**Reporte Final de Testing**

***DIGITAL BOOKING***

***Aplicación de reservas de alojamientos***

**Grupo:** **09**

**Integrantes:**

*Balderrama, Valeria*

*Castro Feijóo, Pedro*

*Donofrio, Gianluca*

*Herrera, Tomas*

*Romero, Tomas*

**Contenido**

[**Introducción**](#_xi99jfz9m49s) **3**

[Resumen de las actividades de prueba](#_936vdgwti2zs) 3

[**Alcance**](#_bib9o2z0f5h2) **3**

[Dentro del Alcance](#_d2gi7eibz1ld) 3

[Fuera de Alcance](#_iglz7ppc8kxg) 3

[Tipos de Pruebas Ejecutadas](#_u7scyhbq3kq) 3

[Enfoque de la Prueba](#_r235vqm2w1ua) 4

[**Exit Criteria**](#_2q0hncwdrrtr) **5**

[**Resumen de Resultados**](#_njtem23xxxoe) **5**

[Diseño de Pruebas](#_7f8rtzklrzfi) 5

[Ejecución de Pruebas](#_24r2icnnfb7w) 6

[Ejecución Manual](#_tt608je7becg) 6

[Ejecución Automática](#_natknhjepgt) 6

[Reporte de Defectos](#_sii7y4y0hsw8) 6

[Todos los defectos](#_5384b1vu9cc5) 6

[Defectos por prioridad](#_q5uq6lnchdig) 7

[Defectos por Severidad](#_kw38y6o4hfld) 7

[Defecto por Estado](#_p77l9wifit) 8

[Defectos Creados vs Resueltos](#_u0mx51xzw54u) 8

[Defectos Abiertos](#_3u007c6wxpl0) 9

[**Lecciones Aprendidas / Conclusión**](#_q7ef8uwmmty2) **9**

# Introducción

El presente documento es el Informe Final de Pruebas del sistema Digital Booking, tiene el propósito de 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

**Resumen del sistema**

Digital Booking es una plataforma de reservas de alojamientos. Funciona como un portal web de viajes que genera una conexión inmediata de los consumidores con los sitios de viajes.

Se comercializan los alojamientos en función de las preferencias que el huésped proporciona (tales como tipo de alojamiento, destino, fechas de check in y check out.), para encontrar hoteles en la zona en la que quieren alojarse y cumplan con estas preferencias.

Las reservas se confirman instantáneamente para los huéspedes cuando finalizan el proceso de reserva en el sitio.

