

Agustín Defferrari



BANK-ito

Proyecto Bank-ito

Introducción

En este proyecto vamos a trabajar sobre una aplicación web llamada “Bank-ito”. Se trata de un homebanking en el cual se pueden realizar operaciones bancarias básicas desde distintos dispositivos. Únicamente se encuentra disponible en el sitio web: <https://homebanking-bankito-production.up.railway.app/web/index.html>

Las funcionalidades principales de la aplicación son:

- Registro;
- Logueo;
- Creación de cuentas;
- Realizar transacciones;
- Solicitar préstamos;
- Solicitar tarjetas;

Metodología de trabajo

Para este proyecto utilizamos metodología *scrum*, comenzando con un sprint planning, donde nos asignamos las tareas, luego realizamos daily meetings por zoom de no más de 15 minutos para organizarnos en el trabajo del día, rotando en la posición de scrum master. Para organizarnos fuera del horario laboral utilizamos las plataformas **Jira**, **discord**, **word** y **excel**(compartidos en Drive) y **whatsapp**.

Al final de cada sprint realizamos un *sprint review* para analizar lo que salió bien, y que se podría mejorar.

Por último realizamos una retrospectiva para ver qué podemos mejorar como equipo para realizar test más completos y permitir entregar un producto de calidad.

Testing:

Para poder organizarnos con el trabajo, en primer momento definimos: la *épica*, las *user story*, y los *test cases* en un archivo .xlsx donde volcamos la información. Luego para poder hacer un seguimiento más exhaustivo del proyecto, empleamos la aplicación **Jira**, donde inicializamos los *Sprints*, y en conjunto con la aplicación **Zephyr Squad** para la organización y ejecución de los *test cases*.

De la información proporcionada y para abordar el análisis de la aplicación, establecimos 3 Sprints, donde evaluamos:

- Sprint-1: Funcionalidad;
- Sprint-2: Accesibilidad - UX/UI;
- Sprint-3: API'S;



Luego de establecer las 11 Historias de usuario, pasamos a determinar y evaluar los casos de prueba.

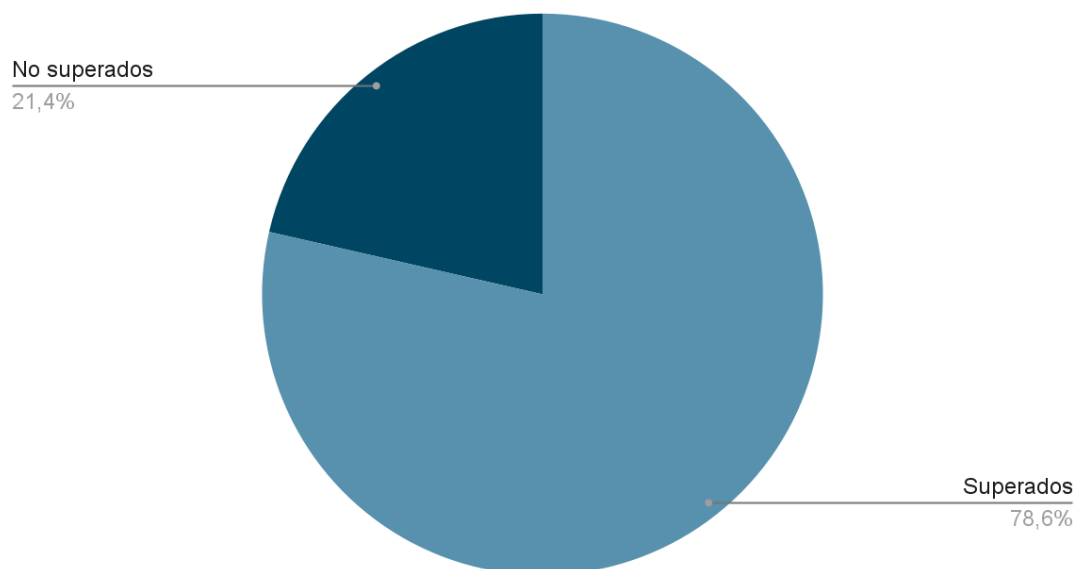
Para el proceso de verificación de los distintos *test cases*, se tomaron como punto de partida diferentes precondiciones establecidas por las historias de usuario y diagramadas previamente por la matriz de casos de prueba, evidenciando cada uno con archivos que luego fueron cargados en Zephyr para su ejecución.

