {{BASE\_URL}}/caracteristica / características / Put características 200 OK 211 ms 477 B

Pass Status code is 200

**DELETE** Del características {{BASE\_URL}}/caracteristica/37 / características / Del características 200 OK 200 ms 473 B

Pass Status code is 200

🍪 Cookies 🔄 Capture requests 🏃 Runner 🗑️ Trash 🏠 🗑️

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	3s 103ms	5	247 ms

All Tests Passed (5) Failed (0) Skipped (0) [View Sumr](#)

teration 1

GET	Get categorias	{{BASE_URL}}/categorias	/ categoria / Get categorias	200 OK	387 ms	1.154 KB
	Pass	Status code is 200				
POST	Post categorias	{{BASE_URL}}/categorias	/ categoria / Post categorias	200 OK	212 ms	665 B
	Pass	Status code is 200				
PUT	Put categorias	{{BASE_URL}}/categorias	/ categoria / Put categorias	200 OK	220 ms	472 B
	Pass	Status code is 200				
DELETE	Del categorias	{{BASE_URL}}/categorias/35	/ categoria / Del categorias	200 OK	223 ms	468 B
	Pass	Status code is 200				
POST	Post categorias (NO ADMIN)	{{BASE_URL}}/categorias	/ categoria / Post categorias (N...	403 Forbidden	192 ms	549 B
	Pass	Status code is 403				

[Cookies](#) [Capture requests](#) [Runner](#) [Trash](#) [⌵](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	1s 975ms	4	298 ms

All Tests Passed (3) Failed (1) Skipped (0) [View Summa](#)

teration 1

GET	Get imagenes	{{BASE_URL}}/imagen	/ imagenes / Get imagenes	200 OK	567 ms	34.349 KB
	Pass	Status code is 200				
POST	Post imagen	{{BASE_URL}}/imagen	/ imagenes / Post imagen	200 OK	213 ms	589 B
	Fail	Status code is 200   TypeError: Cannot read properties of undefined (reading 'to')				
PUT	Put imagen	{{BASE_URL}}/imagen	/ imagenes / Put imagen	200 OK	208 ms	470 B
	Pass	Status code is 200				
DELETE	Del imagen	{{BASE_URL}}/imagen/180	/ imagenes / Del imagen	200 OK	204 ms	466 B
	Pass	Status code is 200				

[Cookies](#) [Capture requests](#) [Runner](#) [Trash](#) [⌵](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	1s 849ms	4	290 ms

All Tests

Passed (4)

Failed (0)

Skipped (0)

[View Summary](#)

Iteration 1

GET

Get politica

{{BASE\_URL}}/politica

/ politica / Get politica

200 OK

546 ms

14.125 KB

Pass

Status code is 200

POST

Post politica

{{BASE\_URL}}/politica

/ politica / Post politica

200 OK

199 ms

1.049 KB

Pass

Status code is 200

PUT

Put politica

{{BASE\_URL}}/politica

/ politica / Put politica

200 OK

208 ms

471 B

Pass

Status code is 200

DELETE

Del politica

{{BASE\_URL}}/politica/46

/ politica / Del politica

200 OK

205 ms

467 B

Pass

Status code is 200

Cookies

Capture requests

Runner

Trash

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	4s 118ms	9	289 ms
<div>All TestsPassed (9)Failed (0)Skipped (0)<div>View Summary</div></div>					
Iteration 1					
GET	Get productos	{{BASE_URL}}/producto	/ producto / Get productos	200 OK	629 ms35.288 KB
	Pass	Status code is 200			
	Pass	Response time is less than 200ms			
GET	Get producto por ID	{{BASE_URL}}/producto/1	/ producto / Get producto por ID	200 OK	203 ms2.911 KB
	Pass	Status code is 200			
	Pass	Response time is less than 200ms			
POST	Post productos	{{BASE_URL}}/producto	/ producto / Post productos	200 OK	286 ms2.757 KB
	Pass	Status code is 200			
	Pass	Response time is less than 200ms			
PUT	Put producto	{{BASE_URL}}/producto	/ producto / Put producto	200 OK	205 ms471 B
	Pass	Status code is 200			
DELETE	Del producto	{{BASE_URL}}/producto/43	/ producto / Del producto	200 OK	217 ms467 B
	Pass	Status code is 200			
POST	Post producto (NO ADMIN)	{{BASE_URL}}/producto	/ producto / Post producto (NO ADMIN)	403 Forbidden	191 ms547 B
	Pass	Status code is 403			

