

ÍNDICE

[**Organización del equipo 3**](#_tz8bawijwnwt)

[Diagrama de Gantt 3](#_tbiylfd6lq42)

[El producto 4](#_pt8ddqg9qwx7)

[Épica 4](#_778jucrawboh)

[Mapa del sitio 4](#_paiwndgk5n2o)

[Diagrama de flujo acumulado 5](#_6mjituajvyyf)

[Ejecuciones totales de casos de prueba 5](#_zaoz0dn2cxm)

[**Desarrollo del trabajo 6**](#_v8ydkd4ddnkg)

[Cantidad total de incidencias 6](#_nbadx3t03l5b)

[Sprint 1 7](#_hn6o3y61orzv)

[Sprint 2 8](#_pqh5o5ewie5r)

[**Seguridad 9**](#_aaenbdiead91)

[**API Testing 11**](#_c36me9aerx8a)

[Introducción para el testing de apis 11](#_bleksngf3jdo)

[Objetivos 11](#_ut5c37h29s73)

[Automatización con Snippets 11](#_b5t5nviqoc0p)

[Resultados obtenidos en postman 11](#_81z0bp2fa5kv)

[**Performance de API Testing 13**](#_tx270wnxpkvm)

[**Automation 15**](#_jc08uvfi4tza)

[**Bases de Datos 17**](#_dye82qhknbaa)

[Diagrama ER original 17](#_6bt03jp4mkwk)

[BBDD Propuesta 18](#_w8yi6rhos79)

[**UX/UI 19**](#_pbv7c3wasib)

[Versión actual 19](#_gjeugws4vcv2)

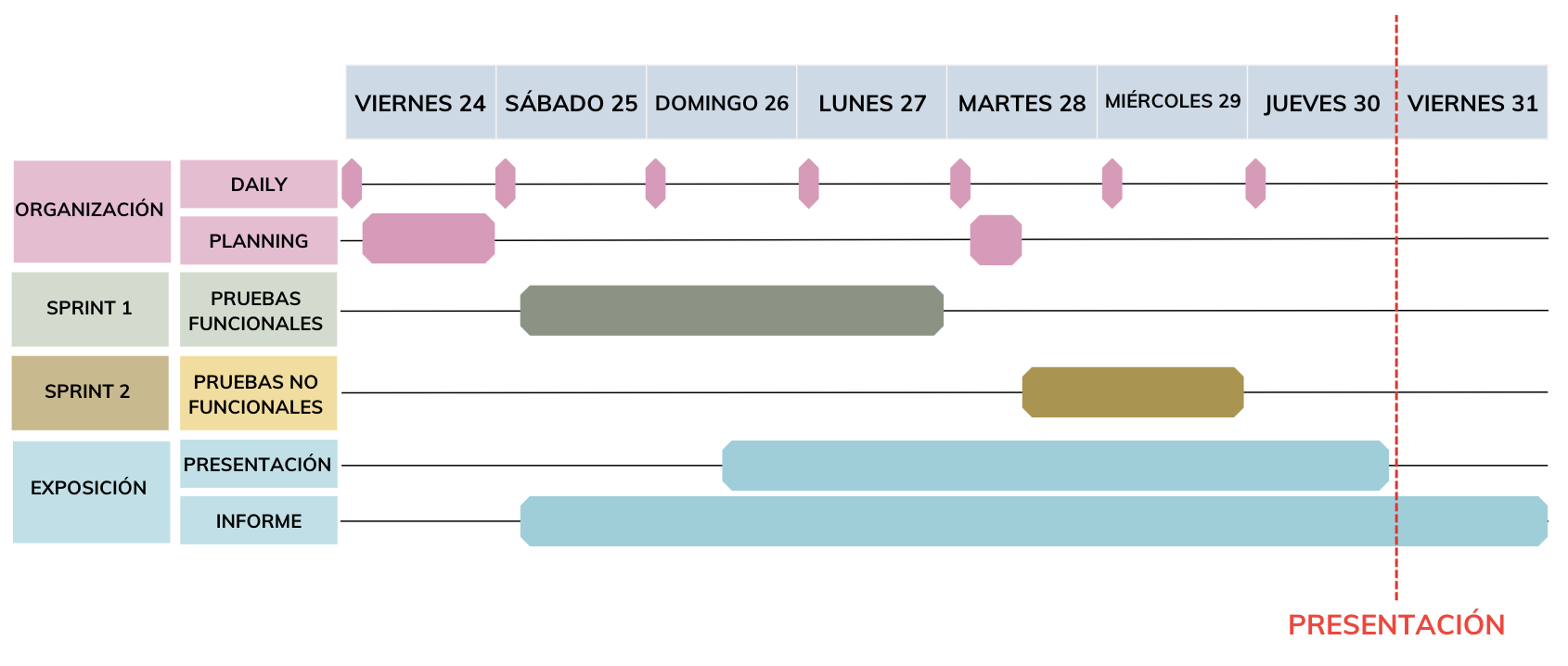
[Propuesta de Mockup 19](#_4vqk9ih2906w)

[**Accesibilidad 20**](#_j0uz2nfq6ri7)

[**Conclusión 21**](#_c58aqcx7ro1l)

## Organización del equipo

### Diagrama de Gantt



A la hora de compartir cómo hemos organizado el trabajo dentro del equipo, decidimos valernos de un Diagrama de Gantt, ya que expresa gráficamente las diferentes instancias que transitamos para conseguir el objetivo final.

Para coordinar el trabajo a realizar, hemos llevado adelante Dailies donde los integrantes compartimos nuestros avances, bloqueos y los objetivos para el día. También hemos efectuado una Planning para definir y organizar el trabajo antes del inicio de cada Sprint. Todos estos componentes conforman la metodología Scrum que llevamos a cabo.

En el Sprint 1 hemos efectuado todas las pruebas funcionales que se encontraban relacionadas con los flujos principales que enunciaremos a continuación con la presentación del mapa del sitio.

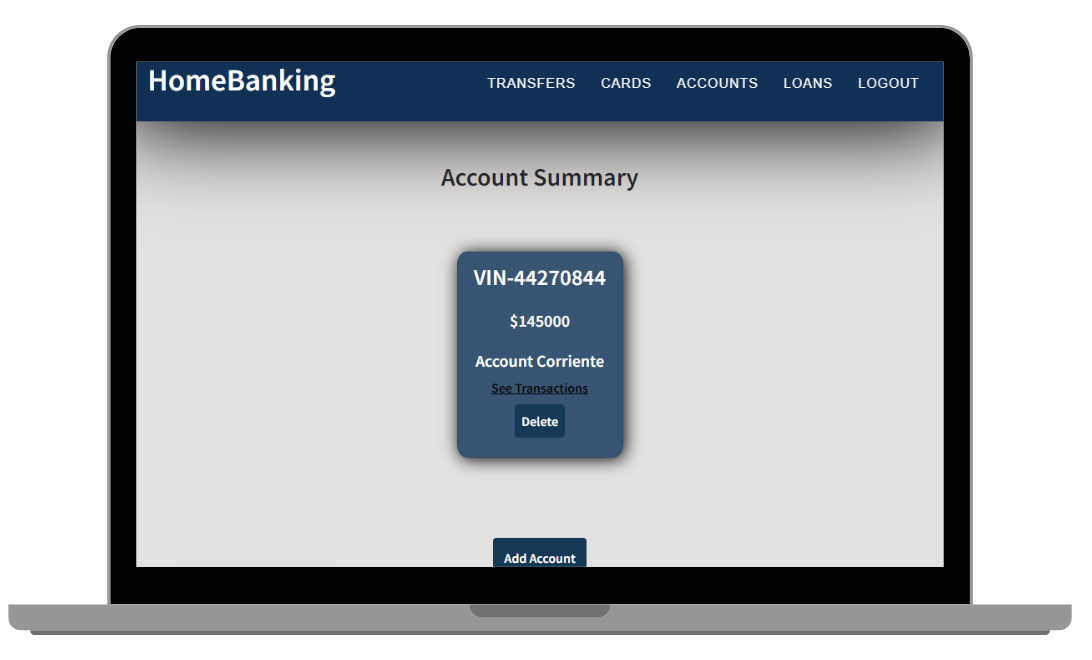
En el Sprint 2 hemos realizado todas las pruebas no funcionales que consideramos esenciales para intentar acercarnos al objetivo de asegurar la calidad del producto analizado. Hemos realizado pruebas de: APIs, Performance, Automation, Seguridad, VPN, Accesibilidad, UX/UI y recomendaciones sobre las BBDD.

A la par de la ejecución del trabajo, hemos desarrollado el presente informe y la presentación que, oportunamente, ha podido visualizarse en la exposición del análisis del producto.

Para llevar adelante la tarea solicitada hemos requerido utilizar una vasta cantidad de herramientas. A continuación listaremos los logos de las principales para que el lector pueda comprender cómo se han efectuado los diferentes análisis que se observarán en el trabajo realizado. Volveremos a cada una de las herramientas cuando sea oportuno a lo largo del informe.



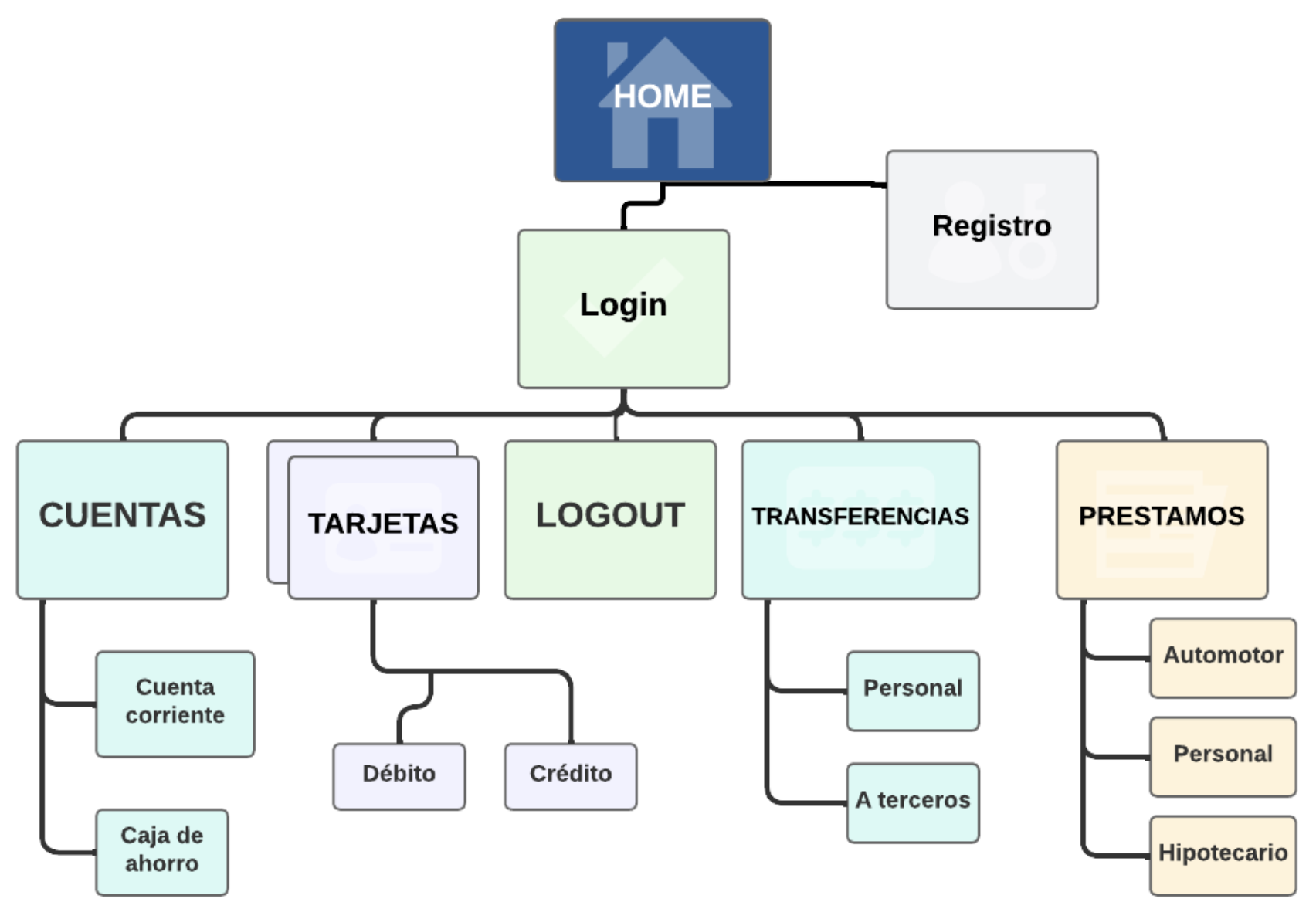
### El producto



### Épica

“Nos gustaría que construyan una aplicación que permita realizar operaciones bancarias desde cualquier dispositivo, pc, teléfono, tablet a través de una aplicación web. Si la aplicación web tiene éxito podríamos pensar en crear una aplicación móvil que utiliza los mismos servicios.”

### Mapa del sitio



Como puede observarse en el mapa del sitio que hemos desarrollado, los flujos principales del Home Banking son: Registro, Login, Cuentas, Tarjetas y Préstamos.

