

Reporte Final de Testing



Sentite como en tu hogar

Integrantes:

- Matías Aplanalp
- Karen Cofone
- Laura Holguin
- Melina Valdiviezo
- Ailen García
- Victoria Hernandez

Contenido

Introducción

Resumen de las actividades de prueba

Alcance

Tipos de Pruebas Ejecutadas

Enfoque de la Prueba

Documentación

Pruebas diseñadas y ejecutadas por funcionalidades generales

Resumen de Resultados

Ejecución Automática

Reporte de Defectos

Todos los defectos

Defecto por Severidad

Exit Criteria

Lecciones Aprendidas y Conclusión

Introducción

Este documento evidencia que se cumplió el criterio de salida para el proceso de Testing y, por lo tanto, se concluye la fase de pruebas y se puede cerrar para la aplicación Digital Booking. La tarea principal del departamento de Testing ha sido garantizar que el software cumpla con los requisitos funcionales y no funcionales establecidos, así como detectar y reportar cualquier error o anomalía en el proceso de desarrollo.

Durante el proceso de prueba, se han realizado diversas pruebas de funcionalidad, compatibilidad, usabilidad, seguridad y rendimiento en distintos entornos y plataformas, como Postman y Selenium IDE. Además, se han evaluado aspectos como la interfaz de usuario, la navegación, la velocidad de carga, la gestión de datos y la integración con sistemas externos, entre otros.

El objetivo de este informe final de prueba es presentar los resultados obtenidos en el proceso de prueba y destacar las fortalezas y debilidades de la aplicación en términos de calidad y rendimiento.

En resumen, este informe es una herramienta fundamental para garantizar la calidad de la aplicación de Digital Booking y proporcionar una visión detallada y objetiva sobre el funcionamiento del software.

Resumen de las actividades de prueba

Durante los 4 sprints se realizaron varios tipos de testeo a la aplicación que se encuentra alojada en el siguiente link:

<http://equipo1.digitalbooking.s3-website.us-east-2.amazonaws.com/>

Se realizaron mayoritariamente test manuales para verificar las funcionalidades y test visuales para garantizar la experiencia del usuario, test automatizados en Postman para verificar el correcto funcionamiento de la API con peticiones de tipo GET/POST/DELETE, testeando que estas funcionaran de manera esperada y que los datos traídos sean los correctos. Se utilizó Selenium IDE para hacer más pruebas automatizadas siguiendo el camino feliz de todas las funcionalidades.

Alcance

Desde QA nos enfocamos en testear las principales funcionalidades de la página a medida que se iban incorporando sprint a sprint. A medida que se avanzaba y se iban agregando, no solo se testeaban las nuevas funciones, sino también las ya existentes para confirmar que la integración de las actualizaciones no causarían defectos a las existentes ni interfirieran con ellas.

Testeamos durante los 4 Sprint:

- Responsivo (Desktop, Tablet, IpadMini y Mobile M).
- Peticiones a la API con Postman.
- Registro de usuario, verificando por Postman, Selenium IDE y manualmente por base de datos.
- Login de usuario.
- Validaciones en los formularios.
- Creación de Reservas, verificando por Postman, Selenium IDE y manualmente por base de datos.

- Creación de productos desde el rol de administrador verificando por Postman y manualmente por base de datos.
- Calendarios de inicio, producto y reservas.
- Filtros por categoría, fecha, ciudad y combinados.
- Página de error 404.
- Verificación de que en las búsquedas solo se muestran los productos disponibles.
- Funcionamiento y redirección dentro de la página.
- Funcionamiento y flujo de la página en general.
- Carrusel automático de imágenes en móvil y Tablet.
- Visualización de productos aleatorios en inicio.
- Locación del producto en el mapa.

Tipos de Pruebas Ejecutadas

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Prueba de Regresión	x	x	✓	✓
Prueba de Humo	x	x	✓	✓
Prueba Estática	✓	✓	✓	✓
Prueba Exploratoria	✓	✓	✓	✓
Prueba de Sistema con Selenium IDE	x	✓	✓	✓
Prueba de integración con Postman	✓	✓	✓	✓

Enfoque de Prueba

Trabajamos en conjunto todos los departamentos, mantuvimos una comunicación constante y utilizamos varias herramientas de apoyo, pero la más efectiva fue el board de GitLab y una planilla de Excel en la que íbamos registrando los avances desde el defecto detectado hasta el cierre del mismo. El equipo de QA era el encargado de verificar que la tarea esté finalizada, funcione correctamente y cumpla con los requisitos solicitados. Para ello, se generaba un test en una planilla, se documentaba el resultado y si cumplía con lo solicitado, se pasaba su estado a cerrado.

Por cada funcionalidad y tarea se creaba un test positivo. Al comienzo de un nuevo sprint, volvíamos a testear las funcionalidades anteriores para verificar que sigan funcionando de manera correcta.