**URL para acceder al sistema**

<http://g9-s3-front-digitalbooking-anita.s3-website-us-west-2.amazonaws.com/>

**Issues de GitLab**

| **N°** | **Actividad** | **Sprint** | **Estado** |
| --- | --- | --- | --- |
| 12 | **Planificación y ejecución de los tests**  Planear casos de test basados en las historias de usuarios que existen en el sprint, y luego ejecutarlos.  Realiza prueba exploratoria del Sprint. | 1 | Finalizado |
| 13 | **Testear la API**  Validar el funcionamiento de la API Rest creada por el Back End durante el Sprint.  Utilicen Postman para las pruebas y creen los scripts de prueba necesarios. | 1 | Finalizado |
| 41 | **Testeo Automatizado (JEST)-Creación de pruebas automatizadas**  Utilizando JEST, crear scripts de prueba automatizados para cubrir sprints I y II.  Crear test unitarios y definir el porcentaje de cobertura que quiere alcanzar el equipo ( mínimo un 40 %). | 2 | En progreso |
| 42 | **Testeo Automatizado (Selenium)**  **Opcional**  Crear test funcionales utilizando Selenium Web driver.  Opcional: Crear script automático utilizando Selenium IDE. | 2 | Abierto |
| 43 | **Testeo Automatizado (Postman)**  Desarrollar scripts de pruebas automatizados utilizando Postman para las API desarrolladas en este sprint. Ejecutar con el test runner de Postman y subir a GitLab el archivo JSON. | 2 | Finalizado |
| 44 | **Testeo Exploratorio - Manual - Estático**  Testing exploratorio del sprint 2. Entregar documento con notas y forma de organización.  Casos de prueba manuales sobre las funcionalidades del sprint II.  Mantenimiento casos de prueba manuales del sprint I.Clasificar en suite de smoke y regression test.  Generar la suite de prueba con los casos que se puedan ejecutar, luego ejecutar esta suite de prueba.  Testing estático:Revisar historias de usuario del sprint II y reportar defectos en la planilla correspondiente | 2 | Finalizado |
| 67 | **Implementar tests manuales**  Pruebas exploratorias en las páginas de reserva y de confirmación de reserva. Verificar y validar las funcionalidades y la interfaz de usuario.  Test cases manuales sobre las funcionalidades a entregar en el sprint y ejecutar un ciclo de prueba con tests cases necesarios. | 3 | Finalizado |
| 68 | **Implementar tests automatizados**  Añadir las pruebas para endpoints de productos, reservas y usuarios a la colección creada previamente en **Postman.** | 3 | Finalizado |
| **Implementar tests automatizados**  Crear y ejecutar las pruebas unitarias (unit testing) de las páginas de reserva y confirmación de reserva a través de **JEST** framework. | 3 | En progreso |
| 81 | **Implementar testing automatizado**  Crear y ejecutar test unitarios utilizando JEST sobre funcionalidades de la página de mis reservas (las realizadas por usuario) y sobre la página formulario de producto (la que se ingresa con rol administrador). | 4 | En progreso |
| **Implementar testing automatizado**  Agregar a la colección de Postman script automatizados para probar los métodos agregados en este sprint. | 4 | Finalizado |
| 82 | **Implementar testing manual**  Realizar y registrar el testing exploratorio sobre las funcionalidades de la página de mis reservas y sobre la página formulario de producto.  Agregar casos de prueba Manuales para ambas páginas y clasificar los mismos en Smoke y Regression Test. | 4 | Finalizado |

# Alcance

## Dentro del Alcance

* Crear cuenta
* Login de usuario
* Seleccionar producto
* Filtrar por destino y fecha
* Calendario de reservas
* Realizar reserva
* Confirmar reserva
* Permisos de administrador
* Administrar producto
* Filtrar reservas por usuario
* Interfaz intuitiva y fácil de usar

Son pruebas que se emplean para saber cuán usable es la aplicación, pero sin entrar en aspectos funcionales. Por ejemplo, se probó que el sitio tenga un menú que haga que la navegación sea intuitiva.

## Fuera de Alcance

* Pruebas de integración de mapas
* Pruebas de seguridad

Donde se aplica el hacking ético, pruebas en las que se buscan vulnerabilidades de seguridad.

* Pruebas de rendimiento

Permiten conocer el comportamiento del software ante una carga determinada, cómo responde y cómo se recupera ante fallos.

* Pruebas de accesibilidad

Más orientadas a que se visualicen bien los elementos y a que existan alternativas para personas con alguna discapacidad visual o auditiva.

## Tipos de Pruebas Ejecutadas

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

## Enfoque de la Prueba

Se crearon test funcionales tanto positivos como negativos para cada historia de usuario y funcionalidad del sistema. Cuando se recibía un nuevo build desde desarrollo se ejecutaban las pruebas de humo y luego las pruebas correspondientes a las nuevas funcionalidades.

Los defectos se reportaron en la planilla de defectos. Antes de finalizar el sprint se ejecutaba una prueba exploratoria y los defectos encontrados se cargaban sobre la misma planilla de defectos.

En el último sprint se ejecutó la suite de pruebas de regresión completa y se reportaron los defectos encontrados en la planilla de defectos.