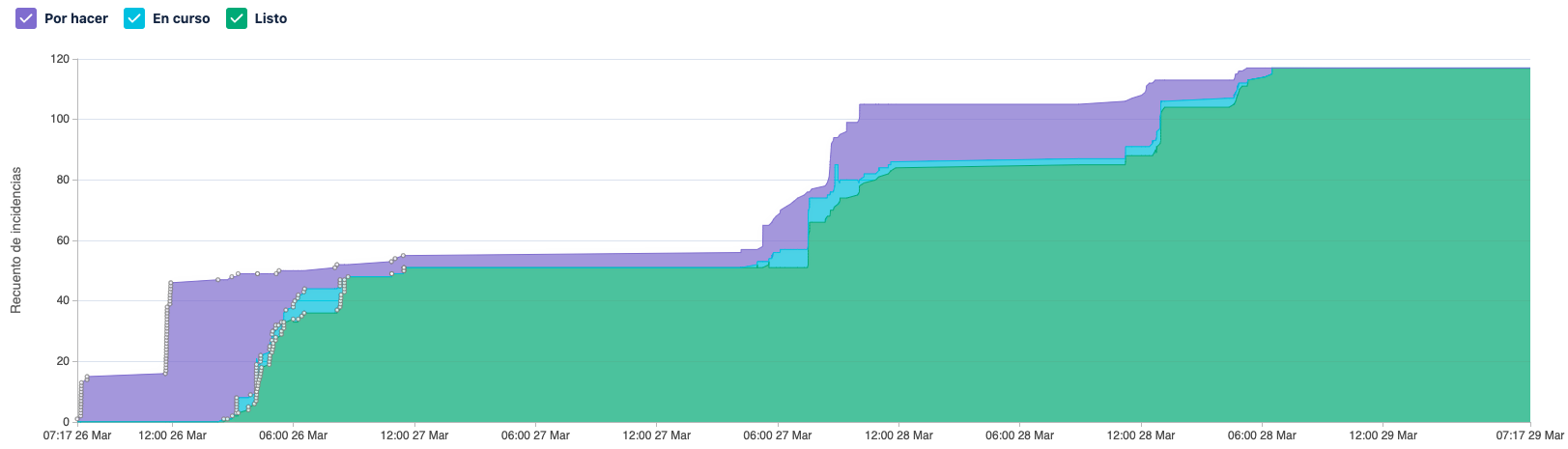
Registro y Login pueden entenderse como funcionalidades principales ya que son condición necesaria para acceder al resto de las funcionalidades que ostenta el sistema.

Por su parte, la administración, manipulación, visualización y eliminación de Cuentas y Tarjetas se comportan como flujos principales de este sistema.

Por último, la solicitud, visualización y ejecución de los diferentes tipos de Préstamos y Transferencias se consolidan como una parte fundamental del sistema, ya que hacen a la circulación del dinero dentro de la aplicación web analizada.

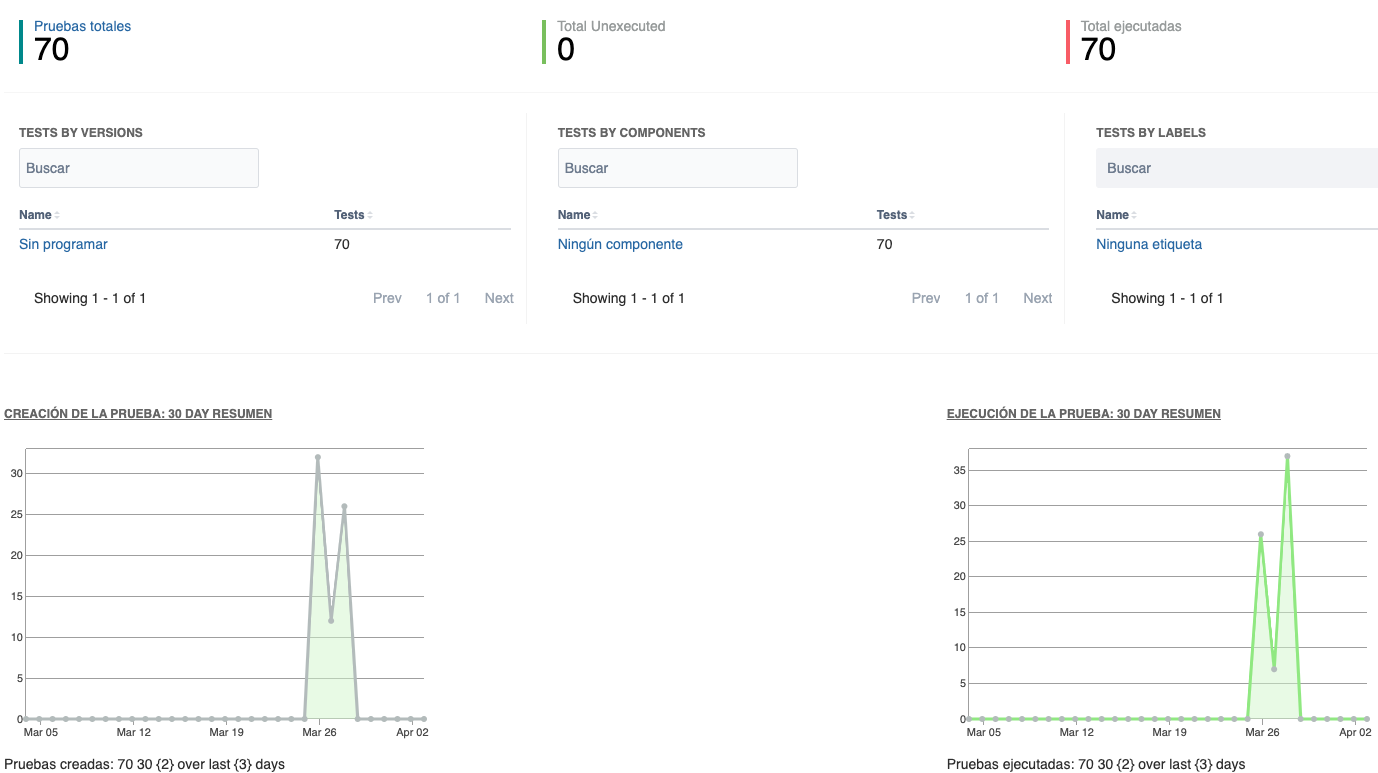
Reportes generales de Jira

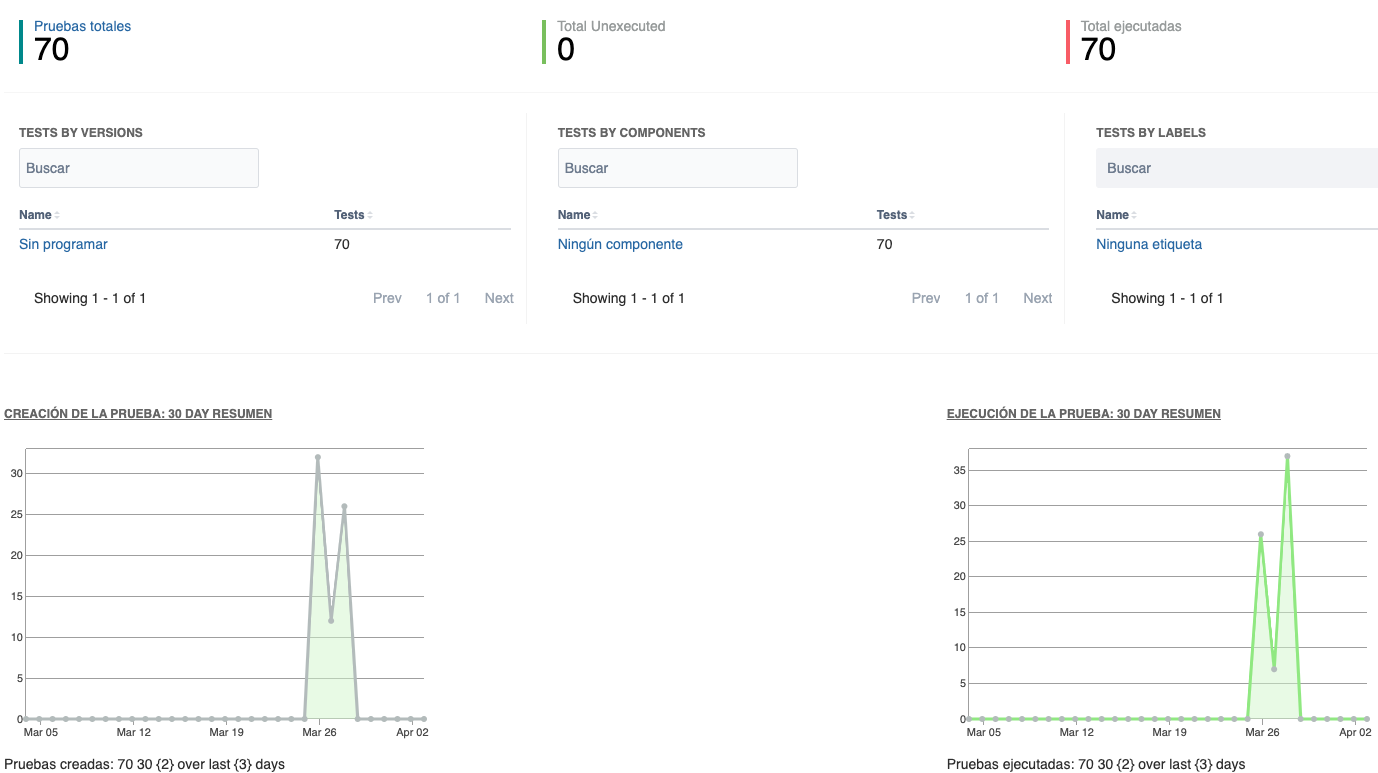
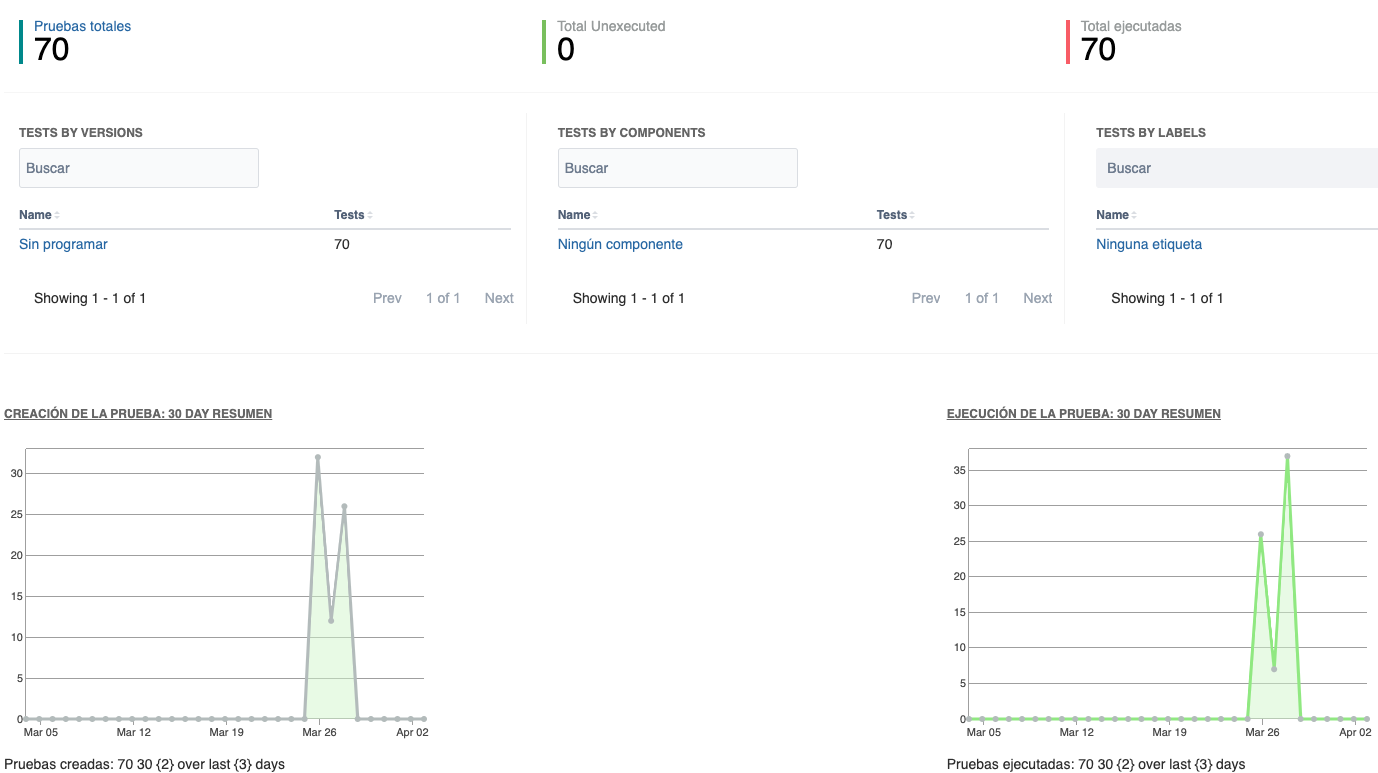
### Diagrama de flujo acumulado



\*Se tomó el criterio de finalizar los Bugs en proceso a modo de cierre del proyecto en Jira, pero en una situación laboral esta instancia debe continuar hasta que los desarrolladores solucionen los defectos y fallos y asignen los casos para ser testeados nuevamente.

