Reporte Final de Testing

Digi Rent

Integrantes:

* Correa, Natalí
* Goncalves Colaco, Mariela
* Lobos, Gustavo
* Merlo, Jonatan
* Rivarola, Agustin
* Suarez, Victoria

Contenido

[Introducción](#_heading=h.gjdgxs) **3**

[Resumen de las actividades de prueba](#_heading=h.30j0zll) 3

[Resumen del sistema](#_heading=h.r5fl0lypdbg8) 3

[Issues de GitLab](#_heading=h.8ts6pvq9j29t) 3

[Alcance](#_heading=h.1fob9te) **3**

[Dentro del Alcance](#_heading=h.3znysh7) 3

[Testing Manual](#_heading=h.c5e6e1pc9je3) 3

[Testing Automático](#_heading=h.ethvdnhkimw9) 3

[Fuera de Alcance](#_heading=h.2et92p0) 4

[Tipos de Pruebas Ejecutadas](#_heading=h.tyjcwt) 4

[Enfoque de la Prueba](#_heading=h.3dy6vkm) 5

[Link Planilla de Casos de Prueba:](#_heading=h.9oahqncrr305) 5

[Link Planilla de Defectos:](#_heading=h.fhvw0t1ecpbe) 5

[Enfoque de testing para Backend](#_heading=h.7jpbjfeigemq) 5

[Enfoque de testing para Componentes de React](#_heading=h.q4wmwmdgsger) 5

[Pruebas exploratorias](#_heading=h.u2eonkh7c27l) 5

[Exit Criteria](#_heading=h.1t3h5sf) **5**

[Resumen de Resultados](#_heading=h.4d34og8) **6**

[Diseño de Pruebas](#_heading=h.2s8eyo1) 6

[Ejecución de Pruebas](#_heading=h.17dp8vu) 7

[Ejecución Manual](#_heading=h.3rdcrjn) 7

[Ejecución Automática de los tests desarrollados en Jest](#_heading=h.26in1rg) 7

[Reporte de Defectos](#_heading=h.agefusczsxiy) 13

[Todos los defectos](#_heading=h.35nkun2) 13

[Defectos por prioridad](#_heading=h.q9bvvk83i6s2) 13

[Defectos por Severidad](#_heading=h.44sinio) 14

[Defectos por Estado](#_heading=h.2jxsxqh) 14

[Defectos Creados vs Resueltos](#_heading=h.z337ya) 15

[Defectos Abiertos](#_heading=h.3j2qqm3) 15

[Lecciones Aprendidas / Conclusión](#_heading=h.1y810tw) **16**

# Introducción

Este documento es el Informe Final de Pruebas del sistema DigiRent. 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

### Resumen del sistema

El sistema permite a los usuarios visualizar diferentes opciones de alojamiento, teniendo la posibilidad de realizar búsquedas por diferentes criterios, seleccionar un alojamiento para conocer más detalles sobre el mismo y poder realizar reservas de manera segura ingresando con un usuario y contraseña que previamente debe haber registrado. Por otra parte, también permite a los propietarios de los alojamientos dar de alta sus productos, indicando todas sus particularidades y datos distintivos de las demás propiedades.

🔗 [URL de acceso a la API en AWS](http://54.202.224.239:8080/swagger-ui/index.html#/n%20AWS)

🔗 [URL de acceso al sitio web en AWS](http://front-digirentg11.s3-website-us-west-2.amazonaws.com/)

### Issues de GitLab

[🔗 Acceso a planilla de Issues](https://docs.google.com/spreadsheets/d/1k1AiAqmZi4FrJQk3QOmqf_eKD9FvFtVPtt3BwrkxRGQ/edit?usp=drivesdk)

# Alcance

## Dentro del Alcance

### Testing Manual

Desde el testeo manual se hicieron todas las pruebas necesarias y lo más abarcativas posible. Pruebas positivas y negativas, pruebas para constatar tanto el funcionamiento como la estética, también se evaluó el responsive y la experiencia de usuario.

### Testing Automático

Se realizaron pruebas automatizadas para comprobar las altas, bajas, modificaciones y consultas a los endpoints establecidos en la API que tenían impacto en la base de datos. Para estás pruebas se utilizó Postman y se verificó a través de tests si los pedidos a la API daban la respuesta esperada (códigos de respuesta satisfactoria cómo los 200 o 201 y códigos de error como 403 o 500). También se tuvo en cuenta si en el body de la respuesta venían los datos esperados, si existían los IDs, y si la respuesta esperada era un array también se tuvo en cuenta este detalle para armar los scripts de pruebas.

