



**Certified Tech
Developer**
The Ultimate Degree

DigitalHouse >
Coding School

Reporte Final de Testing

Rent Me Up

Integrantes:

Florencia Caico

Carina Groisman

Rocío Jimenez

Elena Fischietto

Wilmar Puerta

Natalia Peralta Acosta



Contenido

Introducción	3
Resumen 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 Rent Me Up. 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

A través del siguiente enlace code-inn.ar, se podrá acceder a la aplicación llamada Rent Me Up. Consiste en la administración de reservas de diferentes tipos de hospedajes (cabañas, mansiones, departamentos, bed and breakfast), en la cual es posible registrarse e iniciar sesión como usuario y como administrador, el mismo tiene el control total sobre la información allí brindada.



Sprint 1

Planificación y ejecución de los tests #12 · created 1 month ago by Santiago Vallazza · Sprint 1 · Obligatoria · Testing / QA	CLOSED  0 updated 1 month ago
Testear la API #13 · created 1 month ago by Santiago Vallazza · Sprint 1 · Obligatoria · Testing / QA	CLOSED  0 updated 1 month ago

Sprint 2

Testeo Automatizado (JEST) #42 · created 1 month ago by Santiago Vallazza · Sprint 2 · Testing / QA	CLOSED  0 updated 2 weeks ago
Testeo Automatizado (Selenium) #43 · created 1 month ago by Santiago Vallazza · Sprint 2 · Electiva · Testing / QA	CLOSED  0 updated 1 week ago
Testeo Automatizado (Postman) #44 · created 1 month ago by Santiago Vallazza · Sprint 2 · Obligatoria · Testing / QA	CLOSED  0 updated 4 weeks ago
Testeo Exploratorio - Manual - Estático #45 · created 1 month ago by Santiago Vallazza · Sprint 2 · Obligatoria · Testing / QA	CLOSED  0 updated 4 weeks ago

Sprint 3

Implementar tests manuales #68 · created 3 weeks ago by Santiago Vallazza · Sprint 3 · Obligatoria · Testing / QA	CLOSED  0 updated 1 week ago
Implementar tests automatizados #69 · created 3 weeks ago by Santiago Vallazza · Sprint 3 · Obligatoria · Testing / QA	CLOSED  0 updated 2 days ago

Sprint 4



Implementar testing automatizado #76 · created 1 week ago by Santiago Vallazza · Sprint 4 · Obligatoria · Testing / QA	CLOSED 0 updated just now
Implementar testing manual #77 · created 1 week ago by Santiago Vallazza · Sprint 4 · Obligatoria · Testing / QA	CLOSED 0 updated just now
Confeccionar el reporte final de testing #78 · created 1 week ago by Santiago Vallazza · Sprint 4 · Obligatoria · Testing / QA	CLOSED 0 updated just now

Alcance

Dentro del Alcance

Para probar las funcionalidades del sistema desde el testing en el backend y el frontend de la aplicación, se trabajó con:

Postman

Postman nos permitió testear las APIs para, así, garantizar el correcto funcionamiento de manera externa de la aplicación y, a su vez, crear una estructura de información para el uso del equipo, con el fin de unificar los datos y su correcto despliegue en la aplicación, teniendo la siguiente colección:



RUN SUMMARY			
▼ Rent-me-up			
▼ Users			
POST POST - new admin			1
POST POST - admin already exists			
POST POST - admin login			
POST POST - new client			
POST POST - client login			
▼ Countries			
POST POST - admin			
POST POST - admin			
POST POST - client			
GET GET			
PUT PUT - admin			
PUT PUT - client			
DEL DELETE - admin			
DEL DELETE - client			
▼ Cities			
POST POST - admin			
POST POST - admin			
▶ POST POST - new admin			3 0
▶ POST POST - admin already exists			2 0
▶ POST POST - admin login			1 0
▶ POST POST - new client			3 0
▶ POST POST - client login			1 0
▶ POST POST - admin			1 0
▶ POST POST - admin			1 0
▶ POST POST - client			2 0
▶ GET GET			3 0
▶ PUT PUT - admin			1 0
▶ PUT PUT - client			2 0
▶ DELETE DELETE - admin			2 0
▶ DELETE DELETE - client			2 0
▶ POST POST - admin			1 0



