

TravelBee - Reporte final de Testing

Integrantes - Grupo 8

- Segundo Del Lito
- Valentina Fiumana
- Ivan Marienhoff
- Máximo Toledo
- S. Gisela Urriza y Spreafichi

CONTENIDO	Pág.
Introducción	2
Estrategia de Testing	2
Funcionalidades desarrolladas por sprint	3
Alcances de Prueba	4
Buenas prácticas	5
Exit criteria	5
Resumen de Resultados	5
Fórmulas de cálculo de métricas	5
Pruebas manuales y automáticas - gráfico	6
Pruebas exploratorias (pasadas y falladas)	7
Pruebas Automatizadas: API Postman y Web Service -Selenium - gráfico	7
Pruebas Funcionales y no funcionales - gráfico	8
Pruebas Smoke y Regression - gráfico	8
Cantidad de casos de prueba ejecutados (pasados y fallados) y no ejecutados - gráfico	9
Reporte de defectos por estado, prioridad y severidad - gráfico	9
Porcentaje de tests creados vs resueltos - gráfico	10
Defectos Abiertos - gráfico	10
Lecciones aprendidas / Conclusión	10

Introducción

Este documento explica las distintas actividades realizadas en el marco de las pruebas de la aplicación "TravelBee".

TravelBee es una aplicación que ofrece servicios de reserva de residencias temporales. Las mismas se categorizan en : cabañas, barcos, castillos e islas.

La aplicación es amigable para su uso tanto en ordenadores como en tabletas y smartphones. Cualquier usuario, registrado o no, puede acceder al sitio, buscar propiedades filtrando tanto por fecha, y/o ciudad, y observar los detalles de cada producto.

Un usuario registrado puede no sólo rentar como locador, sino también gestionar la publicación de su propiedad, especificando información de esta como: descripción, fechas disponibles, requisitos de admisión y características. Link a la página: <http://travel-bee.sytes.net/>

Estrategia de testing

A comienzo de cada Sprint se analizaron las Épicas liberadas por nuestro Product Owner, posteriormente se recolectó la información y diseñaron los casos de prueba, respetando el flujo de desarrollo de la aplicación y evitando el diseño de pruebas repetitivas.

1. Se puso un fuerte foco en los **Niveles** de Integración y Aceptación.
2. Se diseñaron **61 casos de prueba** , para despejar dudas técnicas, localizar defectos, y guiar al Developer hacia lo requerido por nuestro Cliente.
3. **Se realizaron:**
 - **Pruebas de aceptación y usabilidad**, para asegurar por un lado que nuestra aplicación cumpliera con los requisitos de nuestro cliente, y por otro lado con las expectativas de los usuarios finales.
 - **Automatización de pruebas.** Se realizaron **26 pruebas con Postman** y **38 con Selenium**, para agilizar el testeo ahorrando tiempo y esfuerzo. 55 de 60 casos fueron automatizados.
 - **34 Smoke Tests**, para asegurar la integridad de las funcionalidades críticas de nuestra aplicación.
 - **52 Regression Tests**, para asegurar la integridad de la aplicación cada vez que agregábamos una nueva funcionalidad.
 - **10 Tests Exploratorios** de las funcionalidades más importantes de la aplicación, que nos ayudaron a encontrar defectos que no estaban contemplados en los casos de prueba

Funcionalidades desarrolladas por sprint

Sprint 1
Template general responsive
bloque buscador
bloque Listado
bloque Categorías
Botones Login y registro
Evento de login de usuario
Evento de HEADER
Visualización de HEADER en todas las páginas de la aplicación
Sprint 2
template responsive de producto
calendario de reservas y botón para acceder
Sprint 3
formulario de registro de usuarios
formulario de login
Spring Security de usuarios
Botón "Cerrar Sesión"
Búsqueda por fecha desde la página principal
Filtro por ciudad e intervalo de fechas
Botón de reserva en página de detalle de producto
formulario de reserva
Submit del formulario de reservas
Calendario interactivo en la página de reservas
Template de reserva exitosa
Sprint 4
Formulario de creación de producto
Notificación de "creación de residencia exitosa"
Bloque "mis reservas"
mail de confirmación de registro
mail de confirmación de reserva
mapa localizador de residencias en Detalles de Producto

