Certified Tech Developer
The Ultimate Degree

Informe final de Testing

Digital Booking

Albarracín Marisabel

Luque Juan

Montero Brenda

Mórtigo Camilo

Ortiz Maldonado Natalia

Romero Nicolas

Índice

Introducción

En este documento se presentan los resultados de los diferentes tipos de pruebas realizadas sobre el software desarrollado Digital Booking durante los 4 sprints del proyecto. Se realizó testing de forma manual y también automatizada con el fin de agilizar el proceso de testeo y poder alcanzar un alto porcentaje de cobertura de las distintas funcionalidades del software. Se deja constancia en el presente documento que se cumple con el criterio para finalizar el testing: no tener defectos en estado abierto de severidad crítica y/o bloqueante.

Alcance

Se testearon las siguientes funcionalidades solicitadas en las historias de usuario:

- Verificar que el sitio sea responsive
- Interacción entre links de slogan y logotipo
- Inicio de sesión/ login
- Registro de un nuevo usuario
- Búsqueda de departamentos por ciudad
- Búsqueda de departamentos por fecha
- Visualización de recomendaciones como usuario anónimo
- Búsqueda de departamentos por categoría
- Visualización de detalle de cada departamento
- Visualización e interacción con carrusel de imágenes de cada departamento
- Verificación de disponibilidad de un departamento
- Realizar la reserva de un departamento
- Cerrar sesión
- Gestionar un producto con rol administrador
- Apis (GET/ POST/ DELETE/ PUT) según lo solicitado en cada sprint

Proceso de testeo

Durante el desarrollo del proyecto hemos pasado por distintas etapas del testing, incrementando la cantidad de pruebas y calidad del testeo a medida que se desarrollaba cada sprint.

Durante el sprint 1 se crearon los casos de prueba positivos y negativos de cada historia de usuario solicitada. Al realizar la ejecución de los mismos, hemos plasmado su resultado en la planilla de casos de prueba. En aquellos casos donde se encontraron defectos, los mismos fueron reportados dentro de la planilla de defectos para monitorear el estado de cada uno. También se realizó test exploratorio con el fin de abarcar aquellos requerimientos no funcionales y funcionales que fueron solicitados en el sprint 1 y no fueron alcanzados por los casos de prueba.

En el sprint 2 se actualizaron los casos de prueba según lo solicitado en este nuevo sprint y se actualizaron también los resultados de las pruebas una vez corregidos los

defectos reportados en el sprint 1. Se clasificó cada caso de prueba en test de humo y test de regresión dentro de la planilla. Luego de la revisión de la planilla de defectos junto al equipo de desarrollo, se realizaron las correcciones necesarias para resolver los defectos cuya criticidad era media / alta y generaba alguna traba para continuar con otra funcionalidad requerida en el sprint 2.

En este sprint agregamos testeo automatizado con Selenium. Se realizó la automatización de cada caso de prueba escrito en la planilla de casos de prueba, abarcando los requerimientos del sprint 1 y 2. También desarrollamos scripts de prueba en Postman para el testo de las diferentes APIs creadas. Utilizando el test runner ejecutamos dichos scripts y generamos un informe en formato JSON con los resultados obtenidos. También se realizó test con JEST en el código del Frontend para verificar que el componente de registro y login se esté renderizando de forma correcta.

En el desarrollo del sprint 3 se realizó test exploratorio sobre la página de reserva y confirmación de reserva. Sobre estas mismas páginas se confeccionaron casos de prueba positivos y negativos y se plasmó el resultado en la planilla correspondiente.

Los defectos encontrados tanto en el test exploratorio como en la ejecución de casos de prueba se evidenciaron en la planilla de defectos.

En este sprint se utilizó Postman para el testeo de las APIs solicitadas en reservas y usuarios, se generaron los scripts automatizados y se actualizó el informe al correr el test runner.