▼ Cities	▶ DELETE DELETE - client	2 0	1
POST POST - admin	▶ POST POST - admin	1 0	
POST POST - admin	▶ POST POST - admin	1 0	
POST POST - client	▶ POST POST - admin	1 0	
GET GET	▶ POST POST - admin	1 0	
PUT PUT - admin	▶ POST POST - client	2 0	
PUT PUT - client	▶ GET GET	3 0	
DEL DELETE - admin	▶ PUT PUT - admin	2 0	
DEL DELETE - client	▶ PUT PUT - client	2 0	
▼ Categories	▶ DELETE DELETE - admin	2 0	
POST POST - admin	▶ DELETE DELETE - client	2 0	
POST POST - admin	▶ POST POST - admin	1 0	
POST POST - admin	▶ POST POST - admin	1 0	
POST POST - admin	▶ POST POST - client	2 0	
POST POST - client	▶ GET GET	2 0	
GET GET			
PUT PUT - admin			
▼ Categories	▶ GET GET	2 0	
POST POST - admin	▶ GET GET by id	2 0	
POST POST - admin	▶ GET GET by city	2 0	
POST POST - admin	▶ GET GET by category	2 0	
POST POST - admin	▶ GET GET by city and dates	1 0	
POST POST - client	▶ PUT PUT - admin	2 0	
GET GET	▶ PUT PUT - client	2 0	
PUT PUT - admin	▶ DELETE DELETE - admin	2 0	
PUT PUT - client	▶ DELETE DELETE - client	2 0	
DEL DELETE - admin	▶ POST POST - admin	1 0	
DEL DELETE - client	▶ POST POST - admin	1 0	
▼ Products	▶ POST POST - admin	1 0	
POST POST - admin	▶ POST POST - client	2 0	
POST POST - admin			
POST POST - client			
GET GET			



▼ Products	▶ POST POST - client	2 0	
POST POST - admin	▶ GET GET	2 0	
POST POST - admin	▶ GET GET by id	2 0	
POST POST - client	▶ PUT PUT - admin	2 0	
GET GET	▶ PUT PUT - client	2 0	
GET GET by id	▶ DELETE DELETE - admin	2 0	
GET GET by city	▶ DELETE DELETE - client	2 0	
GET GET by category	▶ POST POST - admin	1 0	
GET GET by city and dates	▶ POST POST - admin	1 0	
PUT PUT - admin	▶ POST POST - client	2 0	
PUT PUT - client	▶ GET GET	2 0	
DEL DELETE - admin	▶ GET GET by id	2 0	
DEL DELETE - client	▶ PUT PUT - admin	2 0	
▼ Characteristics	▶ PUT PUT - client	2 0	
POST POST - admin			
POST POST - admin			
▼ Characteristics	▶ DELETE DELETE - admin	2 0	
POST POST - admin	▶ DELETE DELETE - client	2 0	
POST POST - admin	▶ POST POST - admin	1 0	
POST POST - admin	▶ POST POST - client	2 0	
POST POST - admin	▶ GET GET	2 0	
POST POST - client	▶ PUT PUT - admin	2 0	
GET GET	▶ PUT PUT - client	2 0	
GET GET by id	▶ DELETE DELETE - admin	2 0	
PUT PUT - admin	▶ DELETE DELETE - client	2 0	
PUT PUT - client	▶ POST POST - admin	1 0	
DEL DELETE - admin	▶ POST POST - admin	1 0	
DEL DELETE - client	▶ POST POST - client	2 0	
▼ CharacteristicsXProducts	▶ GET GET	2 0	
POST POST - admin	▶ GET GET by id	2 0	
POST POST - admin			
POST POST - client			



▼ CharacteristicsXProducts	▶ PUT PUT - admin	2 0	
POST POST - admin	▶ PUT PUT - client	2 0	
POST POST - admin	▶ DELETE DELETE - admin	2 0	
POST POST - client	▶ DELETE DELETE - client	2 0	
GET GET	▶ POST POST - admin	1 0	
GET GET by id	▶ POST POST - admin	1 0	
PUT PUT - admin	▶ POST POST - admin	1 0	
PUT PUT - client	▶ POST POST - admin	1 0	
DEL DELETE - admin	▶ POST POST - client	2 0	
DEL DELETE - client	▶ GET GET	2 0	
▼ Images	▶ PUT PUT - admin	2 0	
POST POST - admin	▶ PUT PUT - client	2 0	
POST POST - client	▶ DELETE DELETE - admin	2 0	
GET GET	▶ DELETE DELETE - client	2 0	
PUT PUT - admin			
PUT PUT - client			