Alcances de Prueba

Dentro del alcance:

1. Pruebas funcionales
 - Búsqueda de residencias por categoría
 - Búsqueda de residencias por ciudad, por fecha, y por ciudad y fecha
 - Persistencia de fechas en el calendario de búsqueda
 - Redireccionamiento dentro del sitio a través de botones en formularios, e isologotipo.
 - Gestión de usuarios a través de roles
 - Registro de usuario
 - Login de usuario
 - Reserva de residencias
 - Carga de nuevas residencias desde rol “Administrador”
 - Visualización
2. Pruebas no funcionales
 - Sitio web responsive y de visualización de componentes.

Fuera de alcance:

1. Pruebas de carga
2. Prueba de rendimiento
3. Pruebas de volumen
4. Pruebas de estrés
5. Pruebas de escalabilidad
6. Pruebas de seguridad
7. Pruebas en entorno UAT
8. Prueba estática de requerimientos financieros de Negocio (financista del proyecto, debe poder acceder a un Producto Mínimo Viable (MVP) gastando lo mínimo posible para poder buscar inversiones)
9. Pruebas unitarias

Buenas Prácticas

- Un porcentaje de las tareas repetitivas realizada manualmente se automatizaron creando scripts y ejecutándolos cada vez, lo que generó ahorro de tiempo y recursos.
- Se automatizaron los casos de prueba con Selenium , para ahorrar tiempo
- Se prepararon scripts de automatización a través de Postman para crear/modificar/eliminar y buscar ciudades, categorías, características, policies, productos, bookings, roles y usuarios.

Exit Criteria

Criterios de aceptación definidos:

1. Cobertura deseada:
 - Defectos abiertos de prioridad y severidad alta – 0%
 - Automatización de casos de prueba – 80% mín.
 - Ejecución de Tests en Postman – 80% PASS mín.
 - Ejecución de Tests en Selenium – 80% PASS mín.
 - Ejecución de casos de prueba Manuales 100% PASS
2. Cumplimiento de los plazos estipulados.
3. Ejecución y actualización de todos los casos de prueba.
4. Mantenimiento de defectos con su estado actual.

Resumen de Resultados

Fórmulas de cálculo de métricas

1. Porcentaje de casos ejecutados =

$$\begin{aligned} & (\text{Nro. de casos ejecutados} / \text{Nro. Total de casos de prueba escritos}) \times 100 \\ &= (60 / 61) \times 100 \\ &= 98.3 \end{aligned}$$

2. Eficacia de casos de prueba =

$$\begin{aligned} & 100 - ((\text{Nro. De defectos detectados} / \text{Nro. De casos de prueba ejecutados}) \times 100) \\ &= 100 - ((4 / 60) \times 100) \\ &= 93\% \quad (\text{los defectos se encontraron realizando test exploratorio.}) \end{aligned}$$

3. Porcentaje de casos de prueba fallidos =

$$\begin{aligned} & (\text{Nro. Total de casos de prueba fallidos} / \text{Nro. de casos de prueba ejecutados}) \times 100 \\ &= (0 / 60) \times 100 \\ &= 0 \end{aligned}$$

4. Porcentaje de casos de prueba bloqueados =

$$\begin{aligned} & (\text{Nro. Total de casos de prueba bloqueados} / \text{Nro. Total de casos de prueba ejecutados}) \times 100 \\ &= (1 / 60) \times 100 \\ &= 1.66 \end{aligned}$$