Sprint-1:

→ Casos de prueba: 14

Del 100% de los casos de prueba, un 78.6% fue SUPERADO, mientras que el 21.4% fue NO SUPERADO.

Test cases



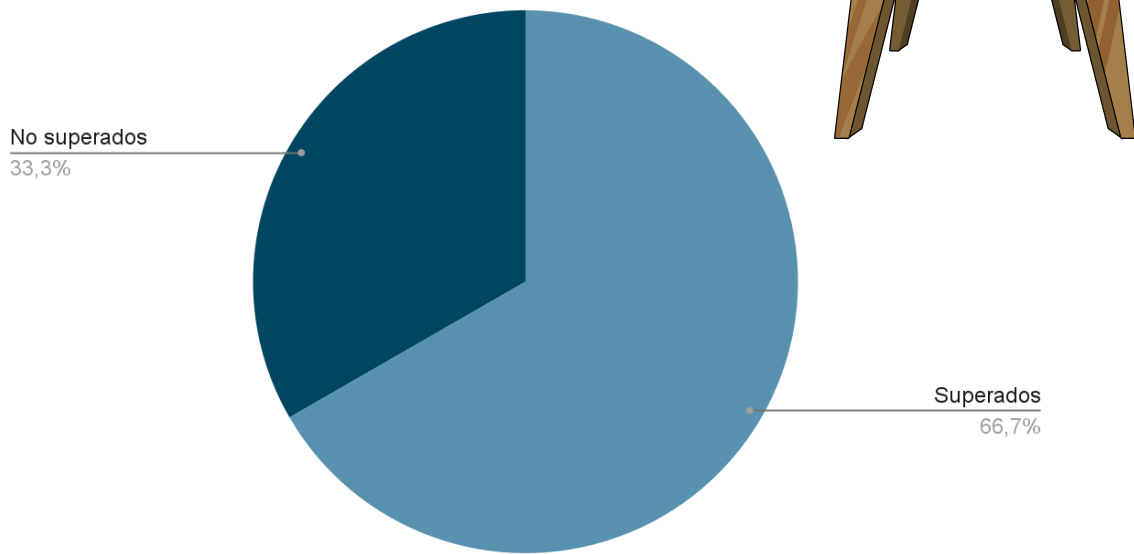
Se probaron manualmente todas las funcionalidades principales del sitio, tomando como punto de partida la información de la matriz de prueba.

Sprint-2:

→ Casos de prueba: 3

Del 100% de los casos de prueba, un 66.7% fue SUPERADO, mientras que el 33.3% fue NO SUPERADO.

Test cases



Para poder realizar estas pruebas empleamos 2 herramientas (extensiones) provistas por la Chrome Web Store:

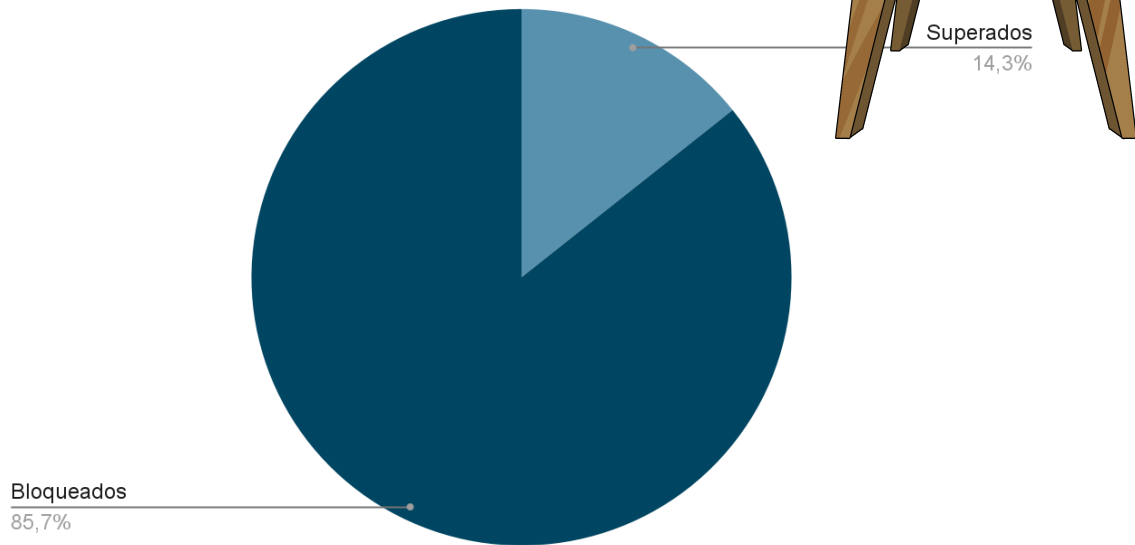
- Selenium IDE: Creímos conveniente el uso de esta extensión con el fin de automatizar el proceso de verificación de la permanencia del *side menu* durante el acceso a los diferentes flujos que poseía el mismo, pudiendo recorrer las distintas funciones del sitio, de manera automática;
- WAVE Evaluation Tool: Esta extensión nos permite visualizar los diferentes errores y recomendaciones que se pueden encontrar en el sitio, respecto a los distintos criterios de accesibilidad, con la opción de mostrar al detalle las diferentes falencias del mismo, incluyendo una opción de mostrar en qué parte del código deberían producirse cambios para poder corregir dichas faltas.