<ul style="list-style-type: none"> Images <ul style="list-style-type: none"> POST POST - admin POST POST - client GET GET PUT PUT - admin PUT PUT - client DEL DELETE - admin DEL DELETE - client PolicyTypes <ul style="list-style-type: none"> POST POST - admin POST POST - admin POST POST - client GET GET GET GET by id PUT PUT - admin PUT PUT - client 	<ul style="list-style-type: none"> POST POST - admin POST POST - client GET GET PUT PUT - admin PUT PUT - client DELETE DELETE - admin DELETE DELETE - client POST POST - admin POST POST - client GET GET - admin GET GET by product id - admin GET GET by user id - admin PUT PUT - admin PUT PUT - client 	<ul style="list-style-type: none"> 1 0 2 0 2 0 2 0 2 0 2 0 2 0 1 0 1 0 2 0 2 0 2 0 2 0 2 0
<ul style="list-style-type: none"> PUT PUT - client DEL DELETE - admin DEL DELETE - client Policies <ul style="list-style-type: none"> POST POST - admin POST POST - admin POST POST - admin POST POST - admin POST POST - client GET GET PUT PUT - admin PUT PUT - client DEL DELETE - admin DEL DELETE - client PoliciesXProducts Bookings 	<ul style="list-style-type: none"> GET GET PUT PUT - admin PUT PUT - client DELETE DELETE - admin DELETE DELETE - client POST POST - admin POST POST - client GET GET - admin GET GET by product id - admin GET GET by user id - admin PUT PUT - admin PUT PUT - client DELETE DELETE - admin DELETE DELETE - client 	<ul style="list-style-type: none"> 2 0 2 0 2 0 2 0 2 0 1 0 1 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0

Testing Automatizado – frontend

Se propuso utilizar Selenium para el test de automatización de frontend, ya que este mismo permite llevar a cabo la tarea de automatización de diferentes funcionalidades desde el frontend.



Para cubrir las principales funcionalidades de la aplicación, hubo que tener en cuenta los siguientes test cases:

- Búsqueda por apartamentos
- Búsqueda por cabaña
- Búsqueda por casas
- Búsqueda por ciudad
- Búsqueda por mansiones
- Carga de Recomendaciones
- Verificación Políticas de productos
- Verificación de botón "Ver más"
- Verificación de calendario en detalles de producto
- Verificación de calendario en reserva
- Verificación de características en detalles de producto
- Verificación de detalles de completar datos en la reserva
- Verificación de detalles de la reserva
- Verificación de horario de llegada de la reserva
- Verificación de mensaje de logueo para iniciar una reserva
- login
- verificación de carrusel de imágenes

Testing Manual

A través del testing manual, primero se redactaron casos de prueba basados en los requerimientos establecidos en cada lectura de sprint, que abarcaron gran parte del testeo de la interfaz gráfica y funcionalidades de la página. La ejecución de los casos de prueba fue documentada en las planillas correspondientes a cada sprint. Por último, al finalizar cada sprint se realizaron las pruebas exploratorias basadas en las user stories y las issues de gitlab. Tanto por los casos de prueba como por el testeo exploratorio surgieron defectos que fueron documentados en la planilla de defectos.

Testing

Fuera de Alcance

Las issues/actividades que no fueron probadas/realizadas son:

- Selenium IDE sprint 1 y 2



Tipos de Pruebas Ejecutadas

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Prueba Estática	SI	SI	SI	SI
Prueba Exploratoria	SI	SI	NO	SI
Prueba de Sistema (Selenium IDE, Selenium Web Driver)	NO	NO	SI	SI
Prueba de Humo	SI	SI	SI	SI
Prueba de Regresión	SI	SI	SI	SI
Prueba de Componente / Unidad (JEST)	NO	NO	NO	NO
Prueba de Integración (Postman)	SI	SI	SI	SI

Enfoque de la Prueba

El enfoque del testing del proyecto se basó en cubrir test funcionales para probar el funcionamiento del sistema, lo realizamos a través de test manuales y exploratorios. Para testear el front utilizamos Selenium y para testear el back utilizamos Postman.