Todo el proceso se reflejaba en planillas de pruebas con descripciones de cómo repetir la acción y si está aprobado o defectuoso. El enfoque se puso en las funciones básicas como el registro de usuario, el login, el funcionamiento correcto de las API, el acceso a los productos, las reservas y los filtros.

En cada sprint se informó al equipo de todos los defectos encontrados y se agregaban a la planilla de tareas por hacer en GitLab y Excel para tenerlos más presentes y en constante consideración.

Documentación

La misma se encuentra en la rama de GitLab de testing.

<https://gitlab.ctd.academy/ctd/hispanos/proyecto-integrador-1/proyecto-integrador-0223/0522ft-c2/grupo-01.git>

Pruebas diseñadas y ejecutadas por funcionalidades generales

	Test Manuales	Test Selenium	Test Postman	TOTAL
REGISTRO	9	1	0	10
LOGIN	11	1	1	13
LISTADO	2	2	4	8
HEADER	3	1	0	4
BUSCADOR	12	2	0	14
CATEGORÍAS	5	1	4	10
PRODUCTOS	9	1	9	19
IMÁGENES	3	2	2	7
RESPONSIVE	18	7	0	25
RESERVAS	8	2	0	10
CREACION PRODUCTOS	10	5	0	15
MIS RESERVAS	4	4	4	12

Postman- 100% de los test aprobados

DigitalBooking S4 - Run results

Run on 29 Mar, 2023 22:50:06 · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	35s 522ms	34	1953 ms

All Tests Passed (28) Failed (6) Skipped (0) [View Summary](#)

- PASS status code is 200
- PASS Response time is less than 500ms

GET CharacteristicsList
http://localhost:8080/characteristics/list 200 OK 925 ms 884 B

- PASS Status code is 200
- PASS Response time is less than 500ms

GET CitiesFind
http://localhost:8080/cities/list 200 OK 1219 ms 1.063 KB

- PASS Status code is 200
- PASS Response time is less than 500ms

GET ImagenesList
http://localhost:8080/images/findProductsWithImages/5 200 OK 1847 ms 8.893 KB

- PASS Status code is 200
- PASS Response time is less than 500ms

Selenium 100% de los test aprobados

Project: TC DB #2 EQUIPO 1*

Search tests...

Command	Target	Value
type	css=imgAdd > input	https://imgcy.trivago.com/c_fill,d_dummy.jpeg,f_auto,h_190,q_auto,w_240/itemimages/10/40/1040224_v2.jpg
click	css=fa-square-plus > path	
mouse over	css=fa-square-plus > path	
mouse out	css=fa-square-plus > path	
click	css=addButton	
click	css=SuccessfulProduct > div	
assert text	xpath=//p[contains(text),'Su alojamiento se ha creado con éxito']	Su alojamiento se ha creado con éxito

Log

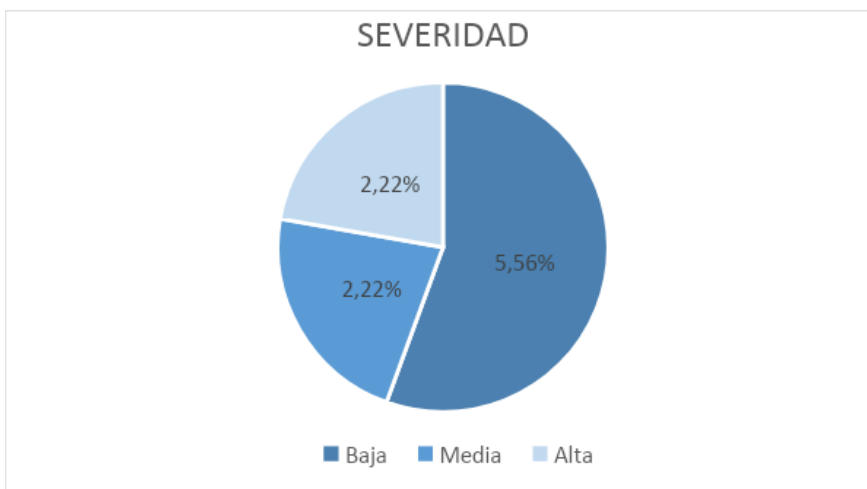
- 36. mouseOver on css=fa-square-plus > path OK 16:31:02
- 37. click on css=addButton OK 16:31:02
- 38. click on css=SuccessfulProduct > div OK 16:31:05
- 39. assertText on xpath=//p[contains(text),'Su alojamiento se ha creado con éxito'] with value Su alojamiento se ha creado con éxito OK 16:31:08

TC 29 Formulario Producto - Creacion de Producto exitosa *completed successfully 16:31:09