En cuanto a las pruebas automatizadas con Jest, se prepararon los entornos de ejecución para las pruebas, con el correspondiente renderizado de los componentes y rutas necesarias. La utilización de React Router hizo necesario tener en cuenta ciertas particularidades al momento de preparar los tests. Se siguieron los caminos que se espera siga un usuario y se tuvieron en cuenta las pruebas manuales como guía para automatizar los recorridos y las respuestas esperadas en cada acción. Se verificó el renderizado de los componentes y se comprobó que se cumplieran correctamente los resultados esperados en la navegación.

## Fuera de Alcance

No se realizaron pruebas sobre librerías externas. Las librerías más importantes excluidas fueron el calendario, la galería de imágenes y el renderizado del mapa.

Tampoco se utilizaron simulaciones para testear los tiempos de respuesta de la API.

## Tipos de Pruebas Ejecutadas

|  | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 |
| --- | --- | --- | --- | --- |
| *Prueba Estática* | *SI* | NO | NO | SI |
| *Prueba Exploratoria* | *SI* | *SI* | *SI* | *SI* |
| Prueba de Humo | Si | Si | Si | Si |
| Prueba de Regresión | No | No | Si | Si |
| Prueba de Componente / Unidad (JEST) | No | Si | Si | Si |
| Prueba de Integración (Postman) | No | Si | Si | Si |

## Enfoque de la Prueba

### Link Planilla de Casos de Prueba:

[Casos de Prueba](https://docs.google.com/spreadsheets/d/1k1AiAqmZi4FrJQk3QOmqf_eKD9FvFtVPtt3BwrkxRGQ/edit#gid=1167556197)

### Link Planilla de Defectos:

[Defectos](https://docs.google.com/spreadsheets/d/1k1AiAqmZi4FrJQk3QOmqf_eKD9FvFtVPtt3BwrkxRGQ/edit#gid=1809237974)

### Enfoque de testing para Backend

Desde el Sprint 2 se desarrollaron pruebas automatizadas utilizando Postman en simultáneo con el desarrollo de la API. Las pruebas fueron realizadas tanto por el equipo de desarrollo como por el equipo de testing, e incluyeron además pruebas manuales. En la medida que se encontraban errores y defectos por parte de desarrollo se solucionaban mientras que los defectos encontrados por el equipo de testing eran reportados y se les realizaba un seguimiento específico.

Se iniciaron pruebas positivas pero también pruebas negativas, estas últimas se utilizaron como guía para incluir en el desarrollo del sistema excepciones descriptivas a los fines de ser más útiles tanto para el equipo de desarrollo como para dar respuestas más orientativas hacía el frontend.

### Enfoque de testing para Componentes de React

Se tuvieron en cuenta las particularidades de utilizar librerías y componentes externos como React Router. Se buscó probar de manera individual los componentes tratando de lograr una cobertura mayor al 40% mediante los tests de Jest.

Para el desarrollo de las pruebas automatizadas se priorizó al usuario, intentando lograr una simulación lo más cercana posible a su comportamiento.

### Pruebas exploratorias

Desde el inicio se realizaron pruebas exploratorias para seguir el desarrollo de la API y del sitio web, evaluando la navegabilidad, considerando el cumplimiento o no de un diseño responsive, tratando de abarcar la mayor cobertura posible. Se fueron reportando los defectos encontrados y se realizó seguimiento de los mismos. Los defectos informados fueron incorporados como incidencias al desarrollo del sistema.

# 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.
* Los tests de Postman requieren 100% de past rate.
* Los tests de Jest requieren al menos un 40% de cobertura total y al menos un 60% de cobertura minima en los componentes testeados en cada Sprint.

# Resumen de Resultados

## Diseño de Pruebas

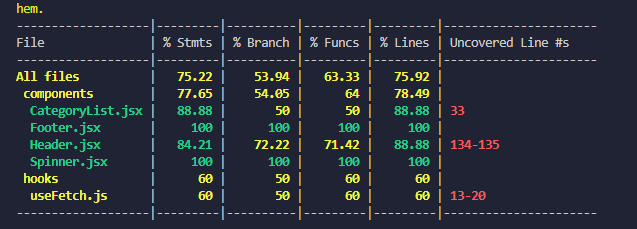
|  | Test Manuales | Test Automáticos | Test de Integración (Postman) | Test Total |
| --- | --- | --- | --- | --- |
| Header de la web | 12 | 1 | 0 | 13 |
| Body y footer de la web | 15 | 1 | 7 | 23 |
| Barra de búsqueda | 8 | 0 | 27 | 35 |
| Login | 12 | 5 | 4 | 21 |
| Registro | 11 | 1 | 4 | 16 |
| Ruteo | 2 | 3 | 0 | 5 |
| Página de producto | 30 | 0 | 27 | 57 |
| Reservas | 24 | 0 | 22 | 46 |
| Administración de producto | 17 | 0 | 85 | 102 |