Cookies

Capture requests

Runner

Trash

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	4s 299ms	6	308 ms
All Tests Passed (6) Failed (0) Skipped (0) <a href="#">View Summary</a>					
Iteration 1					
GET	Get reservas	{{BASE_URL}}/reserva / reservas / Get reservas	200 OK	781 ms	76.288 KB
	Pass	Status code is 200			
GET	Get reserva por ID	{{BASE_URL}}/reserva/2 / reservas / Get reserva por ID	200 OK	207 ms	3.371 KB
	Pass	Status code is 200			
POST	Post reservas	{{BASE_URL}}/reserva / reservas / Post reservas	200 OK	226 ms	1.132 KB
	Pass	Status code is 200			
PUT	Put reservas	{{BASE_URL}}/reserva / reservas / Put reservas	200 OK	240 ms	470 B
	Pass	Status code is 200			
DELETE	Delete Reserva	{{BASE_URL}}/reserva/52 / reservas / Delete Reserva	200 OK	208 ms	466 B
	Pass	Status code is 200			
POST	Post reserva (NO ADMIN)	{{BASE_URL}}/reserva / reservas / Post reserva (NO ADMIN)	403 Forbidden	188 ms	546 B
	Pass	Status code is 403			

Cookies Capture requests Runner Trash

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	3s 56ms	3	587 ms
All Tests Passed (3) Failed (0) Skipped (0) <a href="#">View Summary</a>					
Iteration 1					
GET	Get usuarios	{{BASE_URL}}/usuario / usuarios / Get usuarios	200 OK	208 ms	8.215 KB
	No tests in this request.				
GET	Get usuarios por ID	{{BASE_URL}}/usuario/32 / usuarios / Get usuarios por ID	200 OK	201 ms	849 B
	Pass	Status code is 200			
POST	Post usuario	{{BASE_URL}}/usuario / usuarios / Post usuario	201 Created	1720 ms	935 B
	Pass	Status code is 200			
DELETE	Del usuario	{{BASE_URL}}/usuario/59 / usuarios / Del usuario	200 OK	220 ms	464 B
	Pass	Status code is 200			

Cookies Capture requests Runner Trash

## Selenium:

http://localhost:3000			
	Command	Target	Value
1	✓ open	/	
2	✓ set window size	704x502	
3	✓ type	name=myBrowser	Buenos aires
4	✓ type	name=myBrowser	Rosario
5	✓ type	name=myBrowser	Cordoba
6	✓ click	css=.contenedorbuscador	
7	✓ type	name=myBrowser	Mendoza
8	✓ click	css=.FechaRetiro	
9	✓ click	css=.react-datepicker__day--025 > font > font	
10	✓ click	css=.calendario2	
11	✓ click	css=.react-datepicker__day--030	
12	✓ click	css=.react-datepicker__day--030 > font > font	
13	✓ click	css=.contenedorbotones	
14	✓ click	css=.boton > font > font	



**Running 'bloque buscador'**

1. open on / OK
2. setWindowSize on 704x502 OK
3. type on name=myBrowser with value Buenos aires OK
4. type on name=myBrowser with value Rosario OK
5. type on name=myBrowser with value Cordoba OK
6. click on css=.contenedorbuscador OK
7. type on name=myBrowser with value Mendoza OK
8. click on css=.FechaRetiro OK
9. click on css=.react-datepicker\_\_day--025 > font > font OK
10. click on css=.calendario2 OK
11. click on css=.react-datepicker\_\_day--030 OK
12. click on css=.react-datepicker\_\_day--030 > font > font OK
13. click on css=.contenedorbotones OK
14. click on css=.boton > font > font OK

**'bloque buscador' completed successfully**

The screenshot shows the Selenium IDE interface. At the top, there's a toolbar with icons for play, stop, refresh, and a dropdown menu. Below the toolbar, the address bar shows 'http://localhost:3000/'. A table lists the commands and their targets:

	Command	Target
1	open	http://localhost:3000/home
2	set window size	1382x744
3	click	xpath=//nav[@id='navbar']/div/a/span[4]

Below the table, the text 'Running 'home'' is displayed. Underneath, a list shows the execution progress:

1. open on http://localhost:3000/home OK

At the bottom, the status 'home' completed successfully is shown in green text.