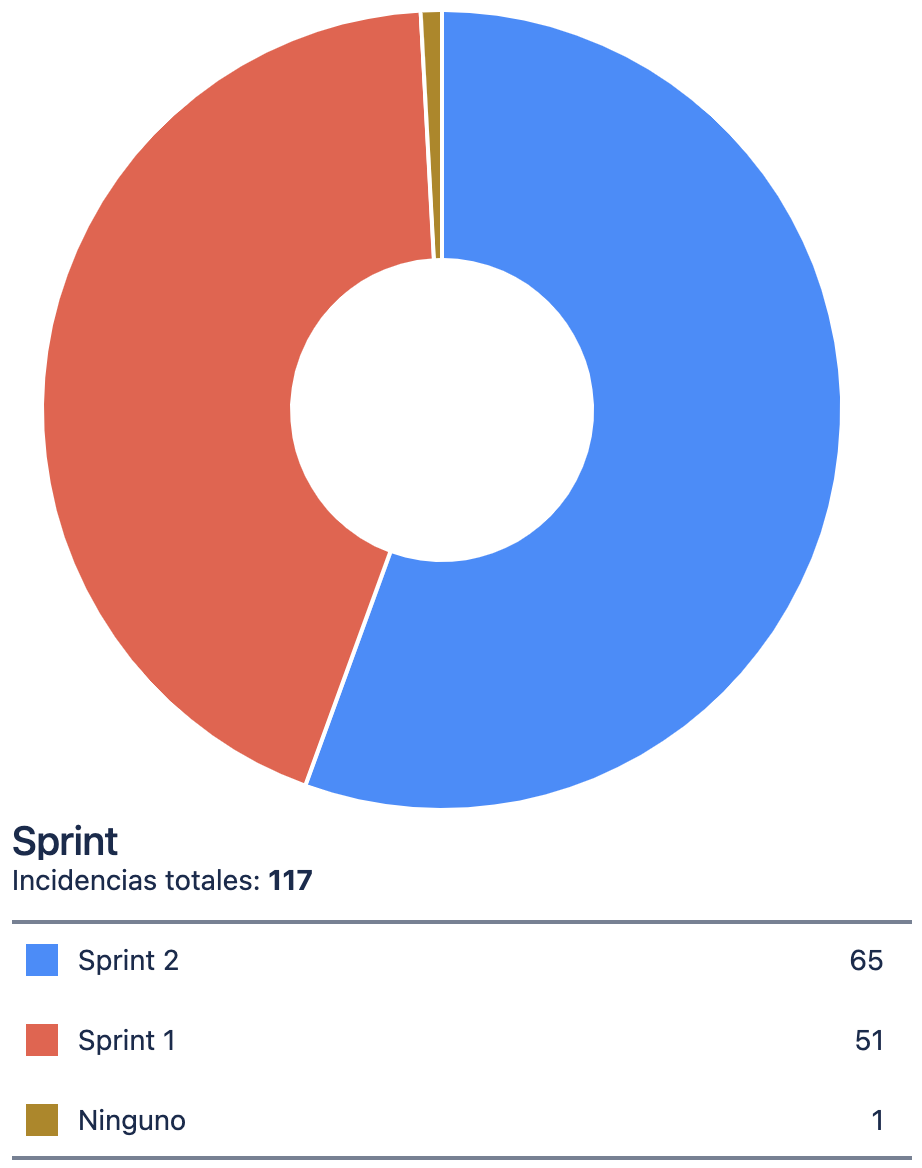
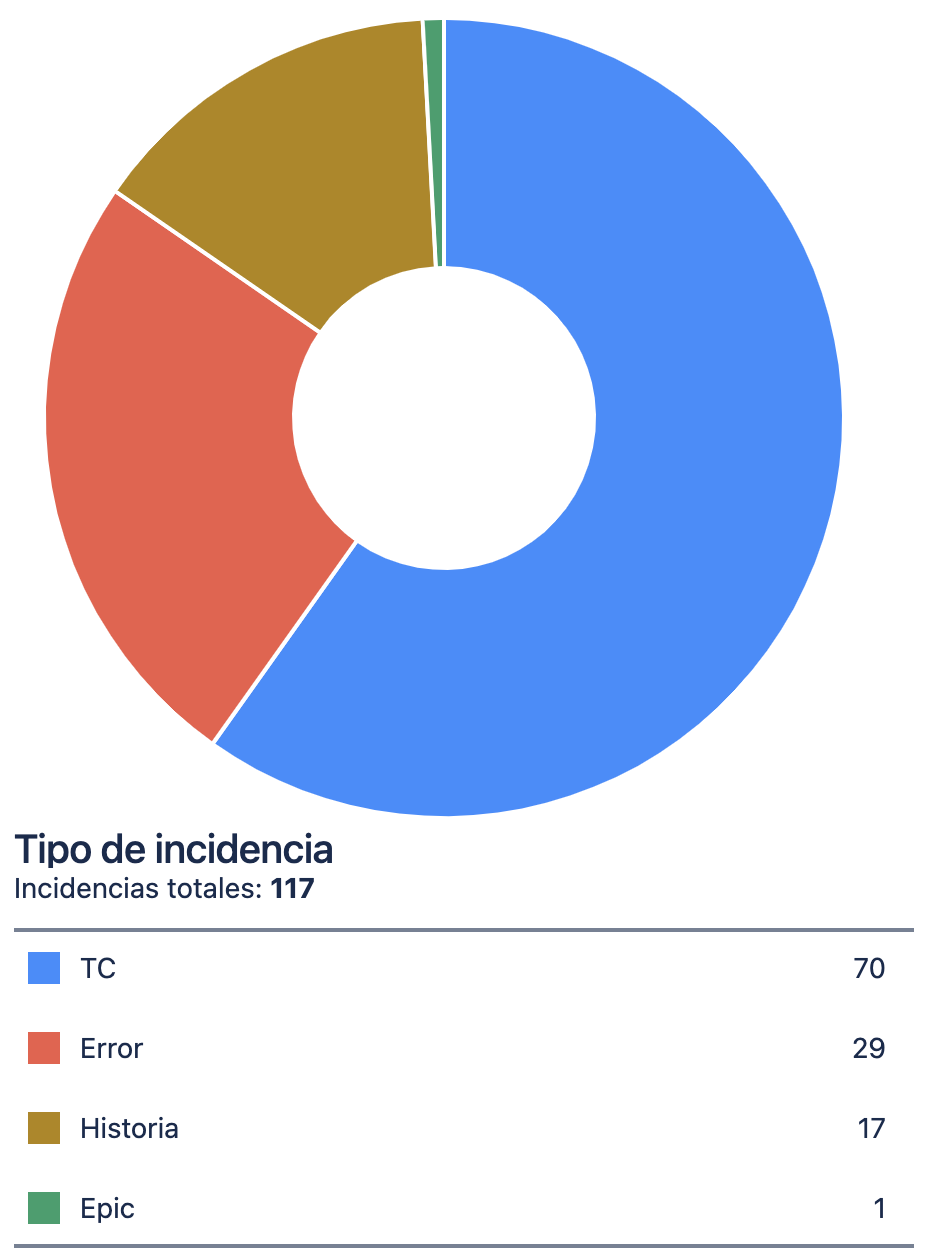
### Ejecuciones totales de casos de prueba