5. Porcentaje de defectos reparados =

$$\begin{aligned} & (\text{Total de defectos corregidos} / \text{Total de defectos reportados}) \times 100 \\ &= (4 / 4) \times 100 \\ &= 100 \end{aligned}$$

6. Porcentaje de defectos aceptados =

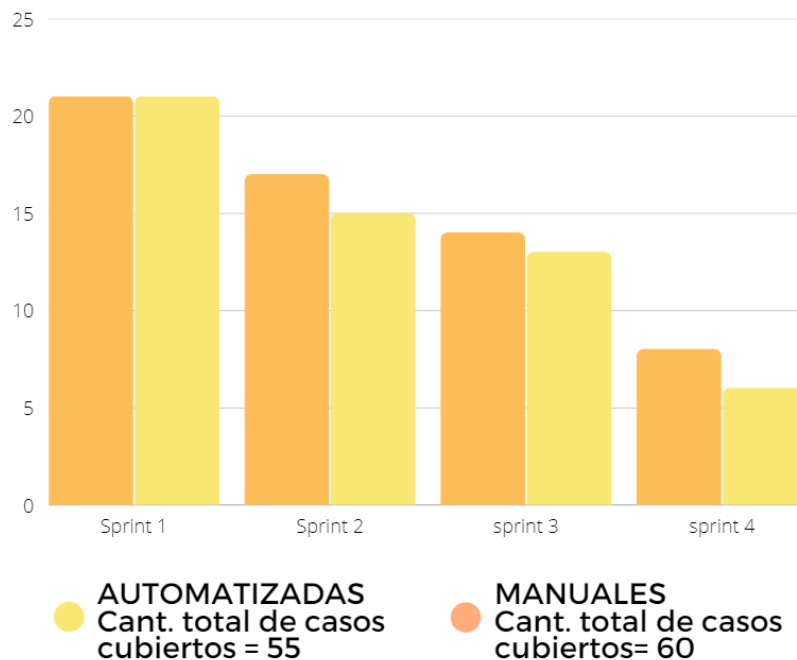
$$\begin{aligned} & (\text{Defectos aceptados como válidos por el equipo de Desarrollo} / \text{Total de defectos reportados}) \times 100 \\ &= (4 / 4) \times 100 \\ &= 100 \end{aligned}$$

7. Porcentaje de defectos diferidos =

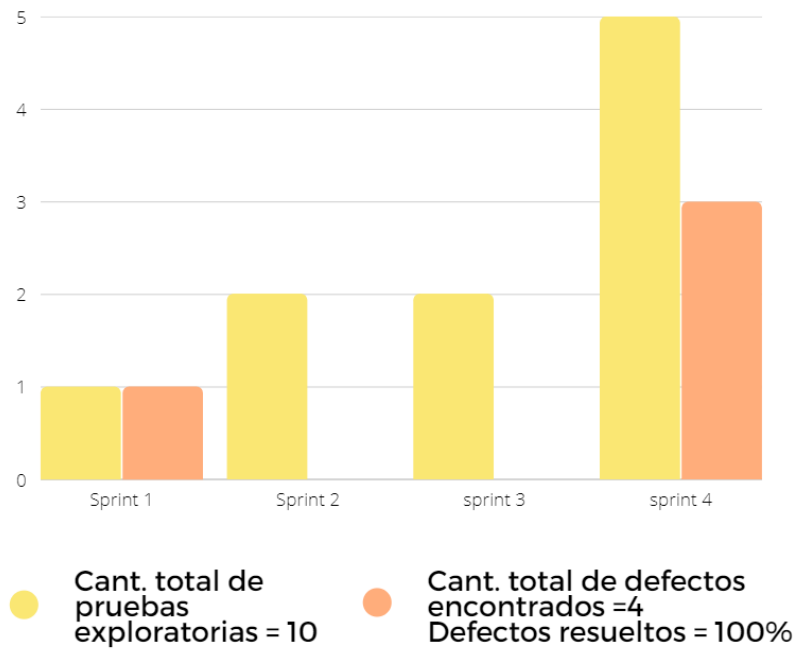
$$\begin{aligned} & (\text{Defectos aplazados para futuras versiones} / \text{Total de defectos reportados}) \times 100 \\ &= (0 / 4) \times 100 \\ &= 0 \end{aligned}$$

