Certified Tech Developer
The Ultimate Degree

DigitalHouse> Coding School

Reporte Final de Testing

DIGITAL BOOKING – GRUPO 6

Integrantes:
Jhon Parra
Abril Santiso
Sol Rocca
Santiago Garlot
Daniel Romero
Cristian Sánchez

Contenido

Introducción

Este documento es el Informe Final de Pruebas del sistema Digital Booking del grupo 6 de la camada 3. 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

Nuestro sistema consiste en una web app de Booking y reservas de alojamientos. En el siguiente link puedes acceder a la web en acción http://digitalbooking.xyz/. Para este sistema se realizaron pruebas unitarias usando JUnit y de integración empleando postman para el backend. Para el frontend se usaron test manuales tales como exploratorios, de sistema y de humo, para el test automatizado se usó Jest para validar el funcionamiento de algunos componentes.

Alcance

Dentro del Alcance

Desde el testing manual se probaron funcionalidades de frontend, backend, base de datos e infraestructura, se implementaron pruebas de regresión y humo. Las funcionalidades probadas fueron las siguientes:

- Frontend:
  - Verificación de renderización header
  - Verificación de renderización footer
  - Verificación login success
  - Verificación de login failed
  - Verificación de pantallas como login, registro, home y producto
  - Verificación de renderización de productos
  - Verificación funcionamiento bloque buscador
  - Verificación formularios de ingreso y registro de usuario
  - Verificación funcionamiento flecha back
  - Verificación funcionamiento galería de imágenes de producto
  - Verificación pantalla detalles del producto

    - Verificación renderización del mapa para el producto
    - Verificación renderización características del producto
    - Verificación cerrar sesión
    - Verificación creación de productos
    - Verificación de urls amigables para el usuario
    - Verificación funcionamiento para seleccionar hora de llegada
    - Verificación funcionamiento del calendario
    - Verificación funcionamiento reservas
    - Verificación filtro de reservas
    - Verificación de visualización en mobile, tablet y desktop
- Backend
    - Verificación método agregar para cada una de las entidades de la API
    - Verificación método listar para cada una de las entidades de la API
    - Verificación método actualizar para cada una de las entidades de la API
    - Verificación método eliminar para cada una de las entidades de la API
    - Verificación de relaciones
- Bases de datos:
    - Verificación de creación del schema
    - Verificación de la creación de las tablas
    - Verificación de inserción de los datos
    - Verificación de lectura de la base de datos
    - Verificación de actualización en la base de datos
    - Verificación de borrado de registros en la base de datos
- Infraestructura:
    - Verificación del funcionamiento de la instancia EC2
    - Verificación de la creación del RDS
    - Verificación del funcionamiento del S3

Las funcionales que se trataron con test automatizados (Jest, Postman y Selenium) fueron las siguientes:

- Jest:
    - Verificación renderización componentes funcionales
    - Verificación renderización textos y etiquetas html
    - Verificación consumo de API en productos y categorias
    - Verificación enrutamiento de los link
    - Verificación eventos click en botones
- Postman:
    - Verificación CRUD de la API
    - Verificación security y roles
    - Verificación integración del sistema
- Selenium:

- Verificación creación de usuario success
- Verificación creación de usuario failed
- Verificación login usuario creado
- Verificación bloque de búsqueda

Se usó el testing manual y automatizado como complemento entre ellas para probar la mayoría de las funcionalidades de nuestro sistema.

Fuera de Alcance

Dentro de las funcionalidades que no fueron probadas bajo ningún tipo de prueba se encuentran las siguientes:

- Selección fechas lejanas en los calendarios
- Funcionamiento de los mapas
- Funcionamiento en el envío de emails
- Selección y actualización de la página de favoritos

Tipos de Pruebas Ejecutadas

Durante el desarrollo del sistema se emplearon diferentes tipos de pruebas para testear cada una de las funcionalidades implementadas, en cada uno de los sprint se aplicaron las pruebas que mejor se ajustaban a los requerimientos del sistema.