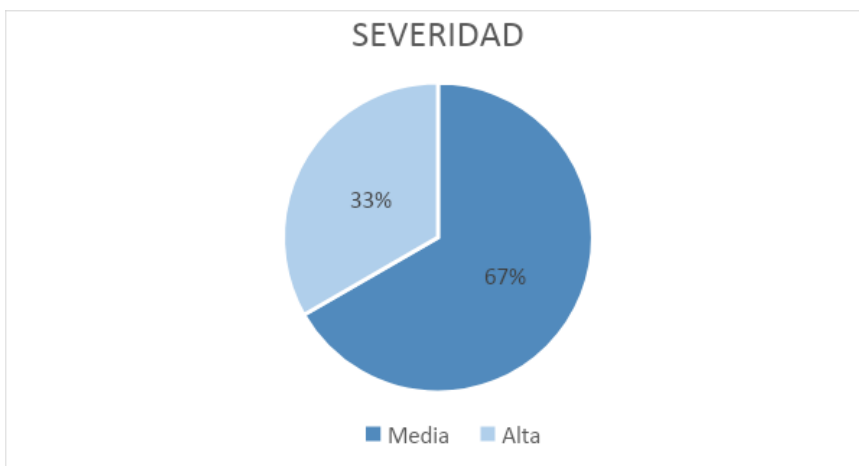
Reporte de Defectos

Se reportaron:

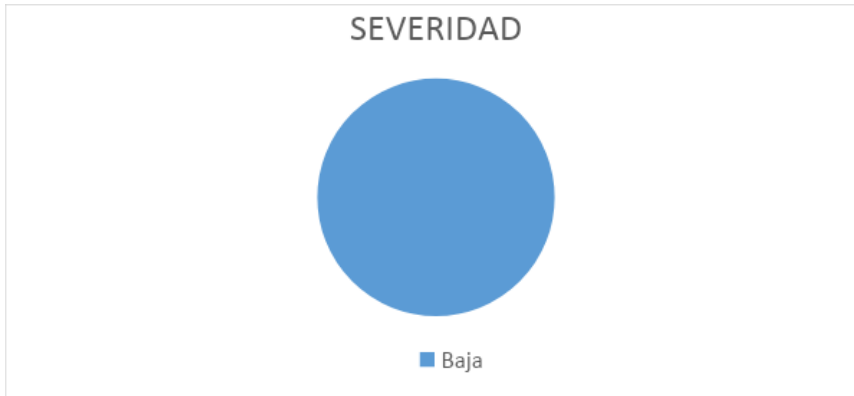
DESKTOP VISUAL



RESPONSIVE MOBILE M



RESPONSIVE IPADMINI/TABLET



Todos los defectos se trataron con prioridad Media y actualmente el 100% de ellos están en **ESTADO: CERRADO**

Exit Criteria

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

- No debe haber defectos en estado abierto de severidad crítica y/o bloqueante
- Test de Postman deben pasar en su 100% para verificar el correcto funcionamiento de la API.
- Test manuales por cada historia de usuario de cada sprint.
- Automatización en Selenium IDE con registro completado exitosamente.

Conclusiones

Hemos aprendido la importancia de la planificación, es esencial para llevar a cabo un proceso de testing eficiente y efectivo. Se deben definir los casos de prueba, escenarios y datos de prueba adecuados, así como establecer un cronograma y prioridades. Dicha planificación también debe incluir la coordinación con el equipo conformado por todos los departamentos.

La necesidad de la documentación, entendimos que es fundamental para llevar un registro de los resultados de las pruebas y errores encontrados. Además, la misma permite al equipo de desarrollo tomar medidas adecuadas y resolver los problemas de manera eficiente.

La comunicación es clave para asegurar la calidad del software. Los errores deben comunicarse claramente al equipo de desarrollo, junto con información detallada sobre cómo se produjo el error y cómo se puede replicar.

La importancia de las pruebas manuales y automatizadas. Manuales permiten encontrar errores que no se pueden detectar con herramientas automatizadas, mientras que las pruebas automatizadas son útiles para detectar errores repetitivos y nos permite ahorrar tiempo.

La retroalimentación es importante para mejorar el proceso de testing y para asegurar que los errores no se repitan. El equipo de desarrollo debe recibir retroalimentación constante sobre los errores encontrados, el rendimiento de la aplicación y las áreas que necesitan mejoras.

En conclusión consideramos, crucial para asegurar la calidad del software y la satisfacción del usuario, la planificación, documentación, comunicación, pruebas manuales y automatizadas, y la retroalimentación elementos esenciales para llevar a cabo un proceso de testing efectivo.

Habiendo posibilidades de mejora, la totalidad de las pruebas realizadas, expone, basándonos en las especificaciones de la planificación, que cumplimos con una tasa de más de un 95% de las mismas solicitadas, lo que nos da la conformidad de que DBooking está funcionando correctamente.

Muchas gracias.

Departamento de Testing/QA