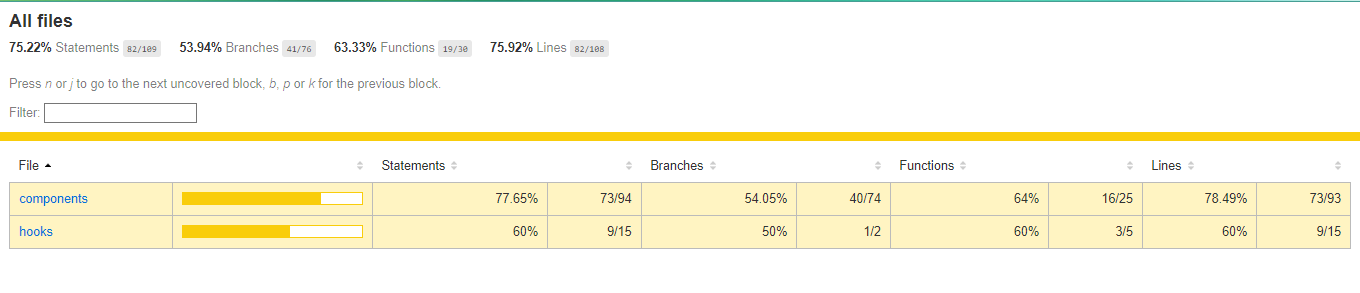
## Ejecución de Pruebas

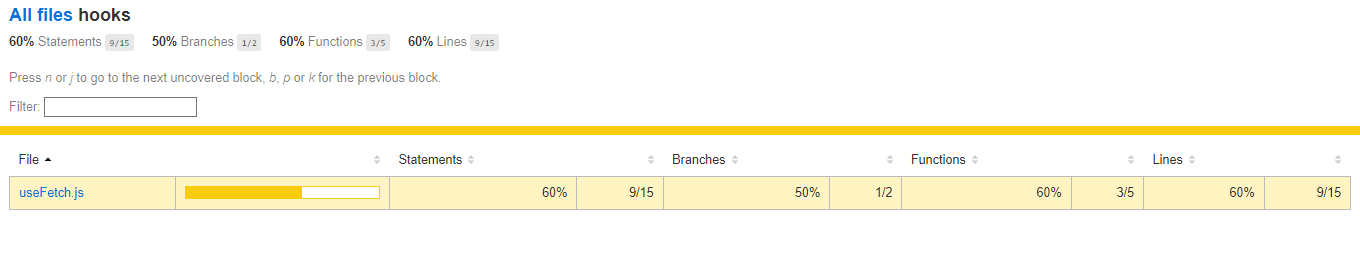
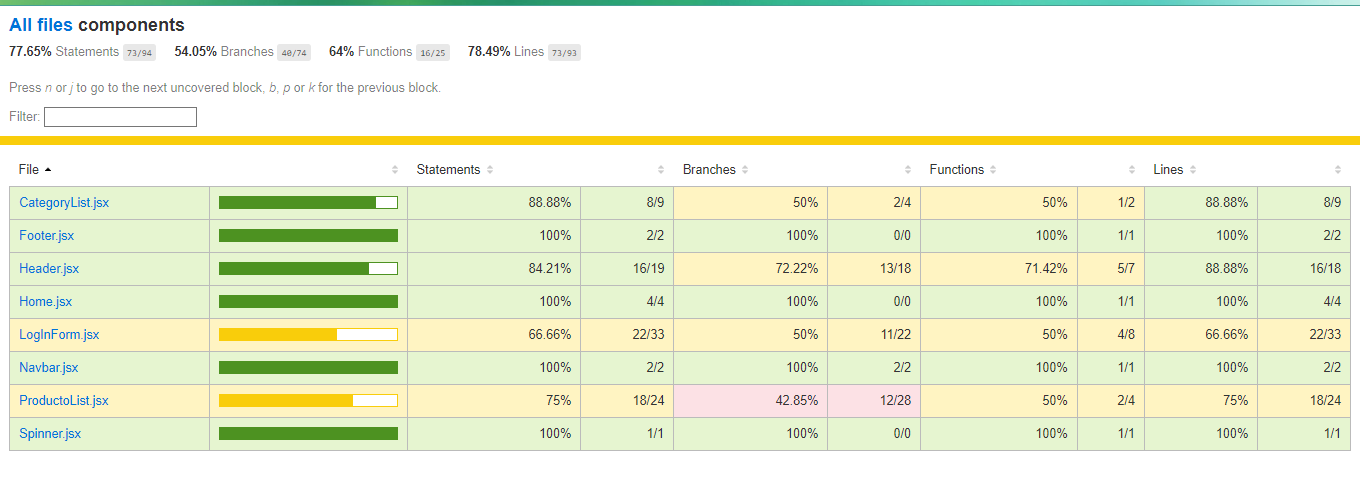
### Ejecución Manual