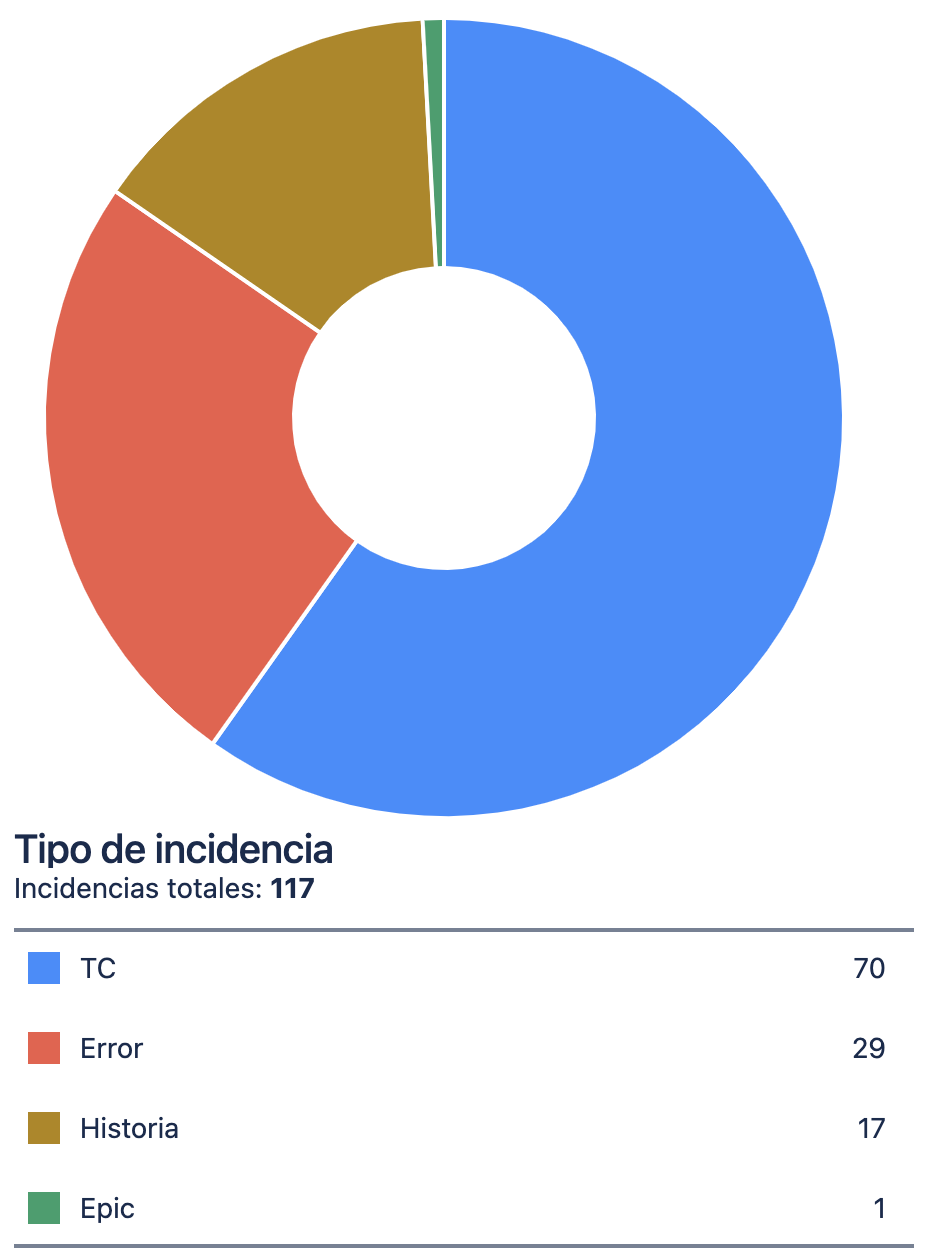




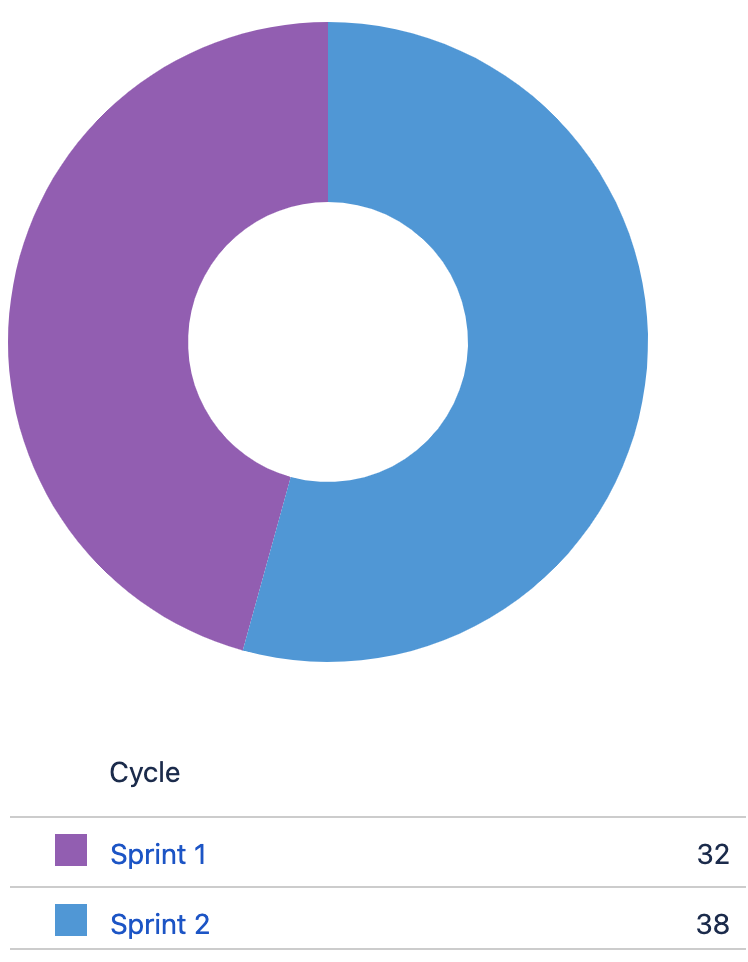
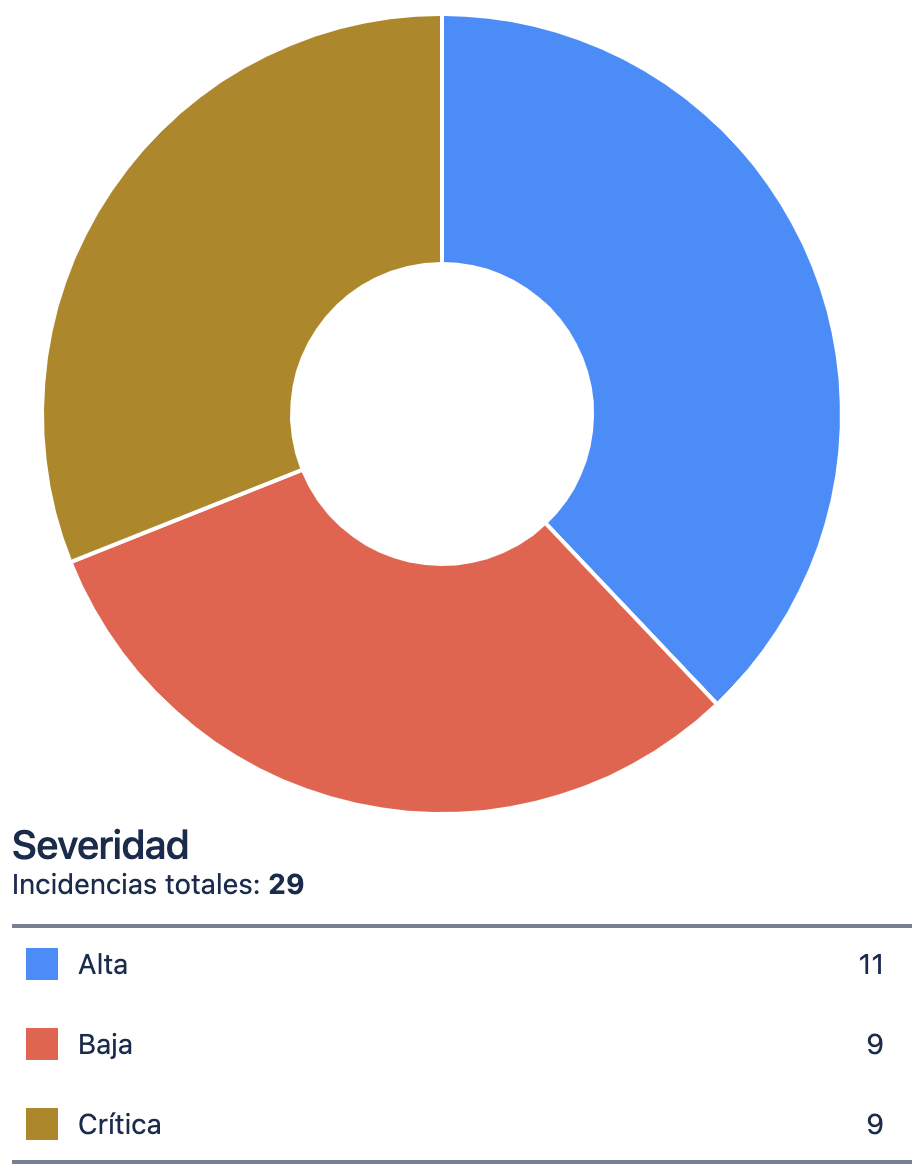
## Desarrollo del trabajo

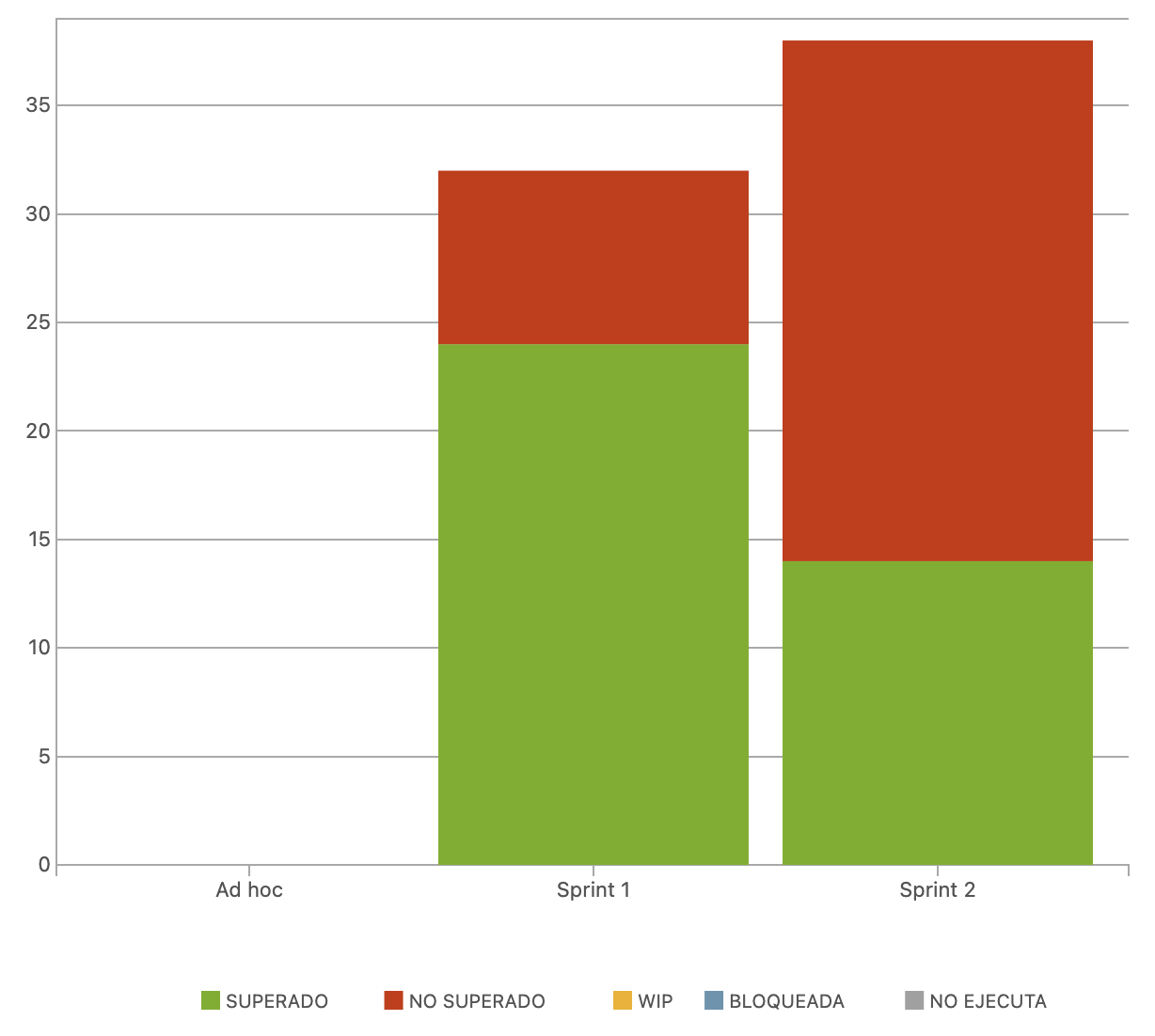
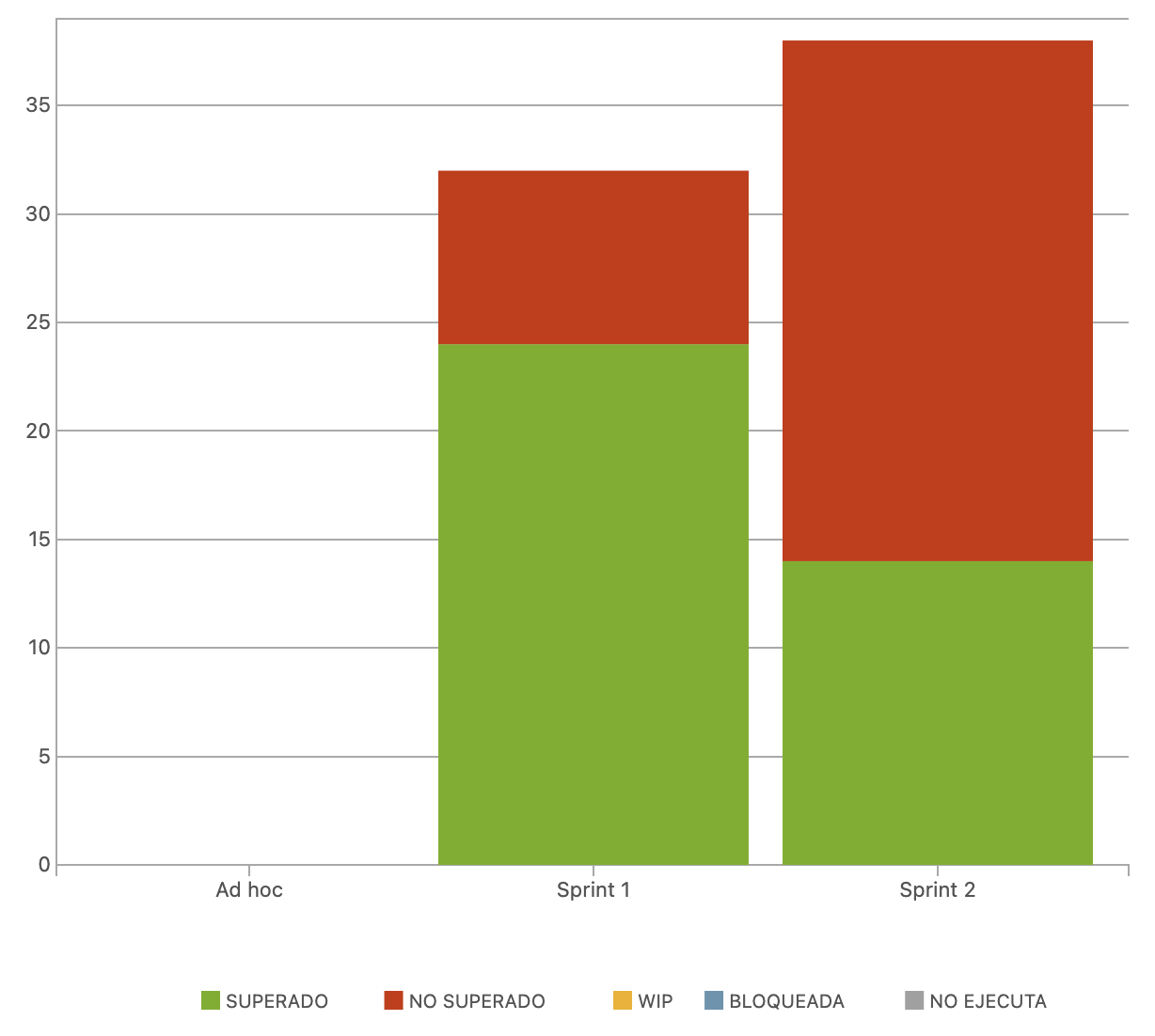
### Cantidad total de incidencias