	Sprint 1	**Sprint 2**	**Sprint 3**	**Sprint 4**
Prueba Estática		x	x	x
Prueba Exploratoria	x	x	x	x
Prueba de Sistema		x	x	x
Prueba de Humo	x	x	x	x
Prueba de Regresión	x	x	x	x
Prueba de Componente / Unidad	x	x	x	

Prueba de Integración (Postman)		x	x	x

Enfoque de la Prueba

En el desarrollo del testing para nuestro sistema, inicialmente se implementaron tests funcionales positivos para el primer sprint, desde el segundo sprint se aumentó la cantidad de tests negativos pero la mayoría sigue siendo del tipo positivo. Decidimos crear desde el inicio del proyecto dos ramas principales para frontend y backend respectivamente y a partir de estas, generar nuevas ramas para funcionalidades en concreto. Cuando se libera alguna funcionalidad de las ramas mencionadas anteriormente se implementa el testing manual y automatizado para verificar su funcionalidad. A la par se realizaba una prueba de humo para detectar defectos y reportarlos en caso de ser necesario.

Antes de finalizar el sprint se unen las ramas principales con sus subtareas y se realizaba la integración entre ramas antes de enviar todo a la rama main, durante este proceso se realizaban tareas de testing al verificar el funcionamiento de las funcionalidades implementadas y mediante pruebas de regresión identificar donde se debía realizar el ajuste en el código.

Las pruebas automatizadas se desarrollaron sobre las funcionalidades mencionadas en el alcance y se ejecutaban cada vez que se agregaba una funcionalidad a la rama principal de frontend o backend y se corría de nuevo antes de generar el build previo a la review del sprint.

Para el último sprint, se realizó la suite completa para pruebas de regresión para identificar defectos durante estas pruebas y corregirlos para la entrega final, también se detectaron aquellos defectos que no logramos ajustar por el tiempo y que no afectan al desempeño del sistema en general para ser corregidos en una próxima iteración.

Link Planilla de Casos de Prueba (Pestaña Casos de prueba):
Casos de prueba (Pestaña casos de prueba)

Link Planilla de Defectos (Pestaña Defectos):
Defectos (Pestaña defectos)

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.
- No se debe tener defectos en estado abierto de prioridad crítica.
- Los tests automatizados con Jest deben tener una cobertura mínima del 40% para todo el sistema.
- La ejecución de las pruebas de regresión deben tener mínimo un 90% de pass rate.

Resumen de Resultados

Diseño de Pruebas

En la siguiente tabla se muestra un desglose de las funcionalidades probadas con sus respectivos tipos de pruebas implementadas para su verificación.

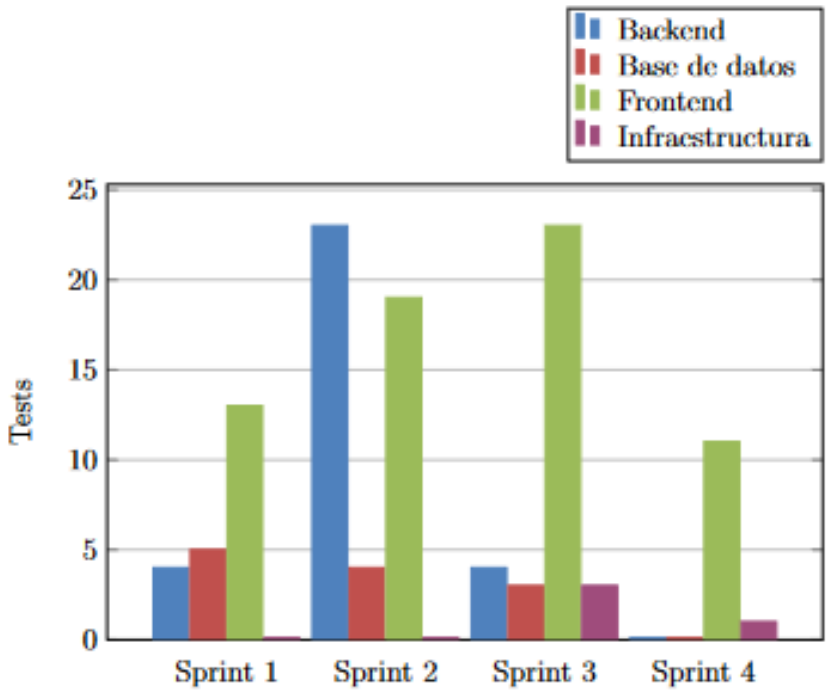
	Test Manuales	***Test Automáticos***	***Test de Integración (Postman)***	***Test Total***
Login de Usuario	*3*	*6*	*0*	*9*
Crear Cuenta	*1*	*5*	*1*	*7*
Logo a Home	*1*	*3*	*0*	*4*
Seleccionar Producto	*3*	*5*	*0*	*8*
Seleccionar fecha (Check in - Check out)	*6*	*2*	*0*	*8*
Búsqueda de producto y fecha	*4*	*3*	*0*	*7*
Filtrado por categoría	*3*	*3*	*0*	*6*