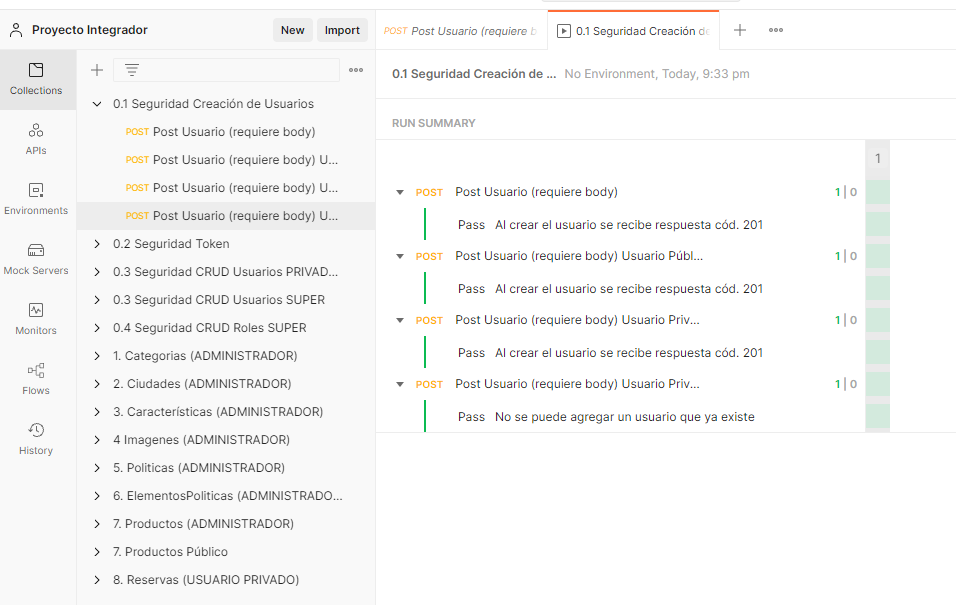
|  | Test Pasado | Test Fallados | Test no ejecutados | Test Total |
| --- | --- | --- | --- | --- |
| Header de la web | 2 | 0 | 0 | 2 |
| Producto | 7 | 0 | 0 | 7 |
| Barra de búsqueda | 4 | 0 | 0 | 4 |
| Login | 6 | 0 | 0 | 6 |
| Registro | 4 | 0 | 0 | 4 |
| Reservas | 28 | 6 | 0 | 34 |
| Administración de producto | 17 | 0 | 0 | 17 |

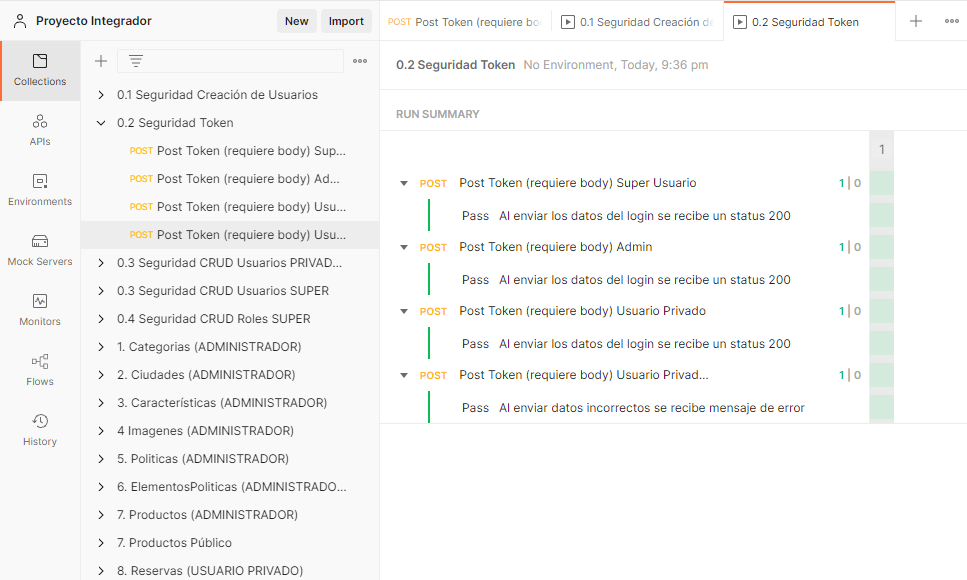
### Ejecución Automática de los tests desarrollados en Jest