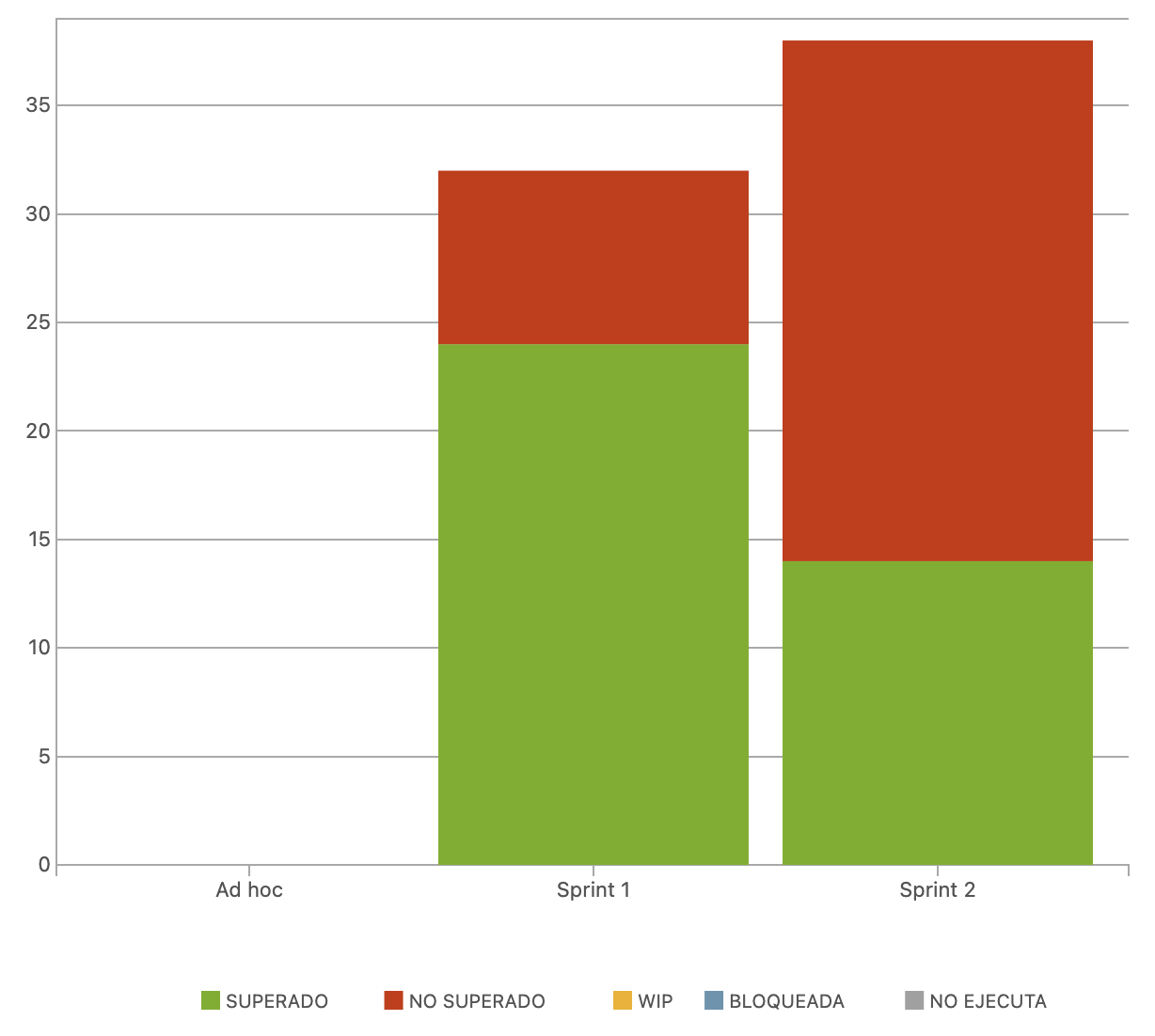


Casos de pruebas y bugs:



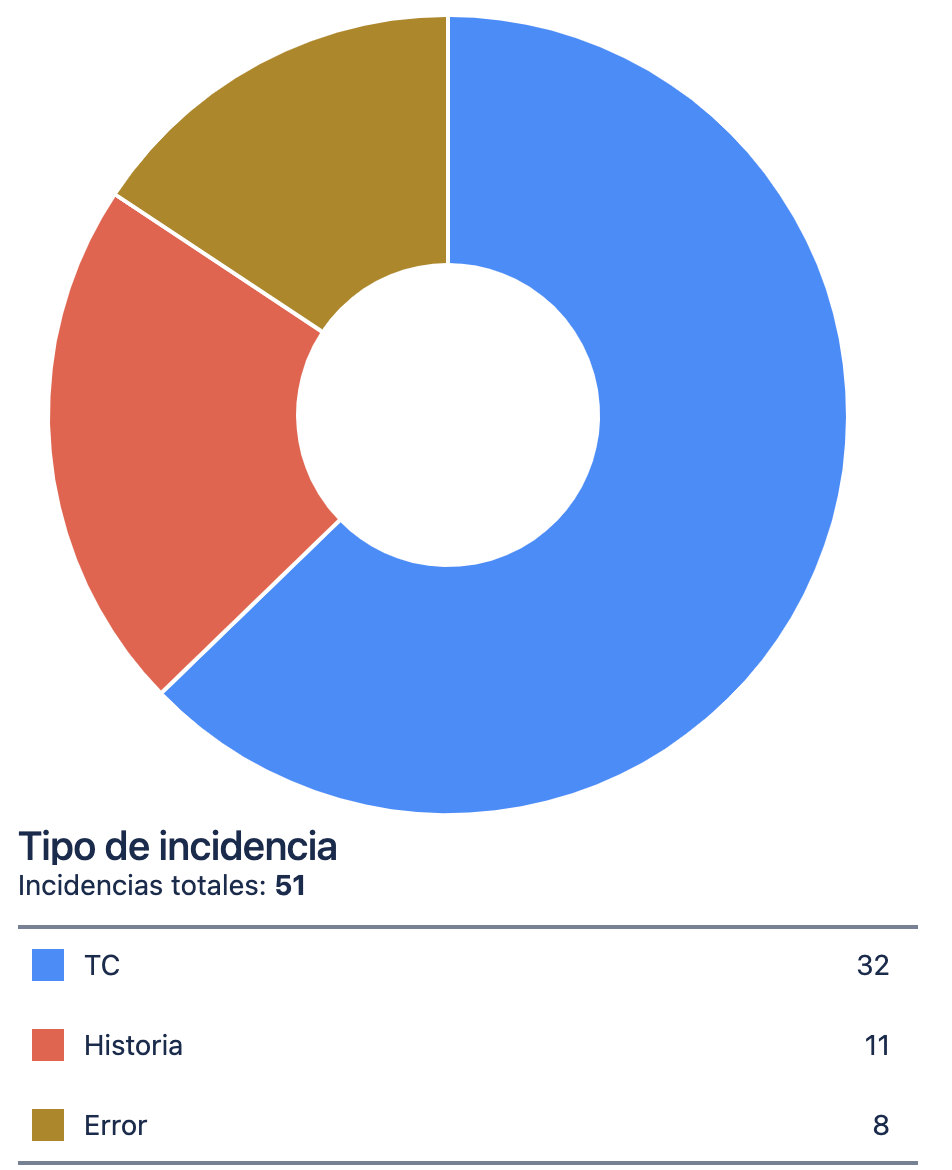




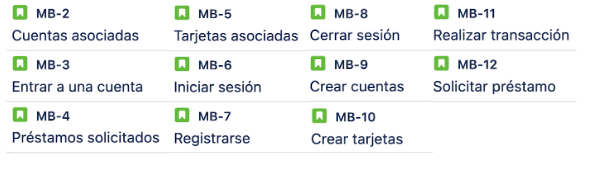


### Sprint 1

Durante el sprint 1 el equipo se centró en la implementación de las funcionalidades básicas de la plataforma. El objetivo principal del sprint.asegurar que todas las funcionalidades básicas se implementaron correctamente y se probaron exhaustivamente antes de continuar con las funcionalidades más avanzadas

Este gráfico puede ayudar a visualizar la relación entre las Historias de usuario y los tipos de incidencia que se encontraron durante el sprint 1.

Historias de usuario:



Severidad de los bugs:



### Sprint 2

En el sprint 2 nos enfocamos en continuar mejorando la plataforma mediante la implementación de nuevas funcionalidades y mejoras en la experiencia del usuario. El objetivo principal fue asegurar que la plataforma sea más segura, estable y fácil de usar para los clientes. A continuación se presentan los detalles del trabajo realizado durante el sprint 2, así como los resultados de las pruebas realizadas y las recomendaciones.

### 

Este gráfico puede ayudar a visualizar la relación entre las Historias de usuario y los tipos de incidencia que se encontraron durante el sprint 2.

Además puede observarse en relación al sprint 1, tienen casi la misma cantidad de TC pero en este caso podemos observar casi el doble de bugs

Historias de usuario:



Severidad de los bugs:



## Seguridad

Para lo que respecta a este tema, podemos decir que es el pilar más importante de la aplicación, al considerar que un banco lo primero que debe ofrecer (históricamente) es la seguridad del dinero y transmitir confianza a sus clientes.

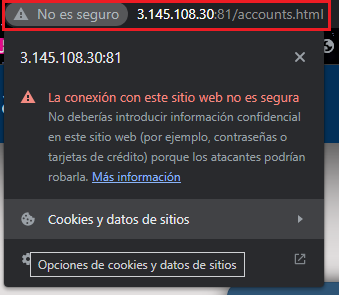
Los casos de prueba propuestos son cinco principalmente y van desde el Test Case 101 al 105:

* Validar dominio de página
* Verificar confirmación de contraseña
* Verificar alerta antes de eliminar cuenta
* Verificar longitud mínima de contraseña
* Verificar mensaje de confirmación al transferir.

Cuatro de estas cinco pruebas ejecutadas han derivado en una falla de seguridad, enumeradas como MB:107, 108, 112 y 113, lo cual corresponde a reportes de bugs, descritos de la siguiente manera:

* Falla de seguridad en dominio
* Falla de seguridad en registro
* Falla de seguridad en área cuentas
* Falla de seguridad en password

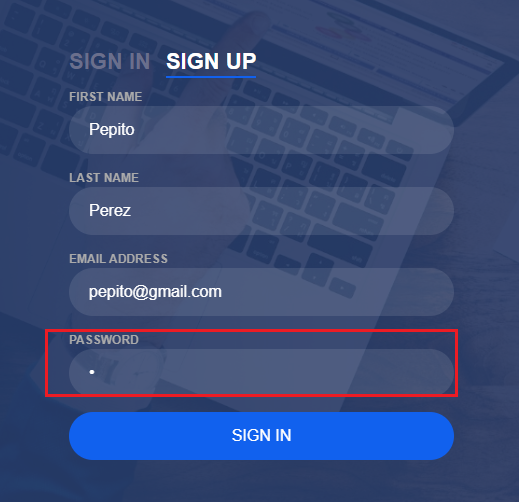
Falla de seguridad en dominio: Se refiere a que el dominio no posee certificado de seguridad (SSL) por lo cual es vulnerable ante robo de datos.



Falla de seguridad en registro: No se solicita confirmación de contraseña a la hora de registrarse

Falla de seguridad en área cuentas: A la hora de eliminar una cuenta no hay ningún tipo de alerta o notificación previa de confirmación. Esto es crítico, y hay que mencionar que en otro caso de prueba se detectó que se pueden eliminar cuentas con dinero dentro, lo cual agrava más a este tipo de error.

Falla de seguridad en password: A la hora de realizar el registro, el campo password no solicita un mínimo de caracteres. Para el caso de prueba se introdujo un carácter y se ha notado esta falla.



## API Testing

### Introducción para el testing de apis

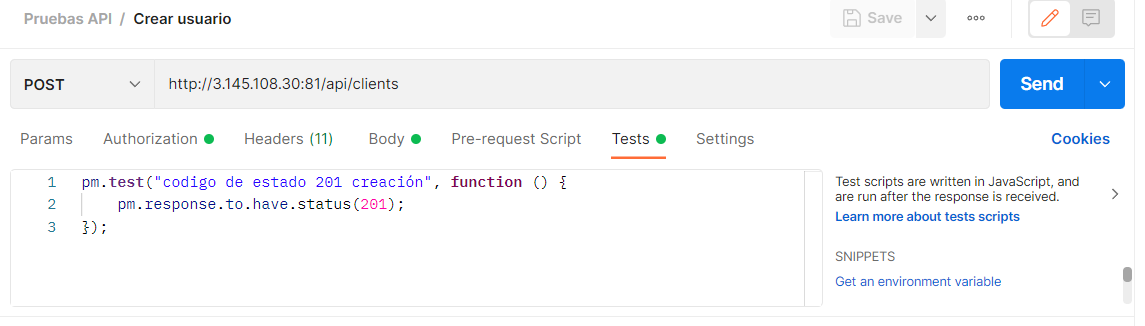
En este caso, se trata de una evaluación de las APIs de una aplicación de Homebanking de un banco ficticio llamado Mindhubbrothers mediante la herramienta Postman.

### Objetivos

El objetivo de la actividad, es evaluar la calidad de las APIs en cuanto a su rendimiento, escalabilidad y capacidad para manejar un alto volumen de peticiones.

### Automatización con Snippets

Para la utilización de Snippets, se tomaron pequeñas partes de código para realizar la automatización del paquete de peticiones, con el código correspondiente al estado de devolución de la petición.