Link Planilla de Casos de Prueba: [📄 Testing](#)

Link Planilla de Defectos: [📄 Testing](#)



Exit Criteria

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

- Tener el 100 % de los casos de prueba resueltos
- Tener el 90% de los defectos solucionados
- Tener el 100% de la ejecución de los test de Postman
- Tener el 80% de la ejecución de los test de Selenium

Resumen de Resultados

Diseño de Pruebas

	Test Manuales	Test Automáticos	Test de Integración (Postman)	Test Total
LogUser	1	5	5	11
Header	11	0	0	11
Body	1	0	0	1
Calendar	1	2	0	3
Footer	4	0	0	4
Home	2	5	0	7
Booking	5	5	0	10
Responsive	4	0	0	4
City	1	1	8	10
Product	6	3	12	21
Filtrado	2	5	10	17
Admin	3		45	
User	4	5	30	39

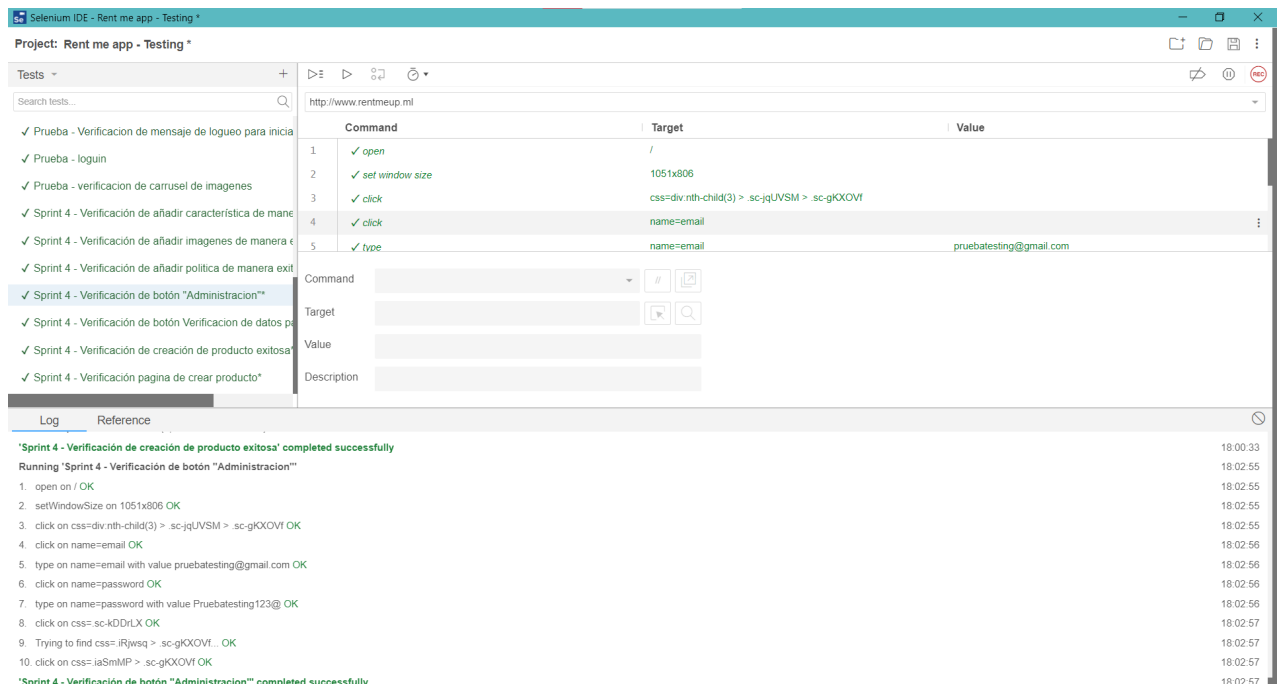
Ejecución de Pruebas

Ejecución Manual

Tabla con los resultados de las pruebas de Regresión ejecutado en el Sprint 4:

	Test Pasado	Test Fallados	Test no ejecutados	Test Total
Log User	1	0	0	1
Booking	3	0	0	3

Ejecución Automática



The screenshot shows the Selenium IDE interface with the project 'Rent me app - Testing *'. The test suite 'Sprint 4 - Verificación de creación de producto exitosa' is selected. The test steps are listed in the 'Tests' panel, and the 'Log' panel shows the execution results. The test suite completed successfully at 18:02:57.