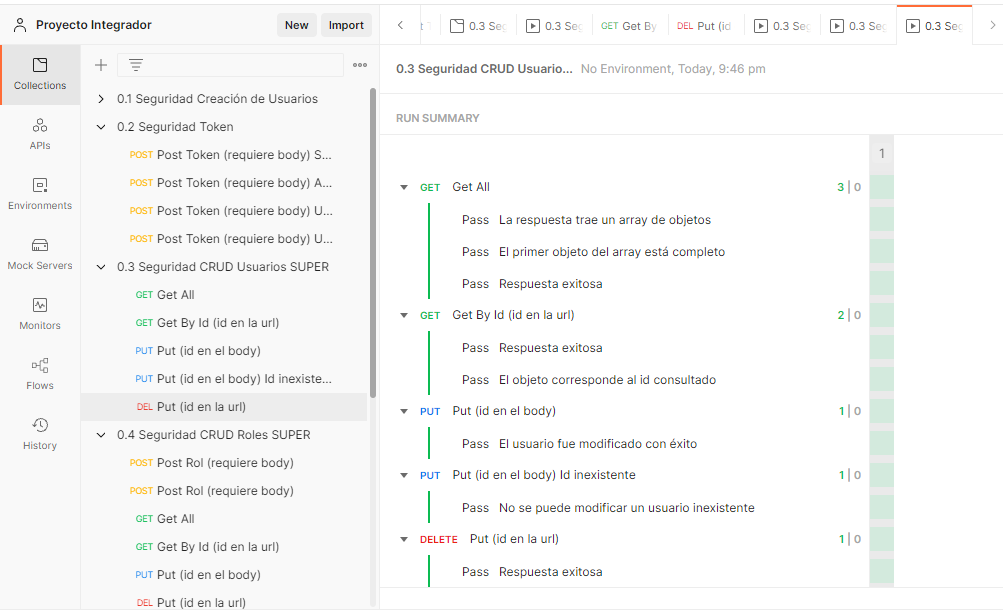
**

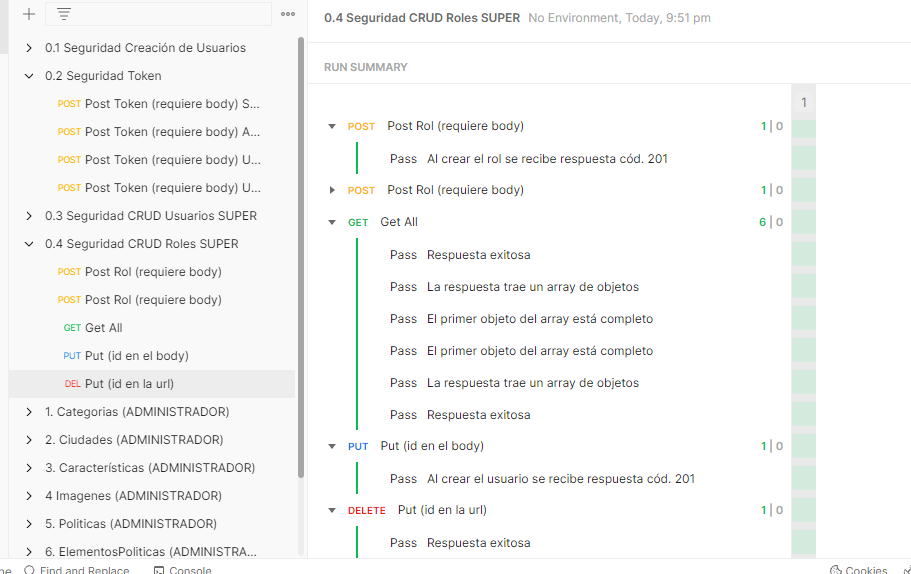
**

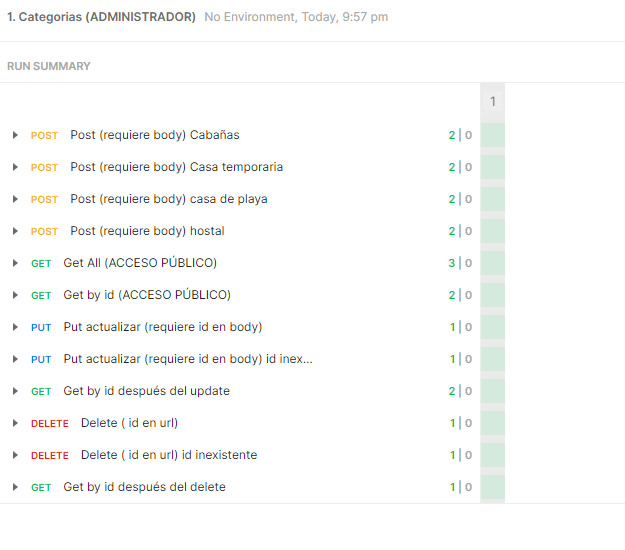
**

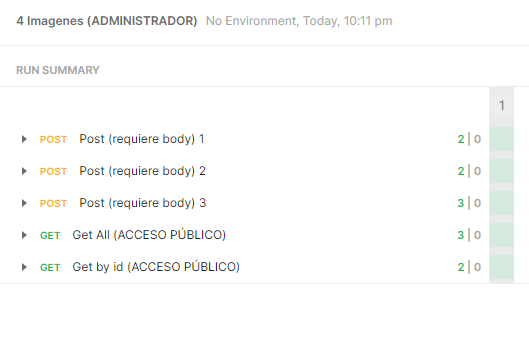