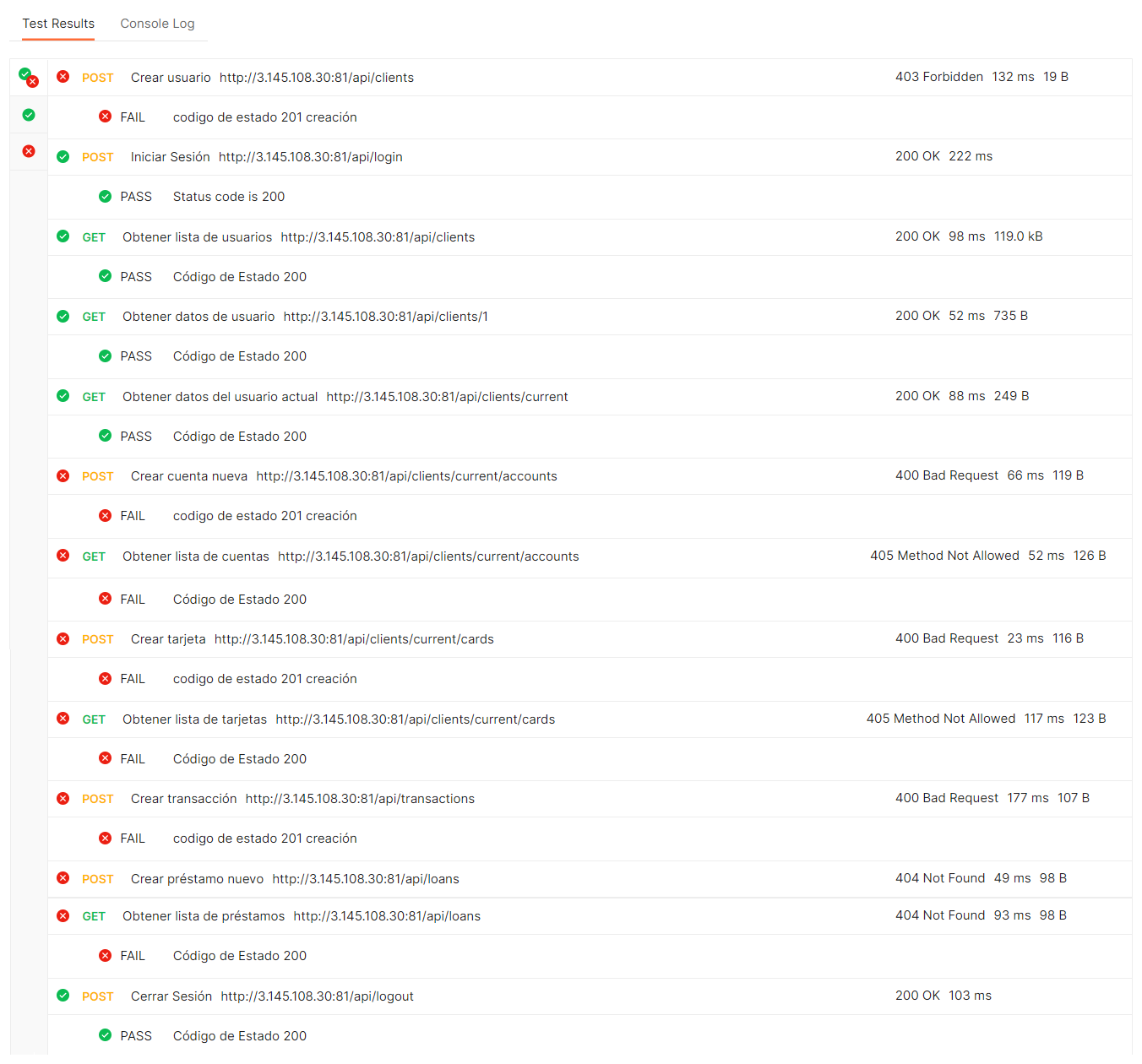
### Resultados obtenidos en postman

Para este caso analizamos ocho (8) endpoints para trece (13) peticiones, con los métodos GET y POST.

Tiempo de respuesta

Se realiza el tiempo de respuesta total de los paquetes durante 4 días, ininterrumpidamente. Con los siguientes resultados:

* El tiempo de respuesta del total de paquetes de peticiones es de 1,27 seg.
* Tiempo de respuesta por petición: 97ms

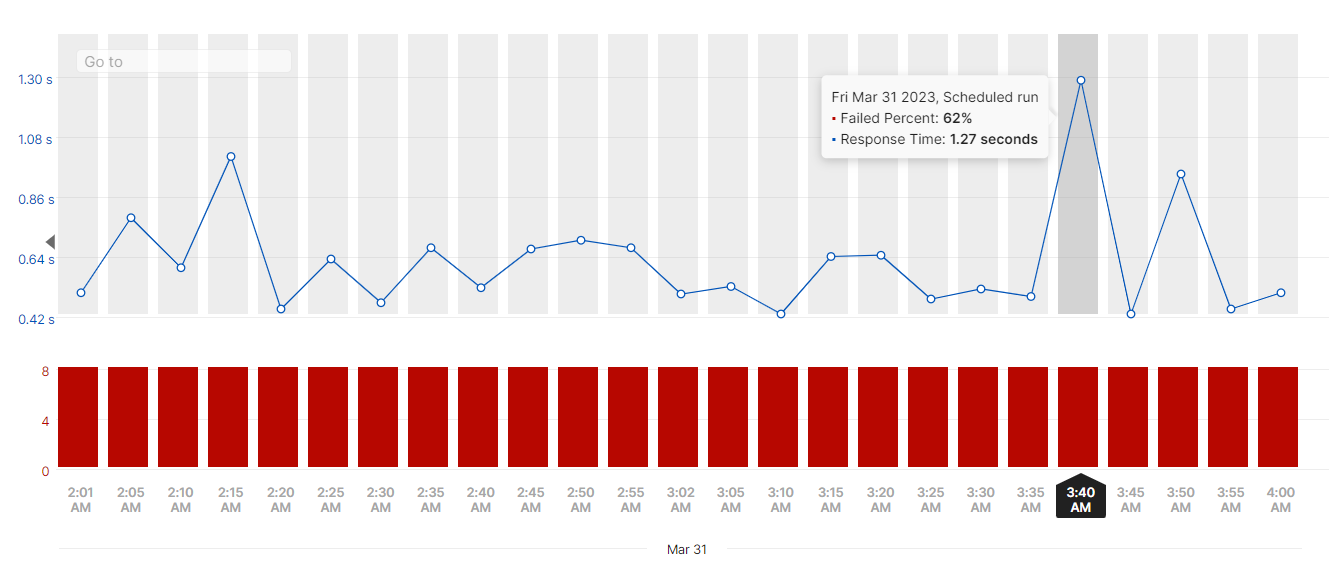


Escalabilidad:

El sistema permite sostener el sistema, con cantidades moderadas de peticiones cada 5 minutos de manera ininterrumpida. Por lo que el sistema no sufre cambios ni caídas bruscas durante ese lapso de tiempo.

Errores y fallos:

Con un 62% de promedio de errores obtenidos de las apis, las cuales no cumplen con lo solicitado por las peticiones.



Conclusiones y recomendaciones de APIs

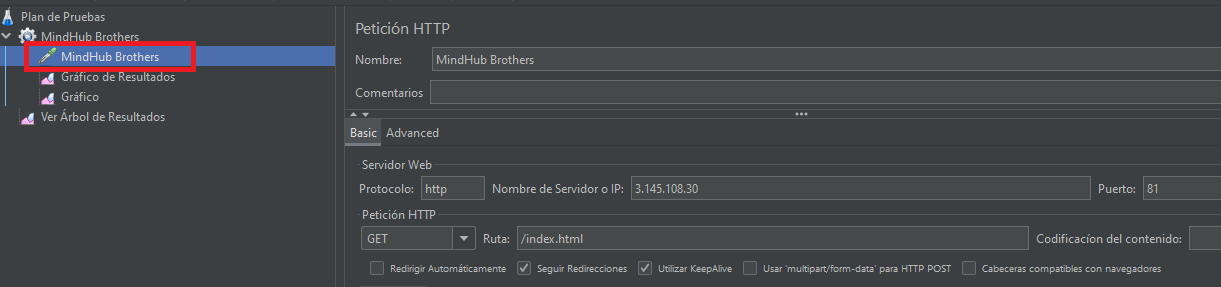
En general, se considera que una aplicación de Home Banking debe responder en menos de 2 segundos para proporcionar una buena experiencia al usuario. Por lo que se puede decir que, en tiempos de respuesta del servidor la aplicación cumple con el tiempo admisible para una aplicación de homebanking. Pero, la API no funciona correctamente ya que devuelve errores en las peticiones por lo que se deben corregir con el equipo de desarrollo pertinente.

## Performance de API Testing

Para poder medir el performance de la página se utilizó JMeter.

Solo se realizaron pruebas con el método GET en cuanto al ingreso a la página, titulada cómo “index”.

Estos son los parámetros se configuraron las pruebas:



Protocolo: Http

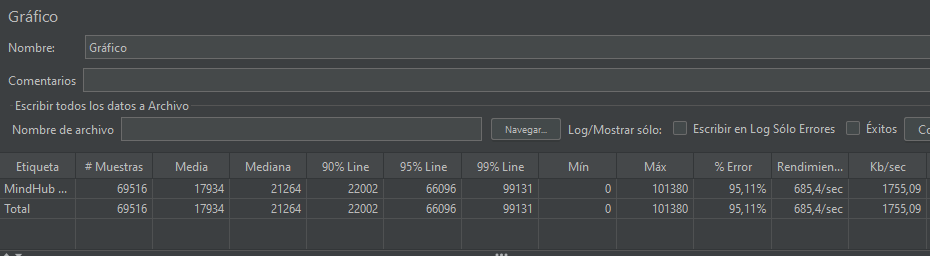
Nombre de Servidor o IP: 3.145.108.30

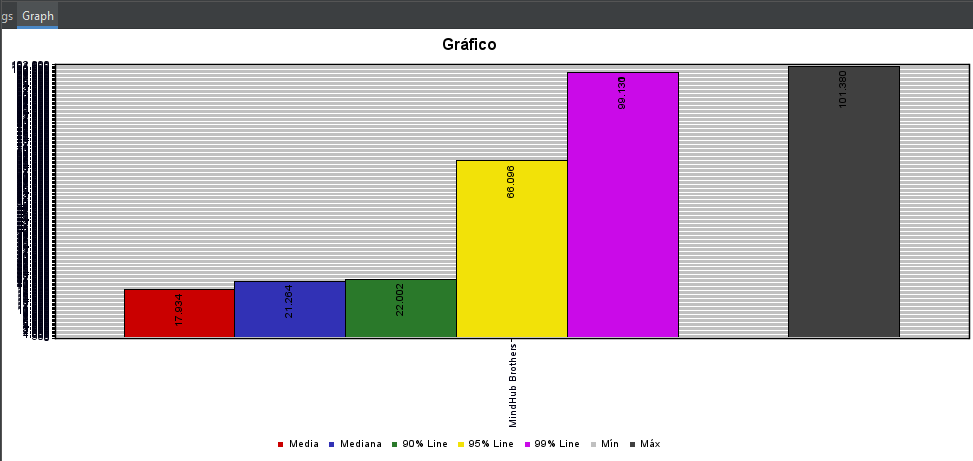
Puerto: 81

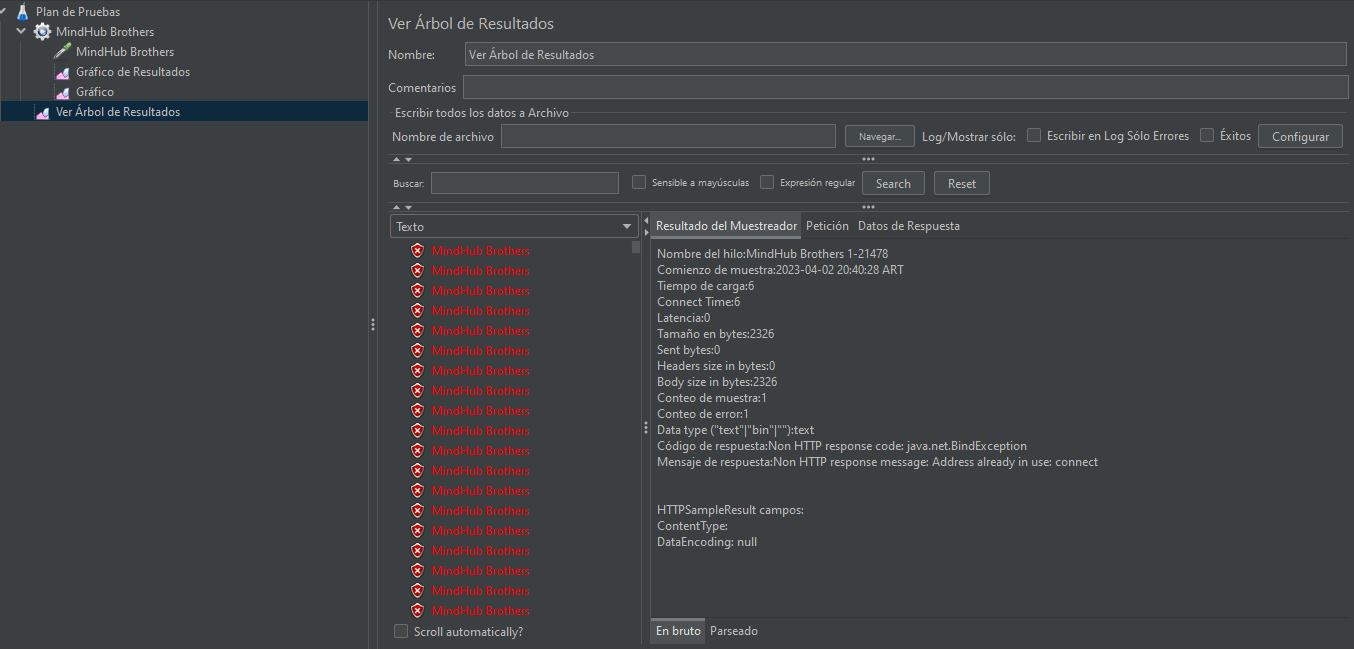
Número de hilos: 200000

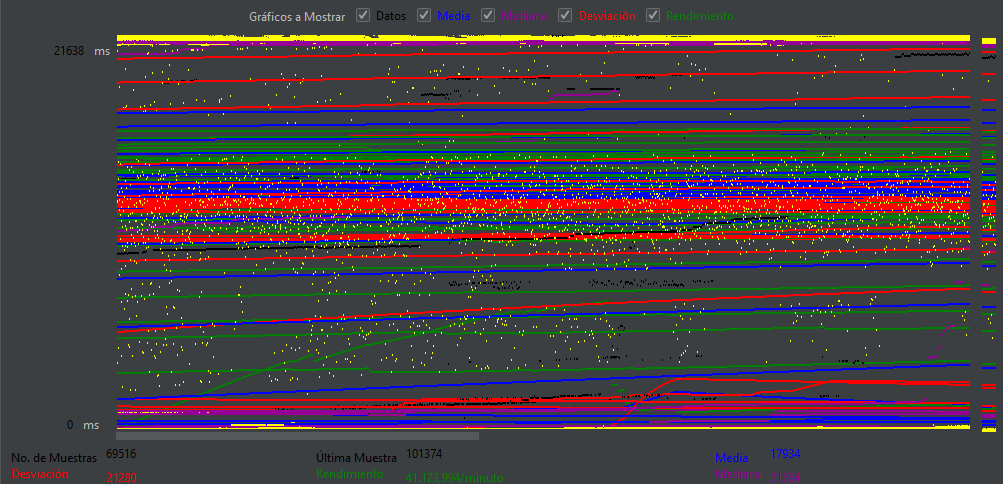
Periodo de subida: 0

Contador del bucle: 1









## Automation

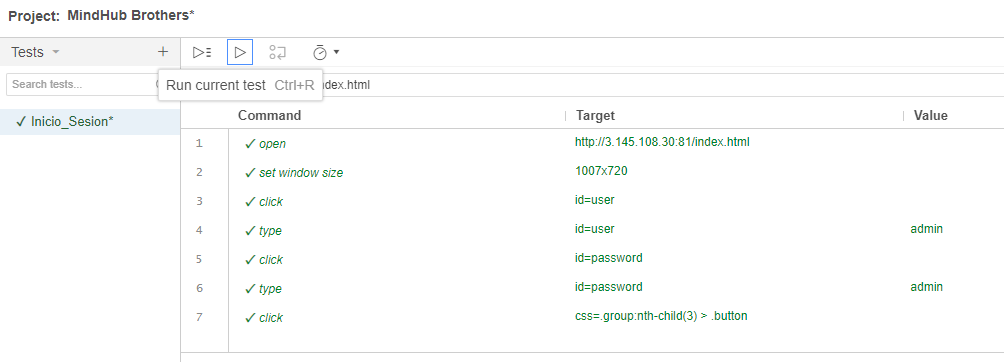
Para las pruebas de automatización se tomaron algunos de los flujos principales, exceptuando el registro. Las mismas se llevaron a cabo en la extensión de Selenium dentro del navegador.

Los casos de prueba propuestos incluían lo siguiente:

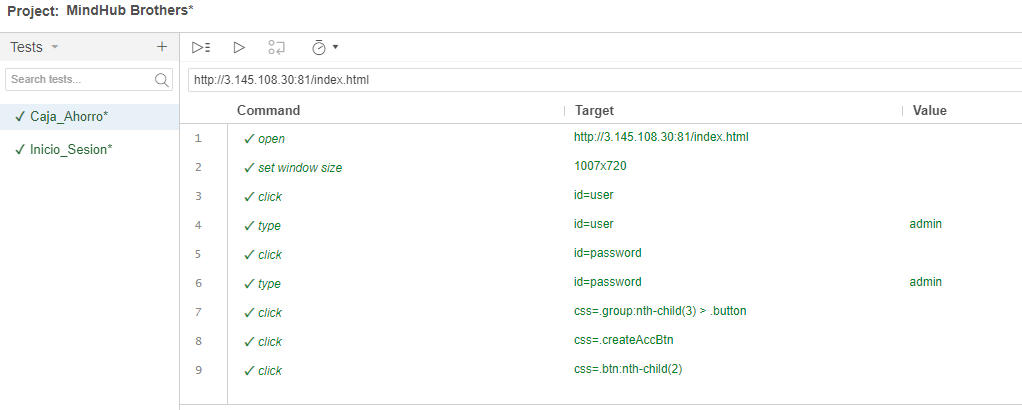
1. Verificación de Inicio de sesión
2. Verificar creación de caja de ahorro
3. Verificación creación de Cuenta Corriente
4. Verificar solicitud de préstamo Personal

Tres de las cuatro pruebas, se superaron con éxito, mientras que la verificación del préstamo personal falló en la última etapa de ejecución (dando así un 94,73% de completado)

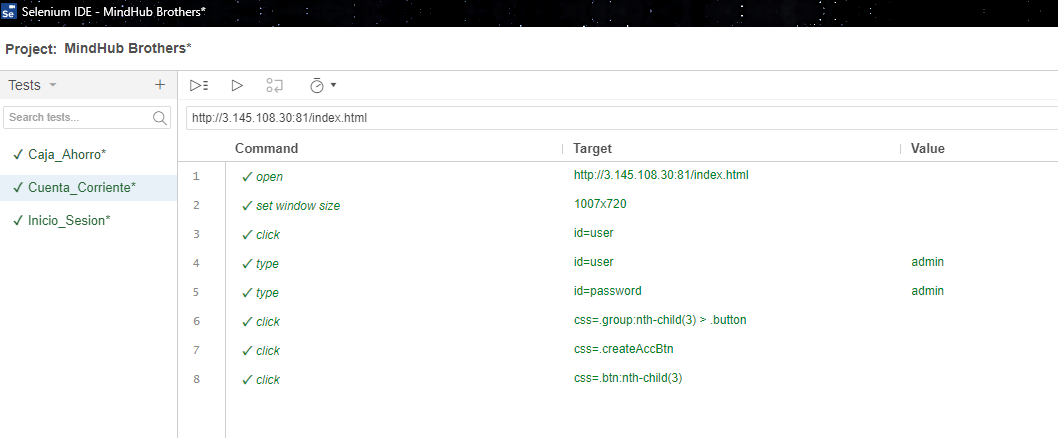
1-



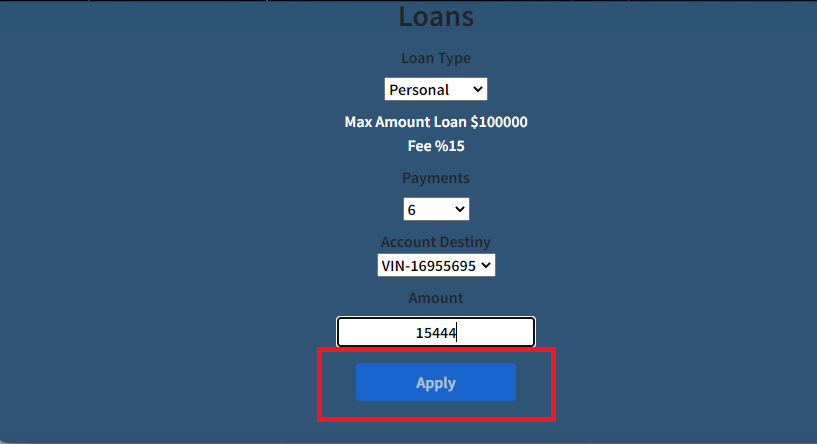
2-

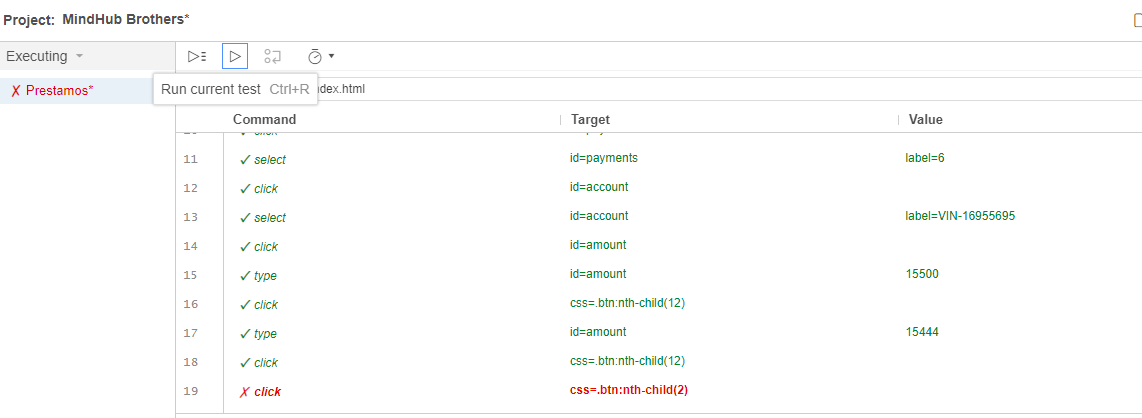


3-



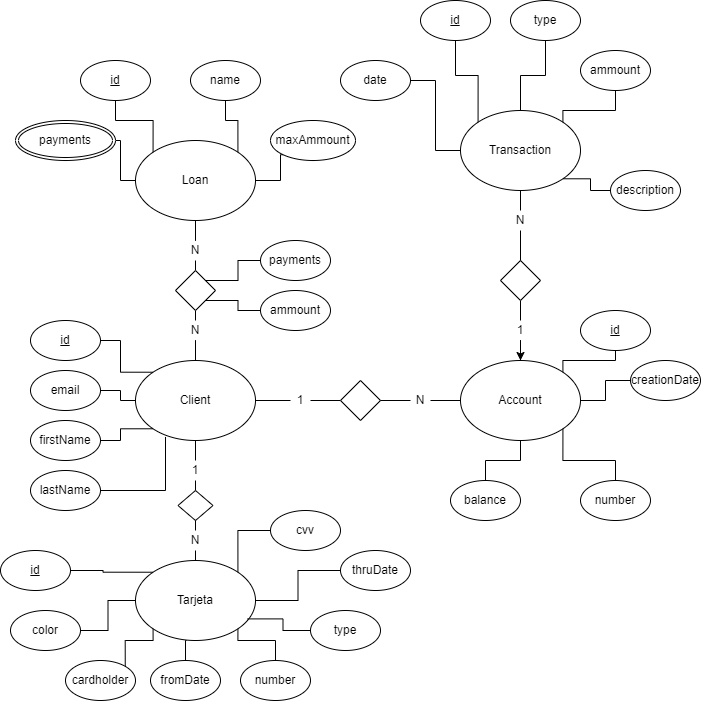
4- El problema observado es que el botón “Apply” queda deshabilitado.





## Bases de Datos

### Diagrama ER original



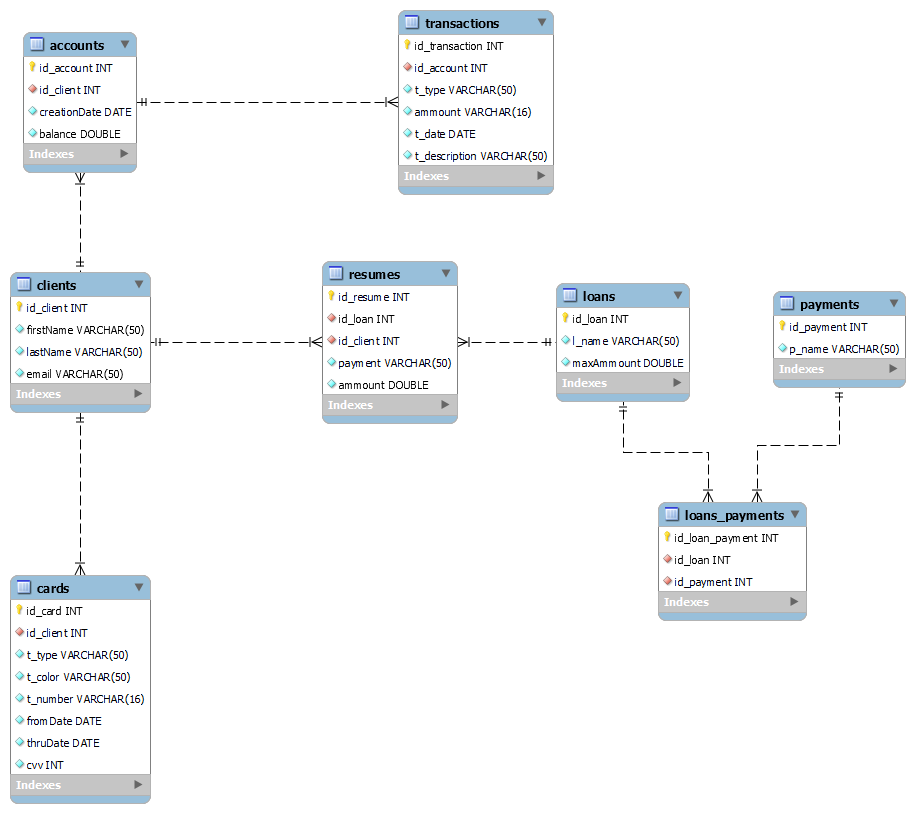
Defectos

1. Cardinalidad N:N : esta clase de relaciones no existe como tal en una base de datos.
2. Valores multivaluados: los campos de este tipo guardan varios datos distintos para una misma entidad lo que genera redundancia a la hora de consultar datos. Esto puede generar una demora en el tiempo de respuesta de las transacciones que se puedan generar.

Soluciones

1. Cardinalidad N:N : para solucionar este defecto se debe generar una nueva tabla que contenga las llaves primarias de las tablas de la relación como llaves foráneas. Esto hará posible el acceso a los datos desde ambas partes de la relación evitando la redundancia de los datos.
2. Valores multivaluados: la solución es generar una nueva tabla que contenga los datos de cada valor del campo y su id, y relacionarla con la tabla a la que estaba vinculada.