Command	Target	Value
open	/	
set window size	1051x806	
click	css=div:nth-child(3) > .sc-jqUVSM > .sc-gKXOVt	
click	name=email	
type	name=email	pruebatesting@gmail.com

Log:

```

'Sprint 4 - Verificación de creación de producto exitosa' completed successfully
Running 'Sprint 4 - Verificación de botón "Administración"'
1. open on / OK
2. setWindowSize on 1051x806 OK
3. click on css=div:nth-child(3) > .sc-jqUVSM > .sc-gKXOVt OK
4. click on name=email OK
5. type on name=email with value pruebatesting@gmail.com OK
6. click on name=password OK
7. type on name=password with value Pruebatesting123@ OK
8. click on css=kDDrLX OK
9. Trying to find css=iRjwsq > .sc-gKXOVt... OK
10. click on css=jaSmMP > .sc-gKXOVt OK
'Sprint 4 - Verificación de botón "Administración"' completed successfully
  
```

Reporte de Defectos

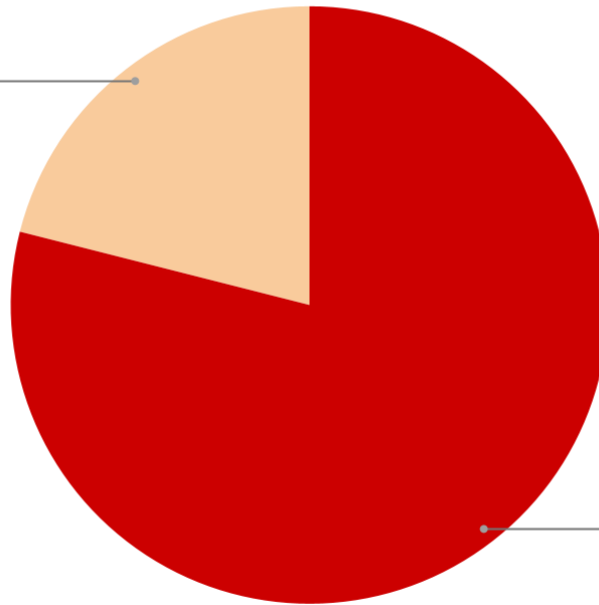
Todos los defectos

La siguiente sección muestra información con respecto al número total de 15 defectos que se han presentado durante la duración de la fase de prueba.