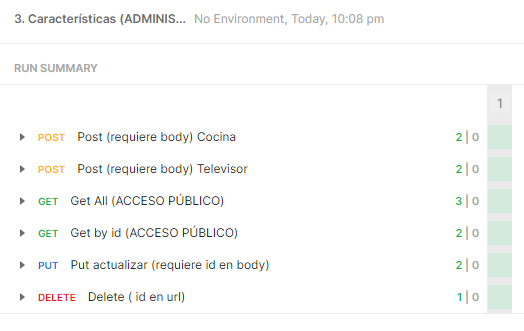
Ejecuciones en Postman

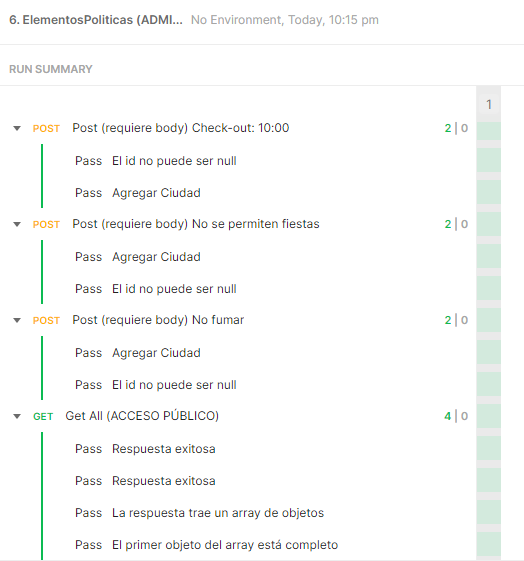


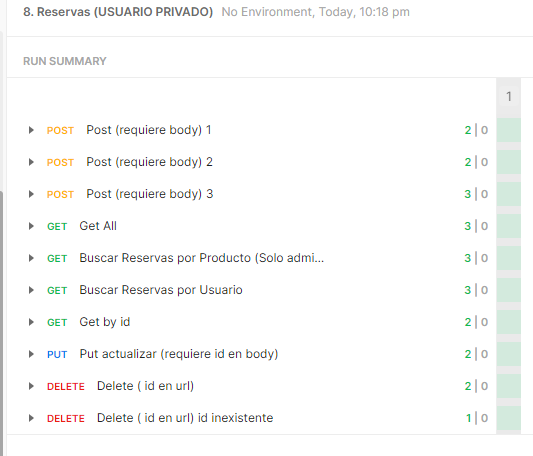
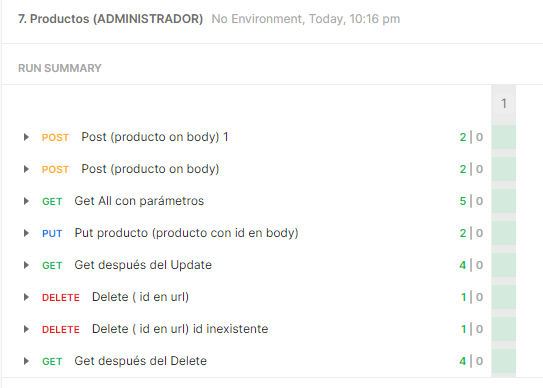