**Link Planilla de Casos de Prueba:**

[**https://docs.google.com/spreadsheets/d/1zGtWJkgpdgxzMv0fv876juBkQ2l7VN51/edit?usp=sharing&ouid=107935230562354087769&rtpof=true&sd=true**](https://docs.google.com/spreadsheets/d/1zGtWJkgpdgxzMv0fv876juBkQ2l7VN51/edit?usp=sharing&ouid=107935230562354087769&rtpof=true&sd=true)

**Link Planilla de Defectos:**

[**https://docs.google.com/spreadsheets/d/1zGtWJkgpdgxzMv0fv876juBkQ2l7VN51/edit#gid=1173992132**](https://docs.google.com/spreadsheets/d/1zGtWJkgpdgxzMv0fv876juBkQ2l7VN51/edit#gid=1173992132)

# 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.
* *La ejecución de las pruebas de unidad debe tener un 100% de tasa de test en estado pasados*
* *La ejecución de los Test de Postman deben tener un 100% de past rate.*
* *La ejecución de los test de regresión debe tener un 90% de pass rate.*

# Resumen de Resultados

## Diseño de Pruebas

|  | ***Test Manuales*** | ***Test Automáticos*** | ***Test de Integración (Postman)*** | ***Test Total*** |
| --- | --- | --- | --- | --- |
| *Login de Usuario* | *20* | *1* | *11* | *32* |
| *Crear Cuenta* | *16* | *1* | *6* | *23* |
| *Seleccionar - filtrar Producto* | *17* | *-* | *10* | *27* |
| *Realizar Reserva* | *18* | *-* | *13* | *31* |
| *Administrar Producto* | *12* | *-* | *9* | *21* |