TIPOS DE PRUEBAS

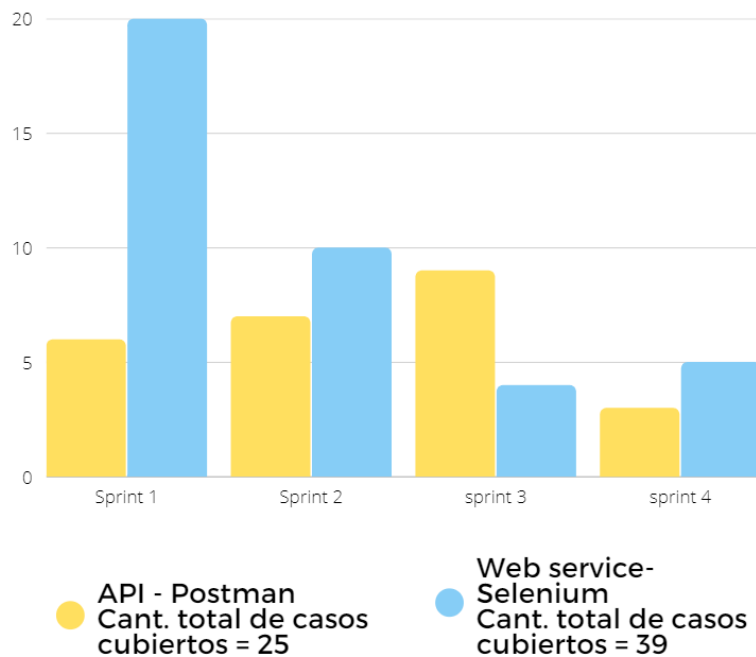
Pruebas manuales y automáticas - gráfico



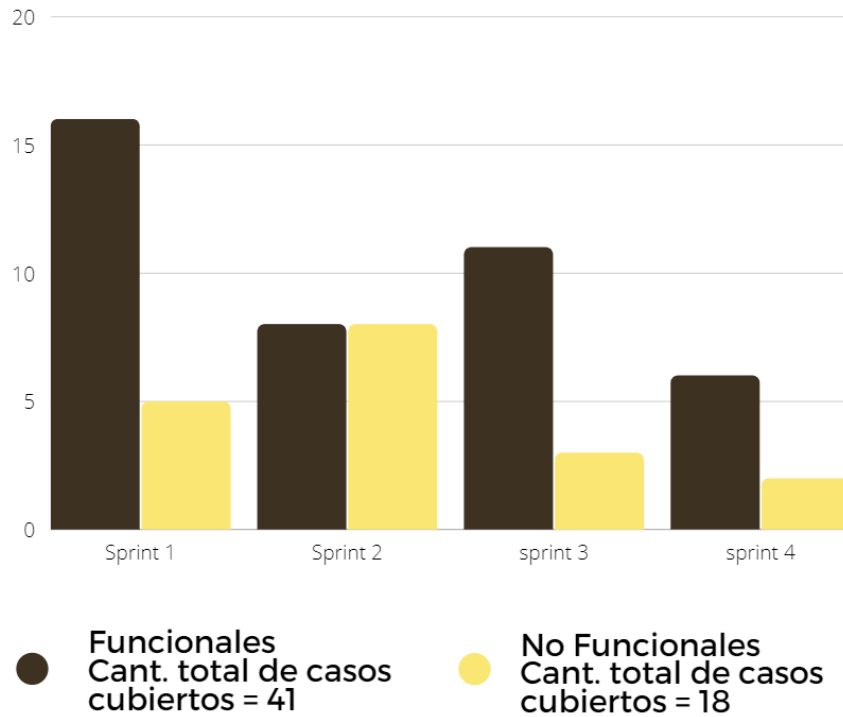
Pruebas exploratorias (pasadas y falladas) - gráfico