http://localhost:3000/			
	Command	Target	Value
1	✓ open	http://localhost:3000/	
2	✓ set window size	702x500	
3	✓ click	css=#offcanvasNavbarLabel-expand-sm .btn-close	
4	✓ type	id=formBasicNombre	patri@gmail.com
5	✓ type	id=formBasicPassword	hola123
6	✓ mouse down at	css=html	-20,333
7	✓ mouse move at	css=html	-20,333
8	✓ mouse up at	css=html	-20,333
9	✓ click	css=html	
10	✓ click	css=.container	

Log	Reference
<b>Running 'evento login usuario'</b>	
1. open on http://localhost:3000/	OK
2. setWindowSize on 702x500	OK
3. click on css=#offcanvasNavbarLabel-expand-sm .btn-close	OK
4. type on id=formBasicNombre with value patri@gmail.com	OK
5. type on id=formBasicPassword with value hola123	OK
6. mouseDownAt on css=html with value -20,333	OK
7. mouseMoveAt on css=html with value -20,333	OK
8. mouseUpAt on css=html with value -20,333	OK
9. click on css=html	OK
10. click on css=.container	OK
11. click on css=.formularioButton	OK
<b>'evento login usuario' completed successfully</b>	

http://3.144.28.205

	Command	Target
1.	✓ open	/
2.	✓ set window size	1296x696
3.	✓ click	css= cardContainer:nth-child(1) a font > font
4.	✓ run script	window.scrollTo(0,0)
5.	✓ run script	window.scrollTo(0,599.3333129882812)
6.	✓ set window size	720x1240
7.	✓ click	css= botonReservaCalendario

'ResponsiveMobile' completed successfully

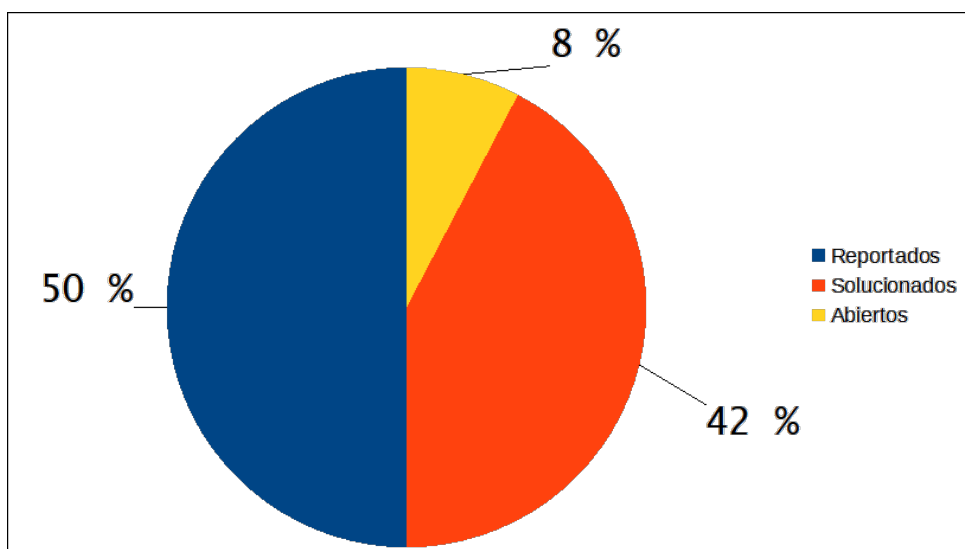
Running 'ResponsiveMobile'

- open on / OK
- setWindowSize on 1296x696 OK
- Trying to find css= cardContainer:nth-child(1) a font > font... OK
- runScript on window.scrollTo(0,0) OK
- runScript on window.scrollTo(0,599.3333129882812) OK
- setWindowSize on 720x1240 OK
- click on css= botonReservaCalendario OK

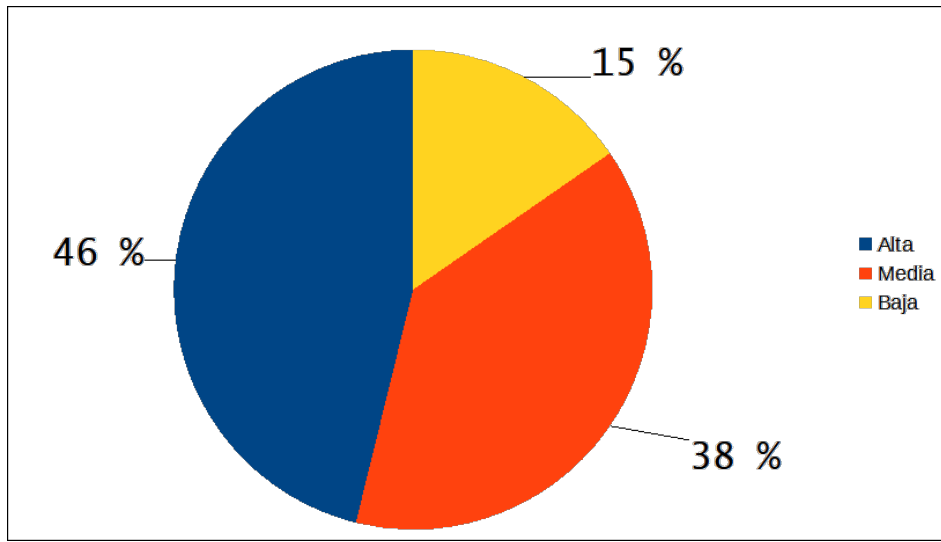
'ResponsiveMobile' completed successfully