## Ejecución de Pruebas

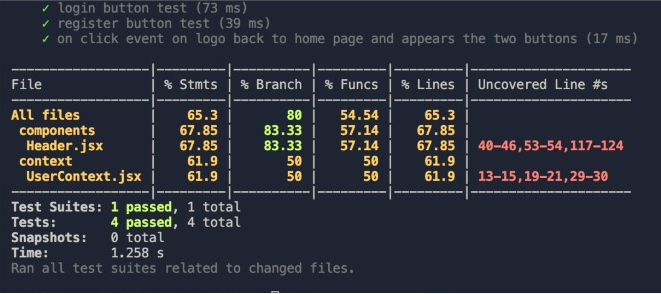
### Ejecución Manual

Resultados de las pruebas de Regresión ejecutado en el **Sprint 4:**

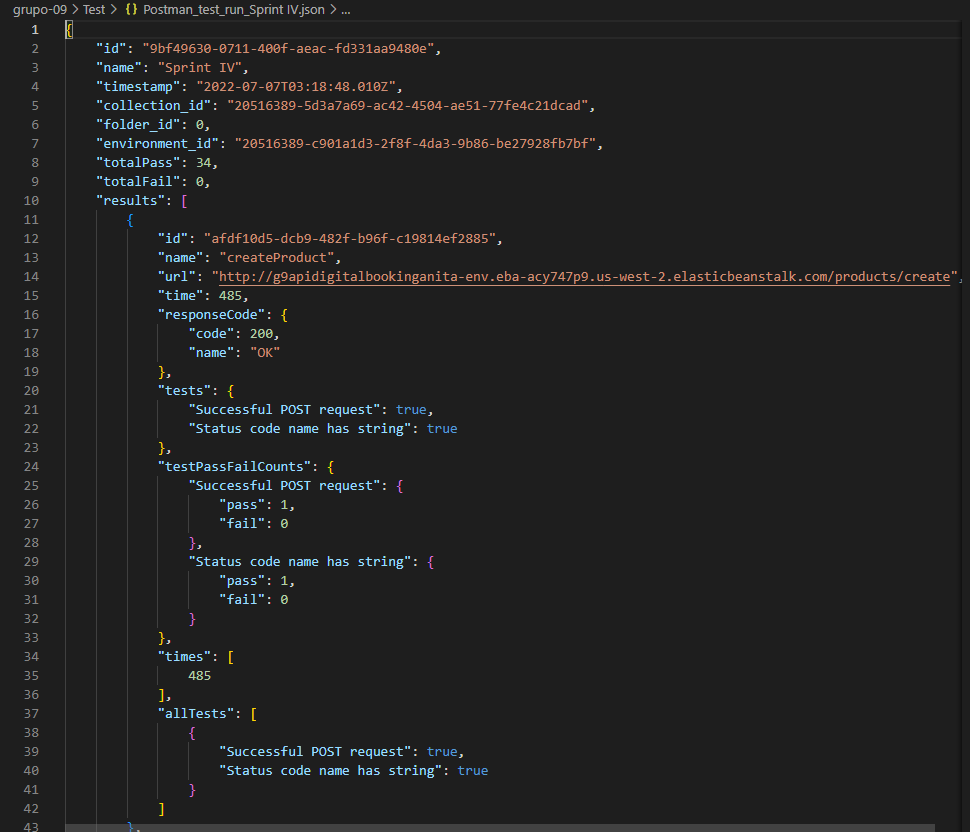
|  | **Test Pasado** | **Test Fallados** | **Test no ejecutados** | **Test Total** |
| --- | --- | --- | --- | --- |
| Login de Usuario | 2 | 0 | 0 | 2 |
| Crear Cuenta | 2 | 0 | 0 | 2 |
| Seleccionar Producto | 2 | 0 | 0 | 2 |
| Realizar Reserva | 8 | 1 | 0 | 8 |
| Administrar Producto | 7 | 1 | 0 | 7 |