Borrar filtro	*2*	*3*	*0*	*5*
Continuar leyendo	*1*	*2*	*0*	*3*
Producto ver más	*2*	*3*	*0*	*5*
Selección favoritos (♥)	*1*	*3*	*0*	*4*
Compartir redes	*2*	*2*	*0*	*4*
Realizar Reserva	*4*	*2*	*0*	*6*
Administrar Producto	*2*	*1*	*0*	*3*
Bases de datos	*12*	*0*	*0*	*12*
CRUD en APIS	*4*	*0*	*4*	*8*
Despliegue de la infraestructura	*5*	*0*	*0*	*5*

Ejecución de Pruebas

Ejecución Manual

En el siguiente gráfico se detallan las pruebas de humo y regresión usadas para cada uno de los sprint y según el tipo (Backend, Base de datos, Frontend e Infraestructura)



Para el sprint 4 se implementaron pruebas de regresión para verificar que las nuevas funcionalidades integradas llegasen a afectar las funcionalidades que ya se encontraban implementadas. En la siguiente tabla se resumen las pruebas realizadas:

	Test Pasado	***Test Fallados***	***Test no ejecutados***	***Test Total***
Creación del producto	*1*	*0*	*0*	*1*
Submit formulario creación	*1*	*0*	*0*	*1*
Pantalla mis reservas	*4*	*0*	*0*	*4*
Filtrar reservas	*1*	*0*	*0*	*1*
Verificación infraestructura	*1*	*0*	*0*	*1*

Ejecución Automática

En la siguiente gráfica se presentan el resumen de las pruebas realizadas con Jest para los componentes implementados en el Frontend con React, para este tenemos una cobertura mínima del 40%, en el caso de componentes sencillos se logró una cobertura del 100% pero en los componentes de integración que tienen la renderización de varios componentes y una lógica más compleja, se testeó que su renderización fuera correcta y eventos de link para redireccionamiento a otras páginas.

All files

40.16% Statements 243/605 **30.30%** Branches 70/231 **31.98%** Functions 71/222 **40.30%** Lines 237/588