## -Reporte de Defectos-

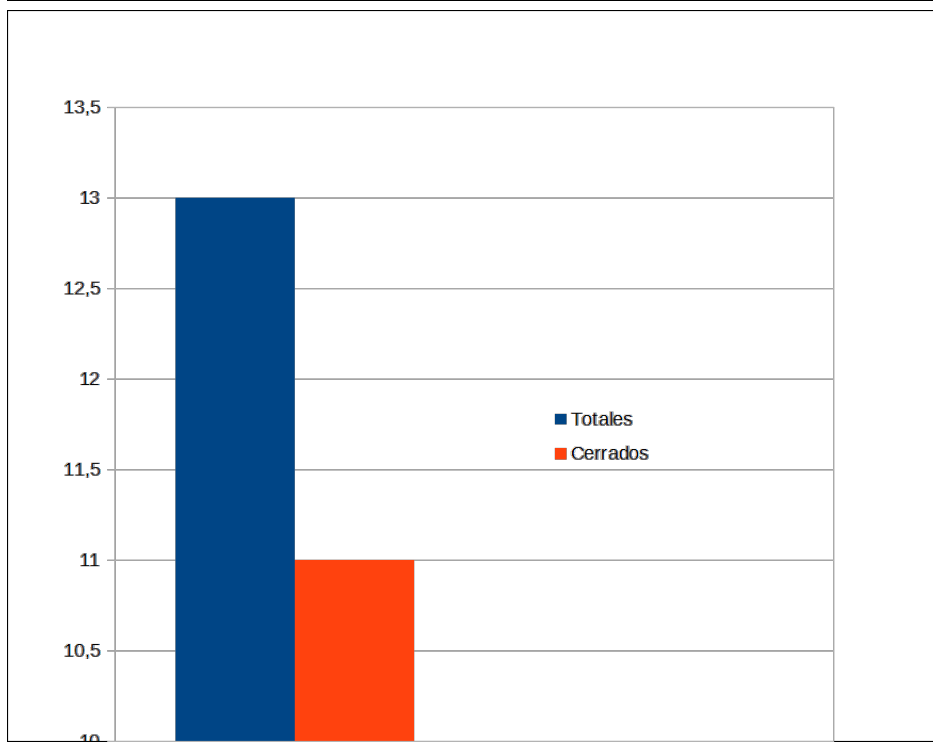
### Defectos por estado:



## Defectos por severidad



## Defectos totales creados / defectos cerrados.



## **-Lecciones Aprendidas. Conclusión-**

El recorrido por todos los sprints nos hizo comprender la gran importancia del testing durante el proceso de desarrollo a medida que se iban agregando e integrando más funcionalidades.

El testeo continuo fue de gran ayuda para encontrar, asignar y relevar mucho más rápidamente los defectos y poder actuar en consecuencia, permitiéndonos optimizar el tiempo, que es probablemente lo más importante, y la mejor estrategia que podemos seguir para lograr cumplir con las entregas de cada sprint.

La posibilidad de experimentar con herramientas que vimos en la cursada es una de las enseñanzas más gratificantes para quienes hicimos testing en este proyecto integrador, como así también haber afianzado los conceptos que aprendimos en clases. Explorar el producto desde diferentes ángulos, con distintas enfoques, ópticas y herramientas nos permitió entregar un producto con una calidad que, para nosotros que no teníamos experiencia en este rubro, nos satisface grandemente y no podemos

dejar de agradecer a todo el equipo de Digital House por hacerlo posible.

Entendimos también lo primordial que es el testing en un proceso como este, convirtiéndose en algo totalmente fundamental para que el producto final tenga una calidad aceptable, con la menor cantidad de errores posibles, accesible, adaptable y con valor para el cliente.