Finalmente, en el último sprint se agregaron los casos de prueba sobre la página de gestión de productos por parte del administrador. Se ejecutaron junto al test exploratorio de esta página y el resultado se plasmó en ambos documentos. Junto al equipo de desarrollo se revisó la planilla de defectos a fin de solucionar aquellos cuya severidad es crítica/ bloqueante para poder dar por finalizado el testeo de la aplicación y aprobar la salida a producción.

Tipos de test y tecnologías utilizadas

- Selenium: se crearon pruebas automatizadas de cada caso de prueba correspondiente a historias de usuario del sprint 1 y sprint 2.
- Jest: se crearon algunos test unitarios sobre los componentes Login y Register.
- Postman: se utilizó para el testeo de las distintas APIs utilizadas en el proyecto. Se desarrollaron scripts de prueba automatizados para corroborar la respuesta exitosa de cada petición GET/ POST/DELETE/ PUT.
- Test manuales: se crearon casos de prueba tanto positivos como negativos para cada una de las historias de usuario del proyecto. A raíz de la ejecución de cada caso de prueba se plasmó el resultado en la planilla y en caso de encontrar algún defecto se plasmó en la planilla de defectos. Se clasificaron en pruebas de humo y pruebas de regresión. Se crearon test suites.

- Test exploratorio: se realizó test exploratorio para abarcar aquellas funcionalidades que no fueron cubiertas por los casos de prueba. Se plantearon distintos escenarios para poder testear la aplicación y de esa manera encontrar aquellos defectos que podían pasar por alto con el resto de los test utilizados.

Cantidad de pruebas realizadas por funcionalidad/ historia de usuario tanto manuales como automatizadas

Funcionalidad	Test manuales (Casos de prueba/ test exploratorio)	Test automáticos (Selenium)	Test APIs en Postman	Total
Verificar que el sitio sea responsive	1	0	0	2
Interacción entre links de slogan y logotipo	2	2	0	4
Inicio de sesión/ login	3	3	0	6
Registro de un nuevo usuario	4	4	0	8
Búsqueda de departamentos por ciudad	2	2	1	5
Búsqueda de departamentos por fecha	1	1	1	3
Visualización de recomendaciones como usuario anónimo	1	0	0	1
Búsqueda de departamentos por categoría	1	1	1	3
Visualización de detalle de cada departamento	1	1	1	3

Visualización e interacción con carrusel de imágenes de cada departamento	2	1	1	4
Verificación de disponibilidad de un departamento	1	0	1	2
Realizar la reserva de un departamento	2	0	1	3
Cerrar sesión	1	1	0	2
Gestionar un producto con rol administrador	2	0	4	6

Cantidad de casos de prueba ejecutados, pasados y fallados

Casos de prueba	Ejecutados	Pasados	Fallados
32	32	32	1

Exit criteria

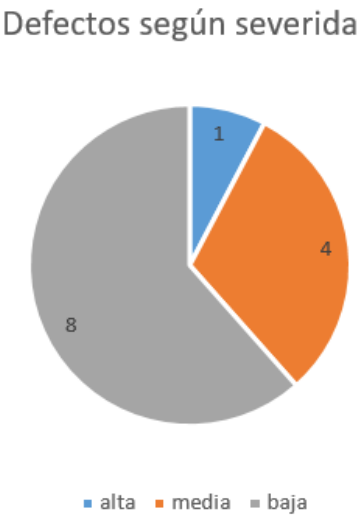
- No se presentar defectos en estado abierto de severidad crítica y/o bloqueante.
- Testing en Selenium: 100% de cobertura de sprint 1 y 2.
- Testing de APIs en Postman: 100% de cobertura con resultado exitoso.