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	100%	2/2	100%	0/0	100%	1/1	100%	2/2
src/components/AboutUsScreen	75%	3/4	100%	0/0	50%	1/2	75%	3/4
src/components/ButtonClearFilter	100%	2/2	100%	0/0	100%	1/1	100%	2/2
src/components/ButtonFav	100%	2/2	50%	1/2	100%	1/1	100%	2/2
src/components/ButtonLogin	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/ButtonProductCreate	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/ButtonRegister	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/ButtonReservation	30%	3/10	0%	0/2	33.33%	1/3	30%	3/10
src/components/ButtonReservationConfirm	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/ButtonShare	46.66%	7/15	64.28%	9/14	50%	3/6	46.66%	7/15
src/components/Categories	50%	7/14	50%	1/2	14.28%	1/7	50%	6/12
src/components/Characteristics	29.72%	11/37	18.75%	6/32	100%	4/4	29.72%	11/37
src/components/Footer	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/GoToTop	40%	4/10	25%	1/4	20%	1/5	40%	4/10
src/components/Header	69.23%	9/13	71.42%	10/14	75%	3/4	69.23%	9/13
src/components/Home	38.88%	14/36	10%	1/10	11.76%	2/17	37.14%	13/35
src/components/ImageGallery	83.33%	5/6	0%	0/1	66.66%	2/3	83.33%	5/6
src/components/Layout	66.66%	14/21	100%	0/0	85.71%	6/7	66.66%	14/21
src/components/LoadingSkeleton	100%	6/6	60%	6/10	100%	2/2	100%	5/5
src/components/LoggedInMenu	63.63%	7/11	50%	1/2	50%	1/2	63.63%	7/11
src/components/Menu	80%	12/15	75%	12/16	100%	3/3	80%	12/15
src/components/Menu/bottom-menu	87.5%	7/8	50%	2/4	80%	4/5	87.5%	7/8
src/components/NavBar	100%	2/2	100%	0/0	100%	1/1	100%	2/2
src/components/OverlayComponent	71.42%	5/7	100%	0/0	33.33%	1/3	83.33%	5/6
src/components/ProductCreate	19.26%	21/109	0%	0/30	2.27%	1/44	19.81%	21/106
src/components/ProductDetails	16.66%	3/18	0%	0/2	16.66%	1/6	17.64%	3/17
src/components/ProductLabel	4.76%	1/21	0%	0/12	0%	0/6	5%	1/20
src/components/ProductPolicies	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/Products	68.42%	13/19	40%	4/10	71.42%	5/7	68.42%	13/19
src/components/RatingStars	100%	2/2	50%	5/10	100%	1/1	100%	2/2
src/components/ReservationForm	6.57%	5/76	0%	0/18	6.45%	2/31	6.66%	5/75
src/components/SearchBar	100%	2/2	100%	0/0	100%	1/1	100%	2/2
src/components/SearchCalendar	26.66%	16/60	15%	3/20	26.31%	5/19	26.78%	15/56
src/components/SearchLocation	60%	6/10	100%	0/0	40%	2/5	60%	6/10
src/components/SignInButton	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/SignUpButton	100%	1/1	100%	0/0	100%	1/1	100%	1/1
src/components/SocialFollow	100%	5/5	100%	2/2	100%	1/1	100%	5/5
src/components/UserPanel	46.87%	15/32	20%	2/10	14.28%	2/14	43.33%	13/30
src/utils	100%	22/22	100%	4/4	100%	2/2	100%	22/22

Tests postman

En el siguiente Link se encuentran los JSON usados para las validaciones del backend empleando postman.

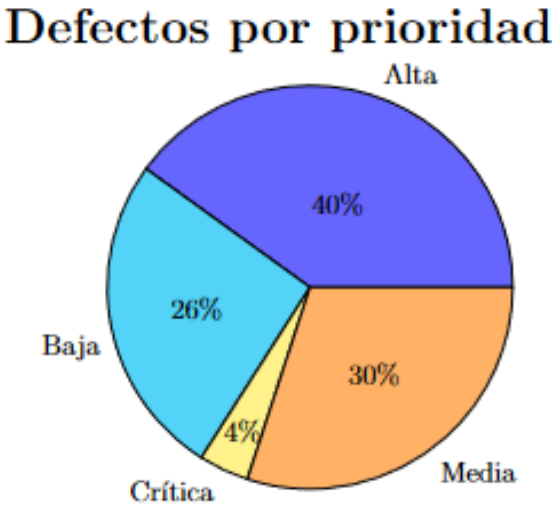
Reporte de Defectos

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.