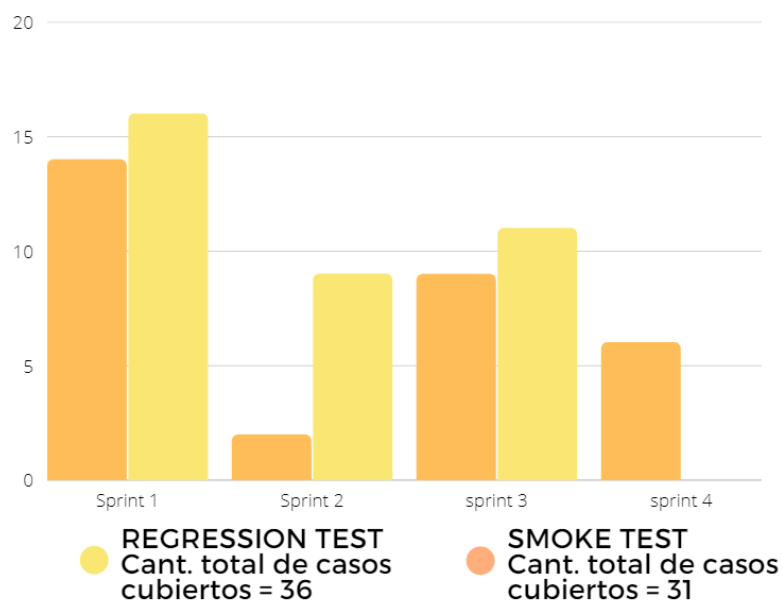
Pruebas Automatizadas: Postman, Selenium – gráfico %