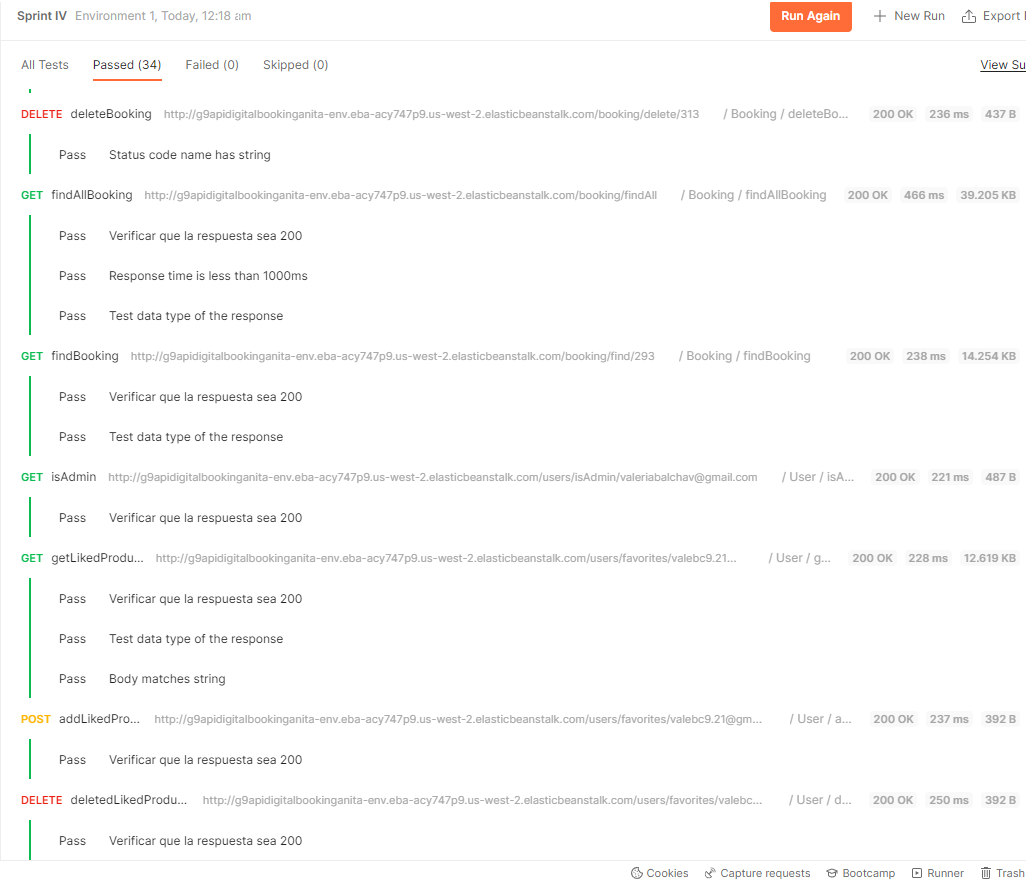
### Ejecución Automática

Resultados de la ejecución de las pruebas de unidad, ejecutados en el **sprint 4**. Captura de pantalla con el informe de coverage que se obtiene de JEST

**

Resultados de la ejecución de los test de integración en Postman, ejecutados en el **sprint 4**.