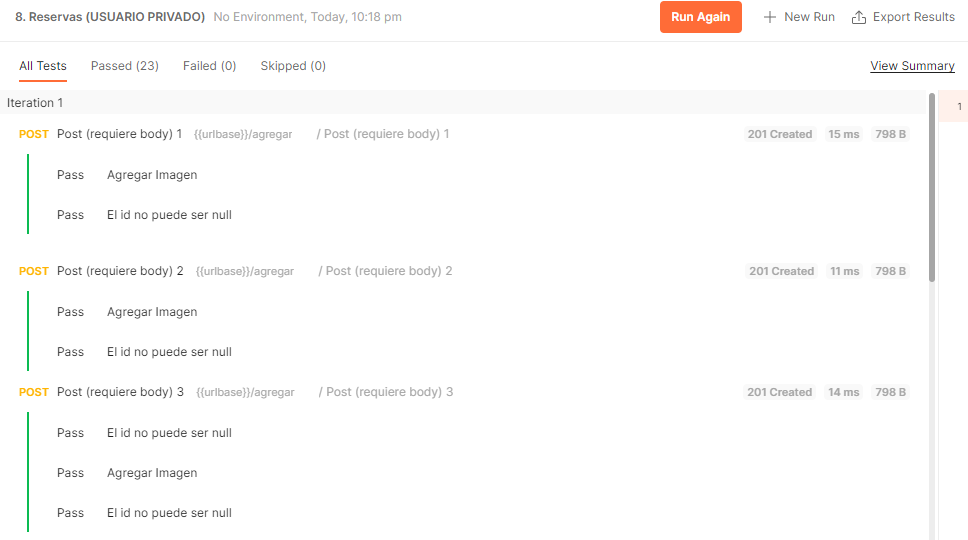


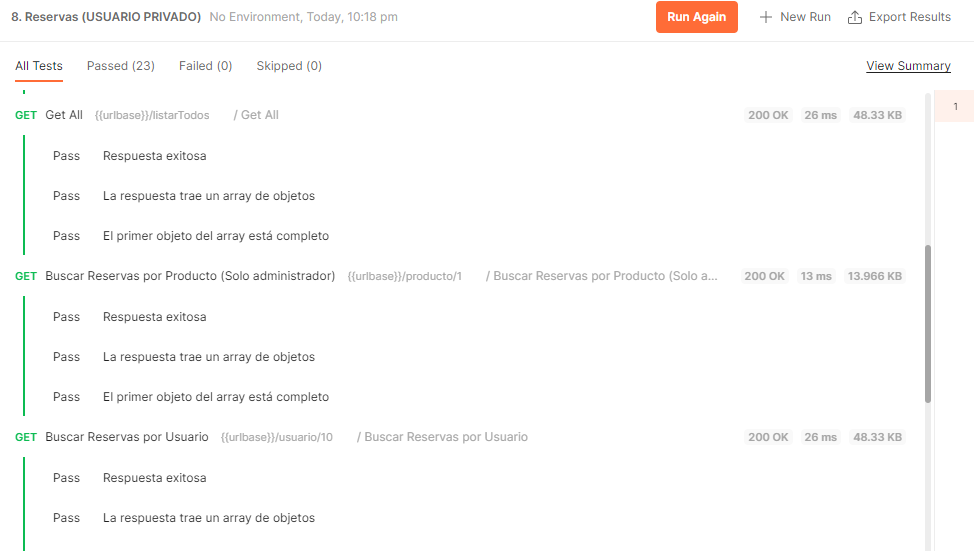


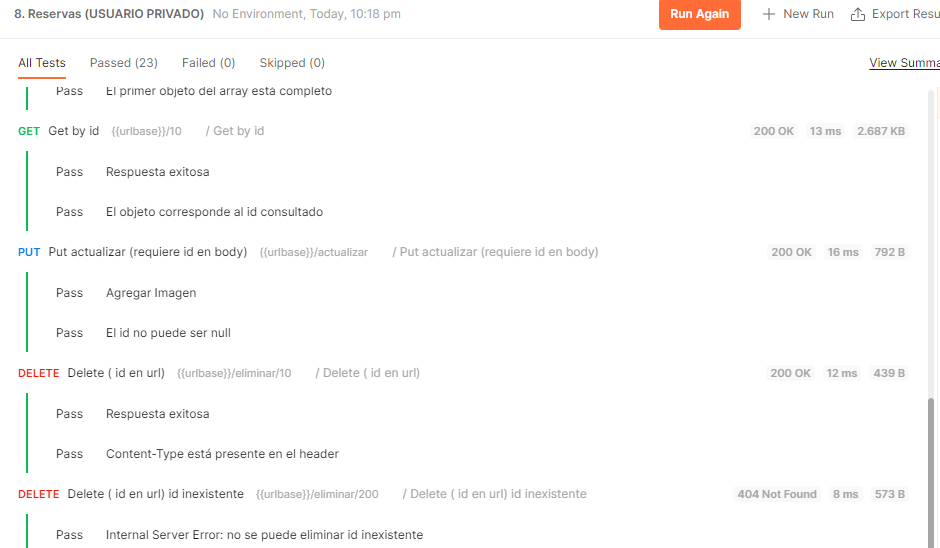










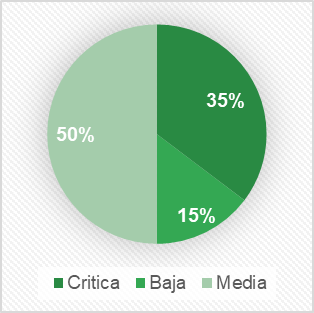


## Reporte de Defectos

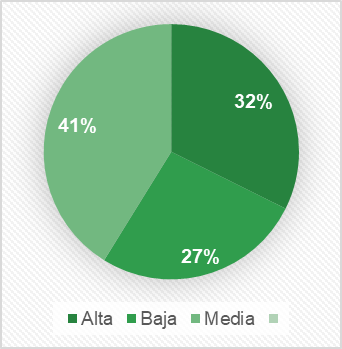
## Todos los defectos

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

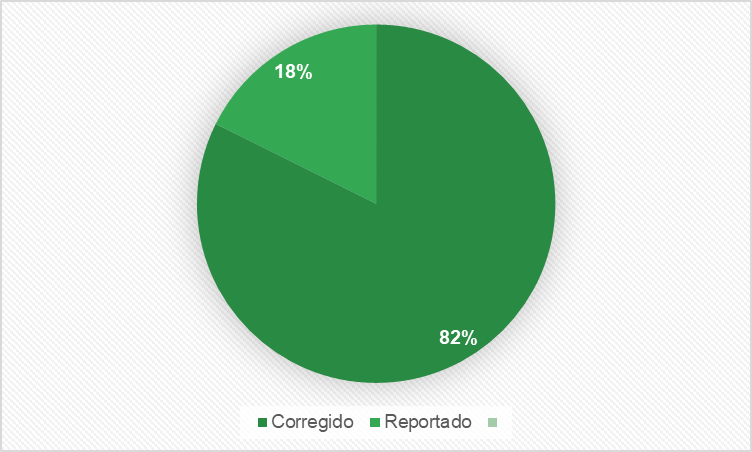
### Defectos por prioridad

**

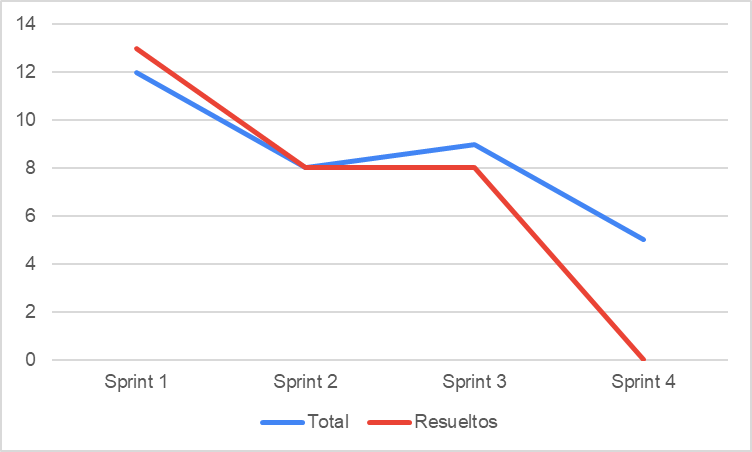
### Defectos por Severidad



### Defectos por Estado

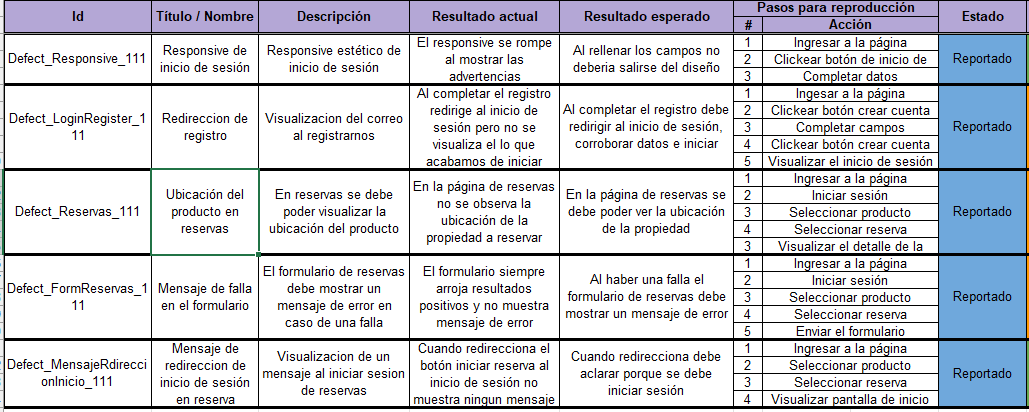


### Defectos Creados vs Resueltos

**

## Defectos Abiertos

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

**

# Lecciones Aprendidas / Conclusión

El desarrollo acompañado del proceso de testing fue fundamental para poder resolver de manera rápida y eficaz los problemas que se iban presentando. El testing no sólo fue útil para mejorar la calidad del software sino que también fue una guía para lograr soluciones más productivas. Permitió detectar errores y defectos en fases tempranas lo que dió tiempo a poder solucionar los contratiempos.