Defectos por prioridad

Media
21,1%

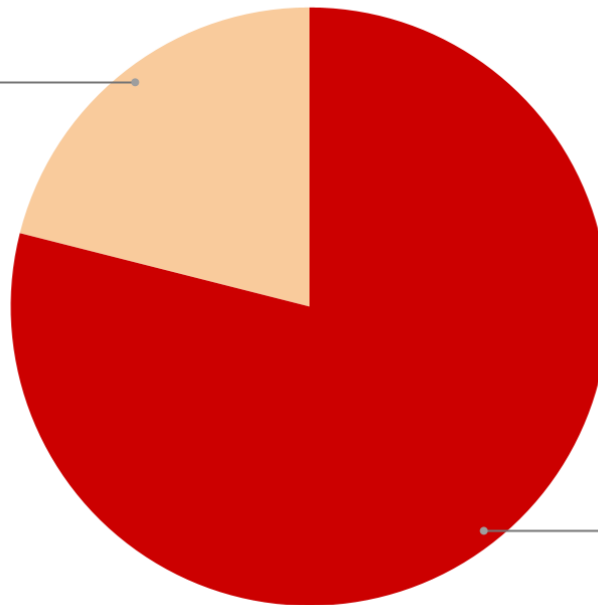


Alta
78,9%



Defectos por severidad

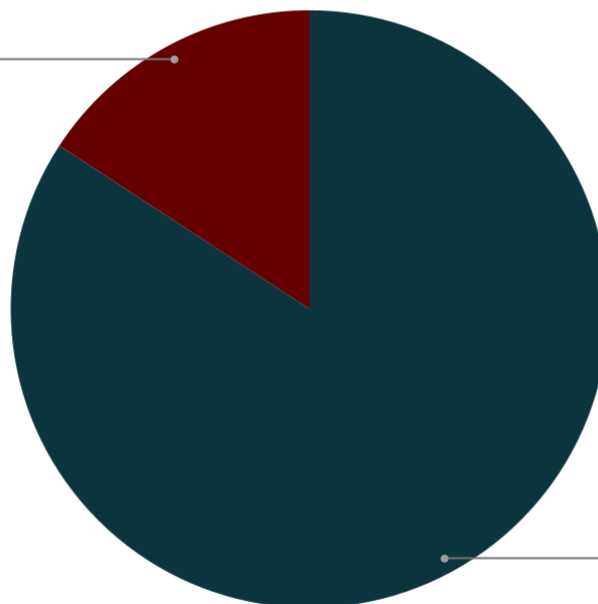
Media
21,1%



Alta
78,9%

Defectos por estado

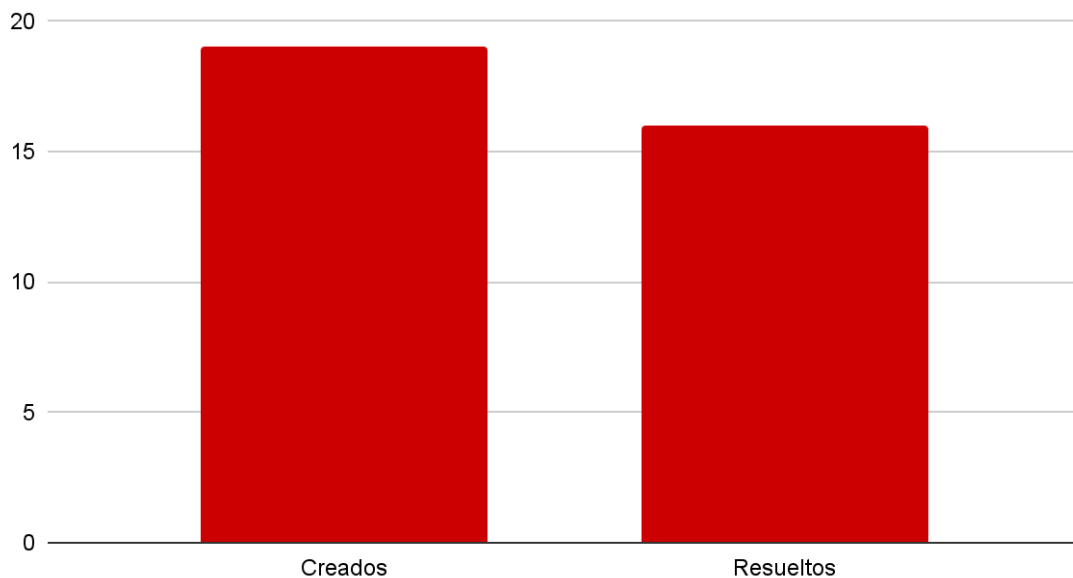
Pendientes
15,8%



Resueltos
84,2%



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.

Lecciones Aprendidas / Conclusión

El proceso de testing nos permitió:

Mejorar en cada una de las áreas de trabajo, ya que pudimos aprender de nuestros errores para evitar futuras equivocaciones.

También, nos ayudó a gestionar mejor nuestro tiempo, es decir, pudimos encontrar defectos en etapas tempranas del proyecto e ir resolviendo nuestro código con más facilidad y rapidez. Esto lo pudimos consolidar gracias al principio 3 “Pruebas Tempranas”.

Por otro lado, aprendimos que un producto de calidad no necesariamente es un producto “perfecto” sin fallas, debido a que lo fundamental es darle importancia a la satisfacción del cliente en cuanto a nuestro producto y no en los defectos mínimos que no cambian a la opinión del cliente, sino que retrasa la resolución del producto. Gracias al principio 7 “La falacia de ausencia de errores” pudimos llevarnos esa gran enseñanza.

Por último, nos facilitó la comunicación entre diversas áreas de trabajo, generando así un ambiente respetuoso y ameno para poder lograr una aplicación de calidad.