Pruebas funcionales y no funcionales - gráfico

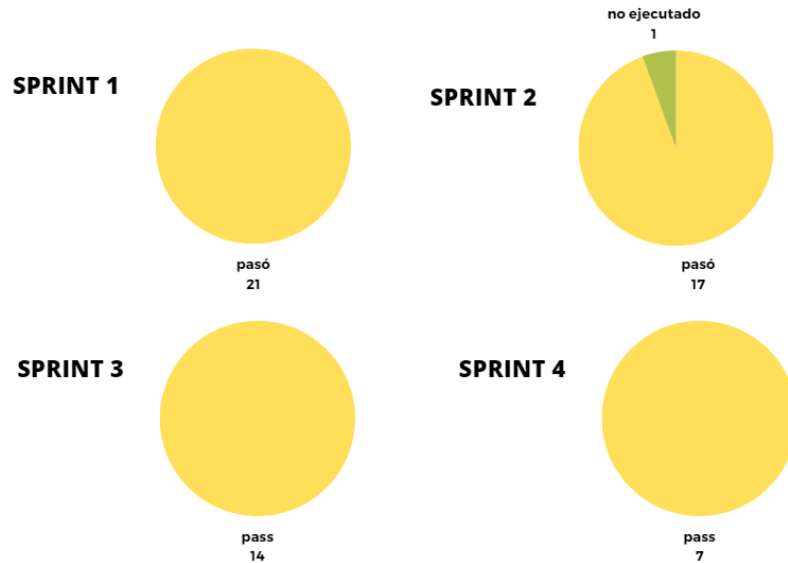


Pruebas Smoke y Regression - gráfico

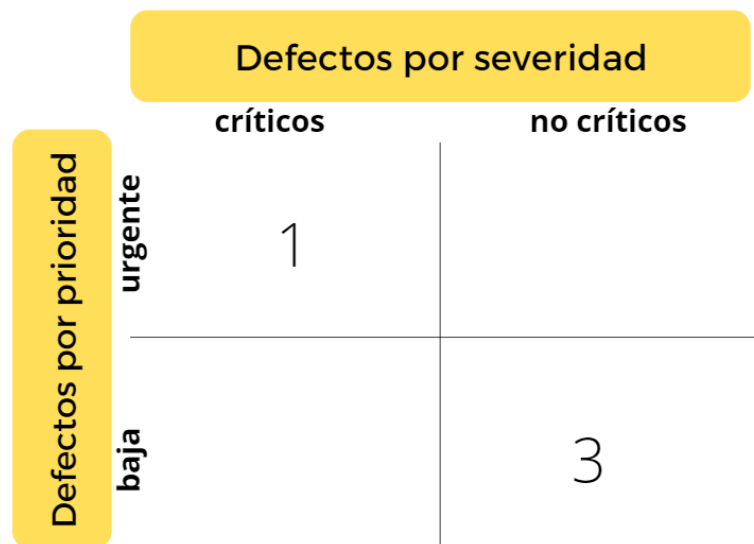


EJECUCIÓN DE PRUEBAS

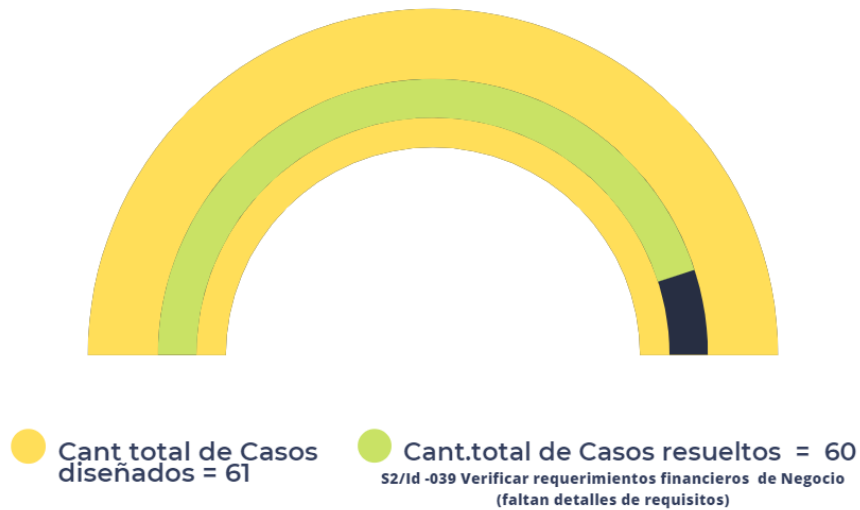
Cantidad de casos de prueba ejecutados (pasados y fallados) y no ejecutados + gráfico %



Cantidad de defectos por estado y prioridad y/o severidad + gráfico %



Porcentajes de tests creados vs resueltos

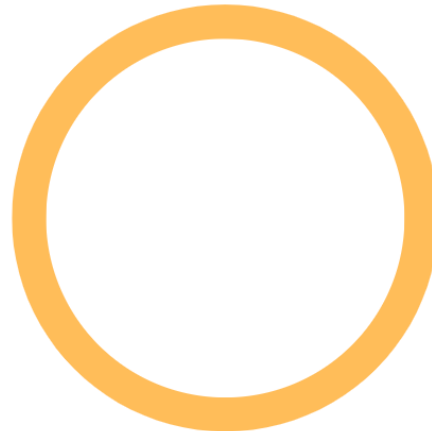


Cantidad de Defectos Abiertos

TOTAL Defectos

Sprints 1/2/3/4

- ABIERTOS = 0
- CERRADOS = 4



4

Lecciones aprendidas / Conclusión

Al haberse cumplido los criterios de salida mencionados en la página 5, el equipo de pruebas recomienda que la aplicación "Travel Bee" sea deployada al ambiente de producción .

Idealmente se recomienda realizar previamente pruebas de aceptación de usuario.