**

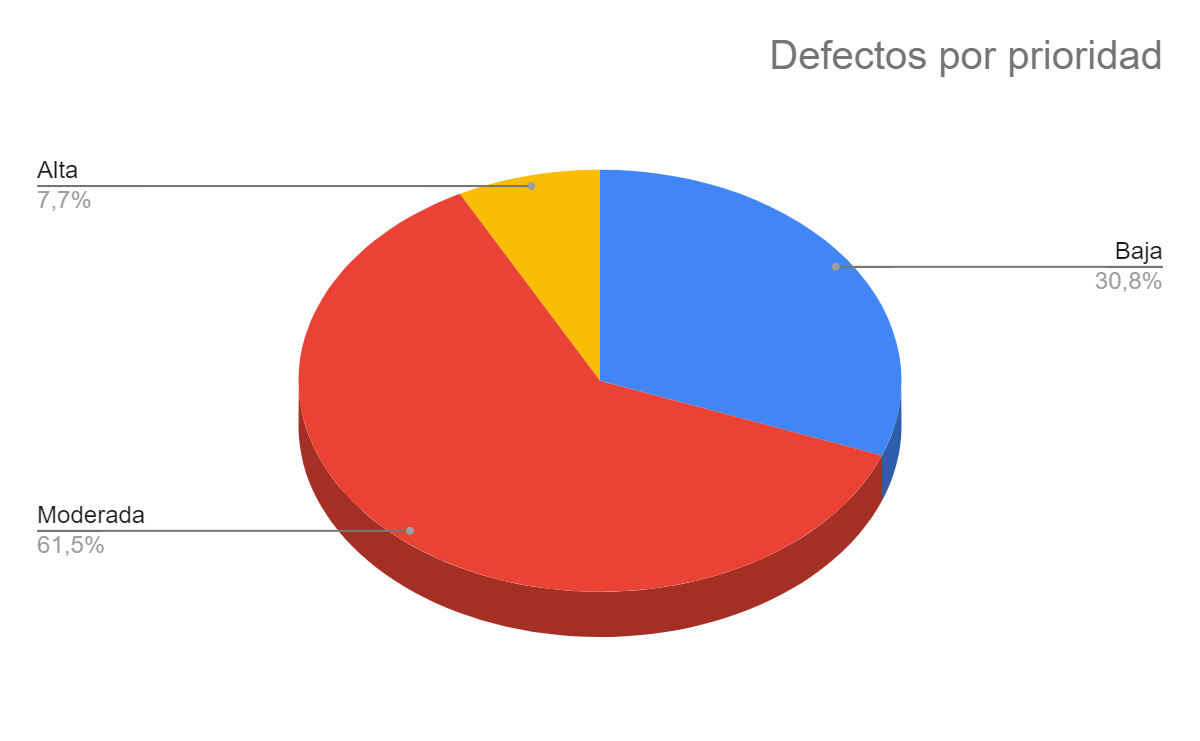
**

## Reporte de Defectos

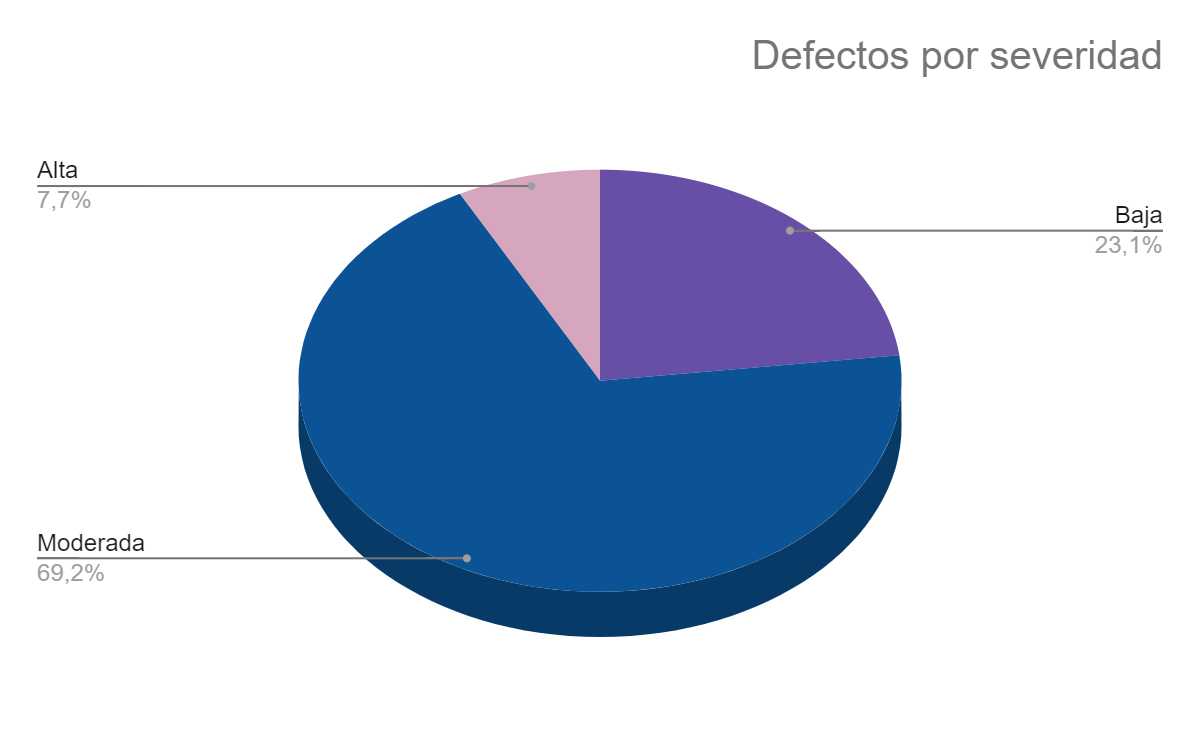
## Todos los defectos

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

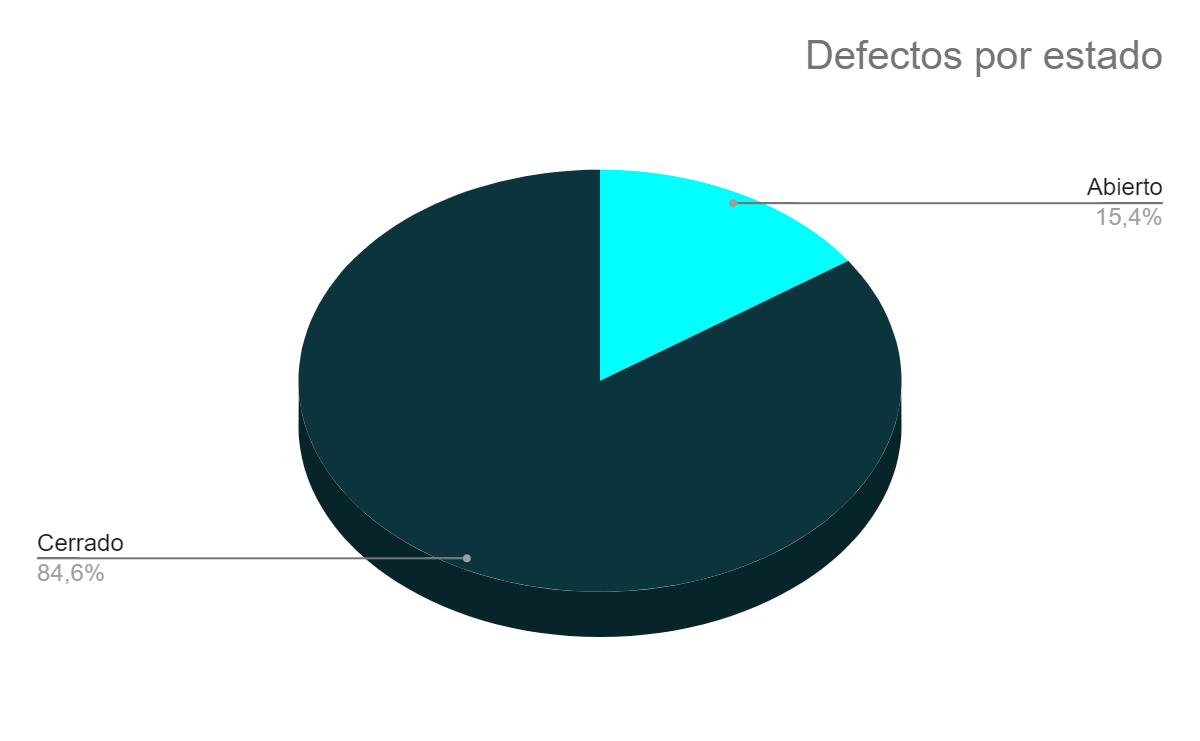
### **Defectos por prioridad**

**

### **Defectos por Severidad**

**

### Defecto por Estado



**Tabla resumen:**



## Defectos Abiertos

Número total de defectos que permanecen abiertos al final de la fase de pruebas:

Al final de la fase de pruebas se tienen dos defectos en estado abierto, sin embargo, se cumple con el criterio de salida: No son defectos de severidad crítica y/o bloqueante. Ambos defectos corresponden al último sprint, el detalle de los mismos se encuentra en el link de la planilla de defectos en el apartado: [Enfoque de la prueba](#_r235vqm2w1ua)

# 

# Lecciones Aprendidas / Conclusión

* El testing es una herramienta fundamental para evitar que los errores o puntos vulnerables en el desarrollo del proyecto lleguen al cliente, o al usuario.
* Las pruebas manuales son indispensables para cada proyecto, ya que por medio de ellas se detectan la mayor parte de los defectos, y de esta forma se logra estabilizar el sistema que se está probando.
* Las pruebas manuales permiten un análisis profundo por parte de una persona, lo cual es un mayor beneficio cuando se quiere mejorar la experiencia de usuario.
* La repetición de pruebas manuales podrían dejar pasar defectos, ya que una acción repetitiva para una persona es cansada y tediosa.
* El testing posibilita los procesos, métodos de trabajo y herramientas necesarias para garantizar la calidad de cualquier desarrollo.
* Las pruebas de software permiten la ejecución de un sistema con el objetivo principal de encontrar errores presentes en el mismo; así conseguimos reducirlos y mejorar la calidad del sistema, haciéndolo más robusto frente a fallos.
* Las pruebas automatizadas son la opción más viable cuando se necesita realizar diversos test cases, de manera repetitiva y por un período de tiempo extenso.
* Las pruebas manuales y las automatizadas son complementarias, ya que las pruebas segundas deben aplicarse hasta que un sistema se encuentre estable, es decir que haya pasado por el proceso de pruebas manuales.