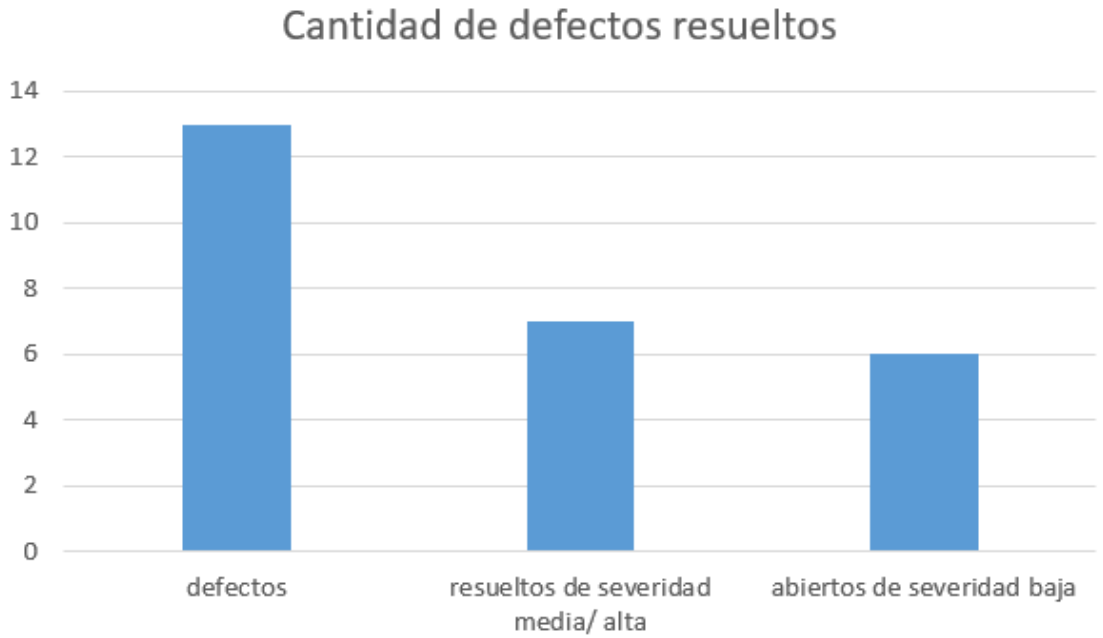
Cantidad de defectos resueltos/ abiertos

Los defectos de criticidad media y alta se encuentran todos resueltos. Quedan 6 defectos de severidad baja abiertos.

Defectos	Criticidad Baja	Criticidad Media	Criticidad Alta
13	8	4	1

Defectos Resueltos	Criticidad Baja	Criticidad Media	Criticidad Alta
7	2	4	1





Documentación

- Test selenium, dentro de una carpeta están las pruebas de humo y en otra las pruebas de regresión. Se incluye el proyecto de Selenium. También se dejó el resultado de cada test en issue #41 : https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/tree/Testing/Testing/Selenium
- Test Postman Informe: https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/blob/Testing/Testing/PI_BackEnd.postman_test_run.json
- Collection Postman APIs: https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/tree/Testing/Testing/Postman%20Collections

- Casos de prueba: https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/blob/Testing/Testing/Casos%20de%20prueba.xlsx
- Test Exploratorio: https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/blob/Testing/Testing/Testing%20exploratorio.docx
- Informe de defectos: https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/blob/Testing/Testing/Informe%20de%20defectos.xlsx
- Test Jest: dentro de la rama Frontend https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0521-pt-c2/grupo-10/-/tree/Frontend/frontend/src/__test__

Conclusión

Durante cada sprint hemos avanzado tanto en la redacción de casos de prueba, su mejora, creación de nuevos casos a partir del testeo global de la aplicación, la detección de defectos, se aprendió a utilizar tecnologías como Selenium, Postman y Jest para alcanzar un porcentaje de cobertura mayor al obtenido con los test manuales. Entendimos que cuanto más tipo de test tengamos podremos obtener un producto sólido y eficaz para el cliente. Aprendimos la importancia de cada ejecución, como impacta una función y posible defecto en otro módulo o función, encontrar un defecto a tiempo, al final de cada sprint, para poder corregirlo y no arrastrar un error que hacia el final sea demasiado tarde y conlleve mayor costo.

Realizar el testing de forma eficaz permite asegurar la calidad del producto entregado y la satisfacción del cliente.