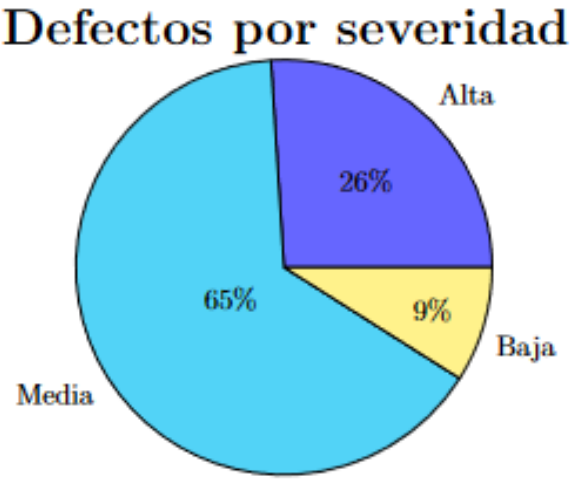
Defectos por prioridad

Para los defectos según la prioridad del cliente definimos diferentes categorías, tales como prioridad Crítica, Alta, Media y Baja. Teniendo en cuenta esta categorización clasificamos cada uno de los defectos de nuestro sistema y los resultados son los que se muestran en el siguiente gráfico.



Defectos por Severidad

Para los defectos según su severidad se tiene la siguiente clasificación: Alta, Media y Baja. Bajo esta categorización se clasificaron los defectos generados durante el desarrollo de los 4 sprints.



Issues: 23

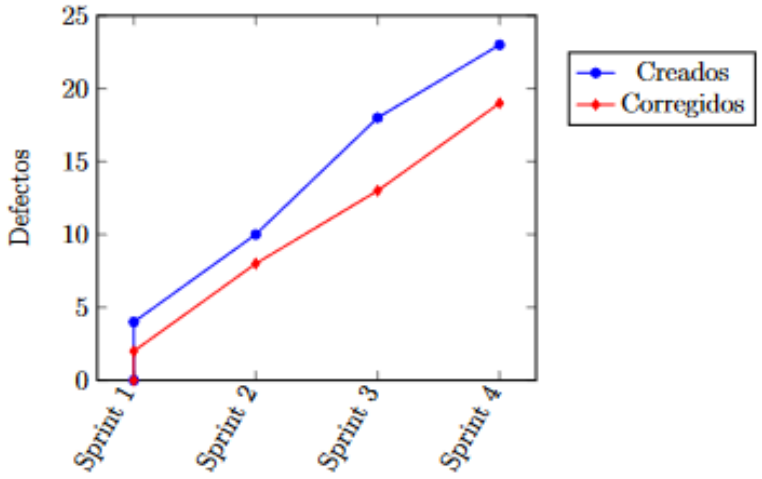
Defecto por Estado

Para la clasificación según el estado, contamos con dos clasificaciones que son defectos nuevos o creados durante las pruebas y los defectos corregidos durante el desarrollo del sistema.

Issues: 23

Defectos Creados vs Resueltos

En el siguiente gráfico se representan los defectos detectados durante los 4 sprint y la cantidad que se corrigió durante esta misma cantidad de tiempo.



Issues: 23 Creados, 19 Corregidos
Duración de tiempo: 2 Semanas (4 Sprint)

Defectos Abiertos

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

- Estilos en el bloque buscador
- Animación del menú desplegable
- Al ingresar un mail en el input de la sección "Inscríbete para registrar tu propiedad" no sucede nada.

Issues: 4

Lecciones Aprendidas / Conclusión

Como grupo nos llevamos la importancia de aplicar un control de calidad en las funcionalidades para aumentar el valor agregado de nuestro sistema, evidenciamos los beneficios de las diferentes pruebas que realizamos porque logramos detectar a tiempo defectos y bugs que de forma individual no se identificaban. Esto nos permitió minimizar los riesgos de presentar defectos en producción. Realizando estas tareas de testing nos llevamos la curiosidad por reforzar y aplicar nuestro conocimiento en esta área, ya que fue necesario realizar investigaciones y consultar con profesionales con más experticia para aumentar la calidad de nuestro producto.