### BBDD Propuesta



Cambios

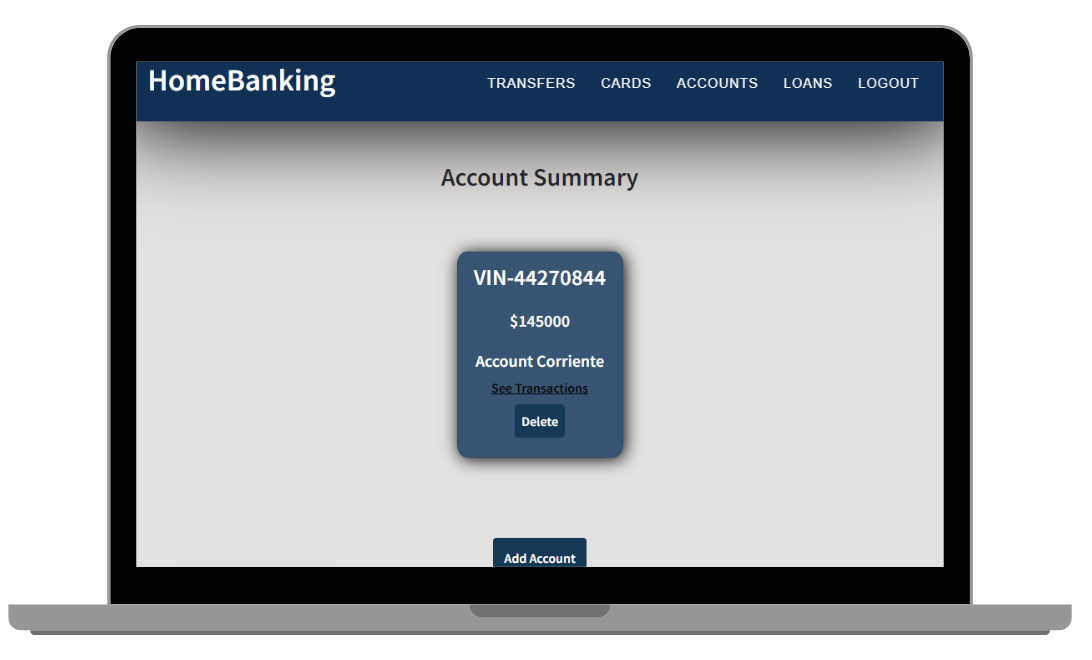
Analizando el diagrama ER original y los posibles defectos de su implementación, proponemos nuestra BBDD implementada en MySQL donde aplicamos las soluciones vistas anteriormente:

1. Para las relaciones N:N : proponemos la creación de las tablas “resumes”. La misma contendrá la llave primaria de las respectivas tablas de la relación como llave foránea, lo que facilitará el acceso a los datos y evitará la redundancia de los mismos.
2. Para los valores multivaluados: proponemos la creación de la tabla “loans\_payments”. Esta guardará la relación N:N que resultará de la descomposición del campo multivaluado. Por otro lado, los distintos valores del campo multivaluado “payments” se verán reflejados en una nueva tabla “payments” con su propio ID.

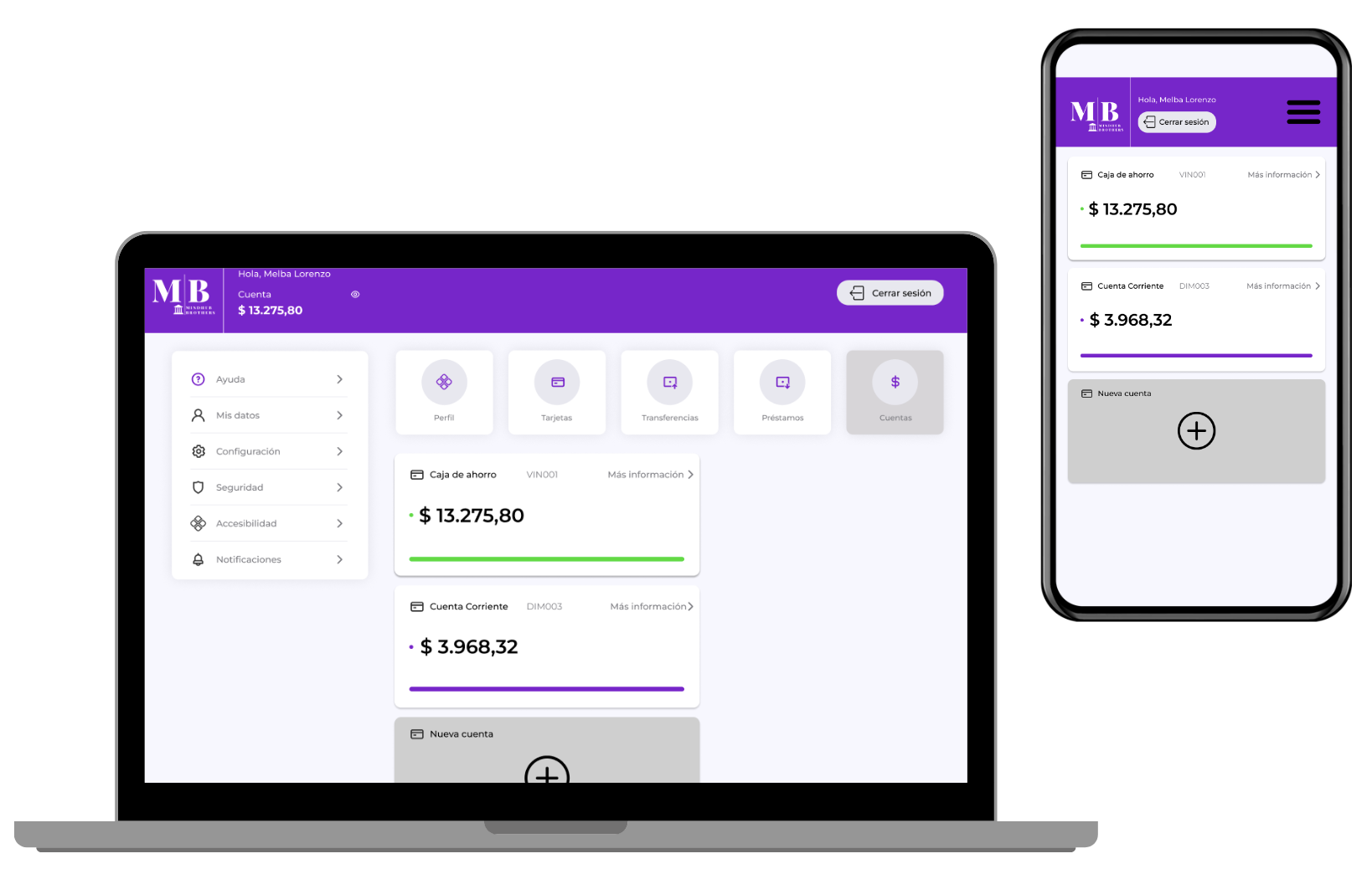
Estos cambios propuestos evitarán la redundancia de los datos, la confusión y facilitarán el acceso a los mismos, además de reducir los tiempos de respuesta de las transacciones.

## UX/UI

### Versión actual



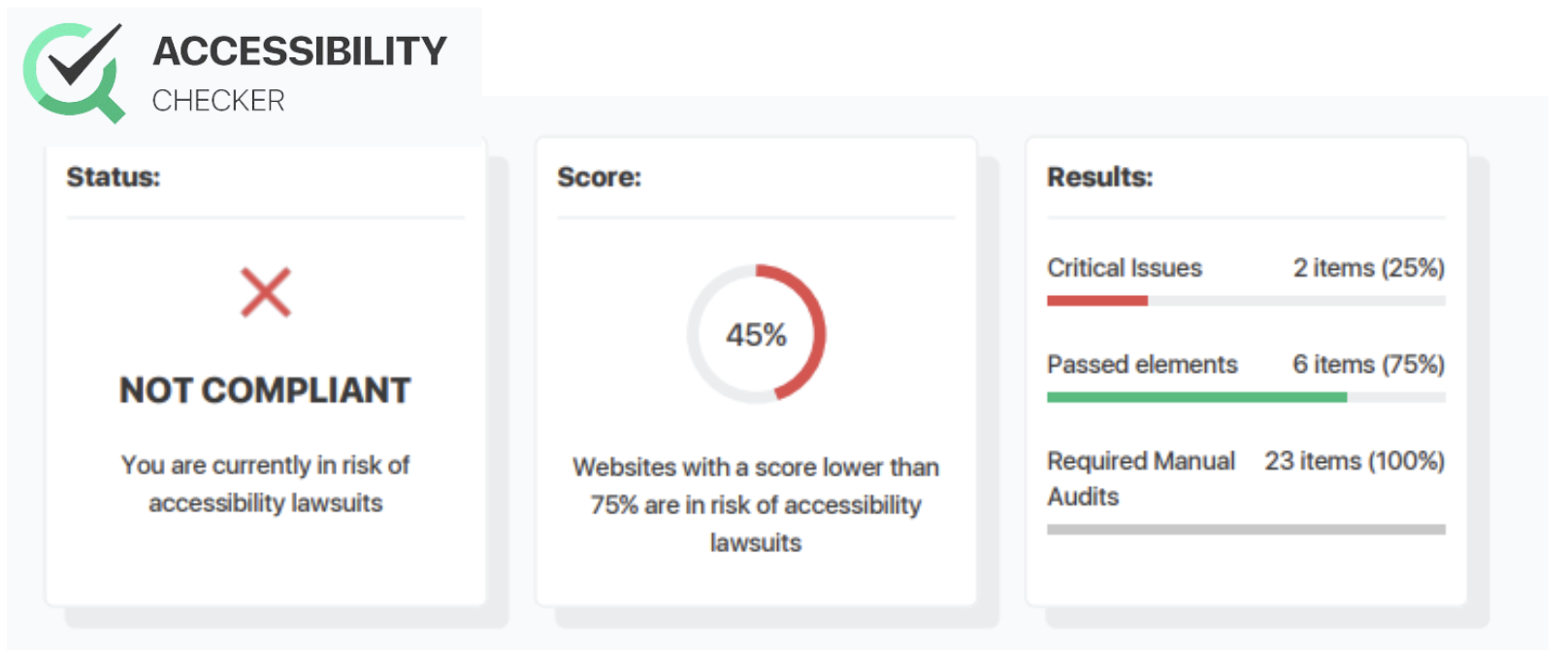
### Propuesta de Mockup



A modo de mejora, se propone intentar conseguir una interfaz más amigable e intuitiva para el usuario, tomando como referencia la oferta de otras entidades que se dedican al mismo rubro y poseen aplicaciones web similares a la desarrollada.

Además, se sugiere la adición de menús y elementos que permitan al usuario la posibilidad de personalizar su experiencia al utilizar el programa. Como puede observarse en el mockup diseñado, sería interesante agregar opciones tales como: Perfil, Configuraciones, Accesibilidad, Seguridad, Ayuda, Mis datos y configuración de Notificaciones.

## Accesibilidad



Hemos realizado pruebas en Accessibility Checker y en la herramienta Talkback para validar la accesibilidad de la página y recibimos información un tanto desalentadora en ese sentido. Se observaron elementos que no deberían encontrarse en ese sitio para una correcta utilización de herramientas de accesibilidad y se evidenció un contraste muy bajo en el texto de la aplicación -lo que limita la visibilidad por parte de personas con discapacidades visuales-.

## Conclusión

Luego de haber realizado un exhaustivo análisis sobre el producto estudiado, podemos inferir que las funcionalidades principales del mismo se encuentran conservadas y la aplicación web cumple con la finalidad para la cual fue desarrollada.

Como se desprende del informe presentado, un 75% de las pruebas realizadas en el Sprint 1 han sido superadas correctamente. Por su parte, únicamente un 44,74% de las pruebas llevadas adelante en el Sprint 2 han resultado satisfactorias. En función de lo mencionado anteriormente, llegamos a la conclusión de que los mayores defectos y fallas se encuentran en lo que atañe a las cuestiones no funcionales del programa.

La seguridad del sistema se encuentra comprometida, ya que el acceso a datos sensibles de los usuarios es irrestricto para terceros, se carece de certificado de seguridad en la página (SSL), se encuentran vulnerabilidades en el registro de los datos al crear una cuenta y la base de datos está mal implementada.

En cuanto a la accesibilidad, se evidencian problemas serios y limitaciones para que las personas con discapacidad puedan hacer uso del programa, ya que la aplicación web no cumple con los estándares de la aplicación Accessibility y no reconoce elementos indispensables para el funcionamiento de herramientas tales como Talkback.

Haciendo referencia a la experiencia de los usuarios, podemos observar que, en relación con otros Home Bankings de entidades bancarias, no se respetaron muchos elementos de la página que contribuyen a satisfacer las necesidades de usabilidad de las personas que interactúan con el programa. Por destacar las más importantes, se evidencia la falta de menús desplegables con opciones de Ayuda, Configuraciones, Perfil, etc.

A modo de conclusión general, resulta menester comunicar que el programa no se encuentra en condiciones de salir a producción si no se analizan y resuelven las problemáticas planteadas en el presente informe.