Sprint-3:

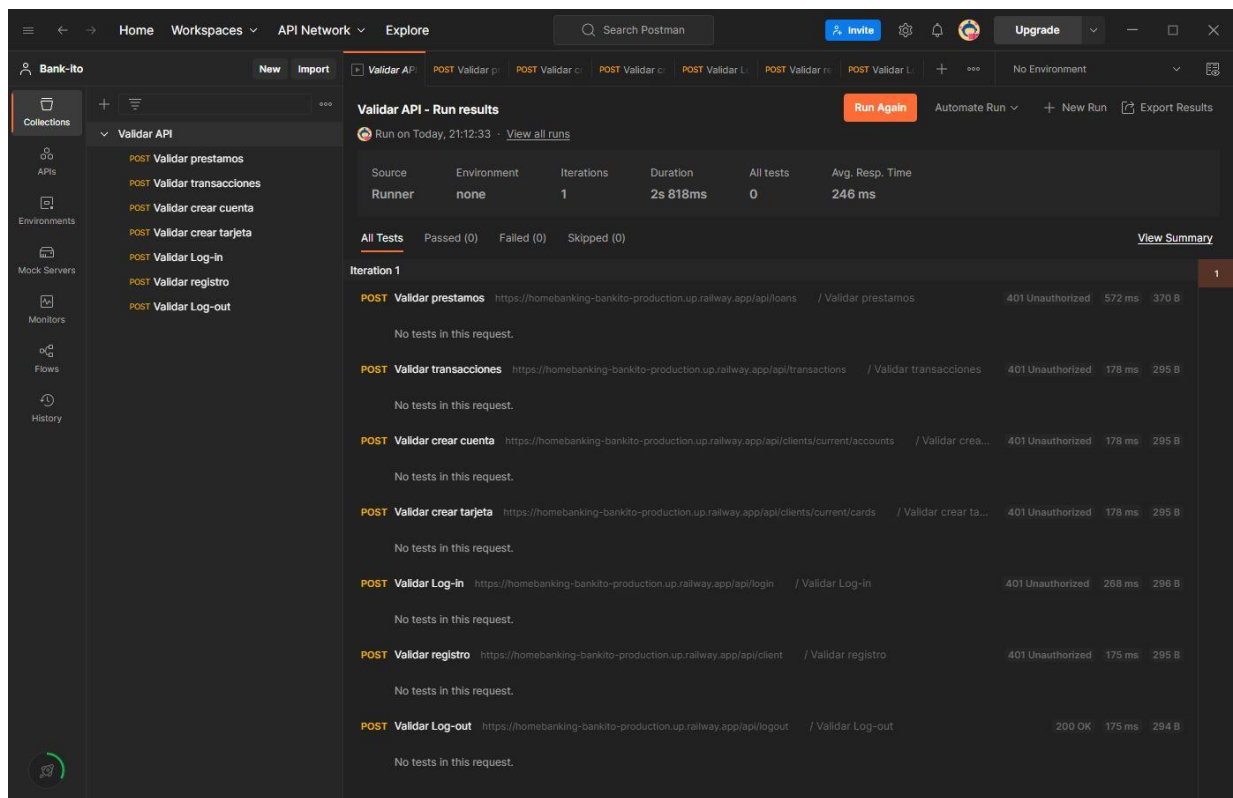
→ Casos de prueba: 7

Del 100% de los casos de prueba, el 14.3% fue SUPERADO, mientras que el 85.7% fue BLOQUEADO.

Test cases

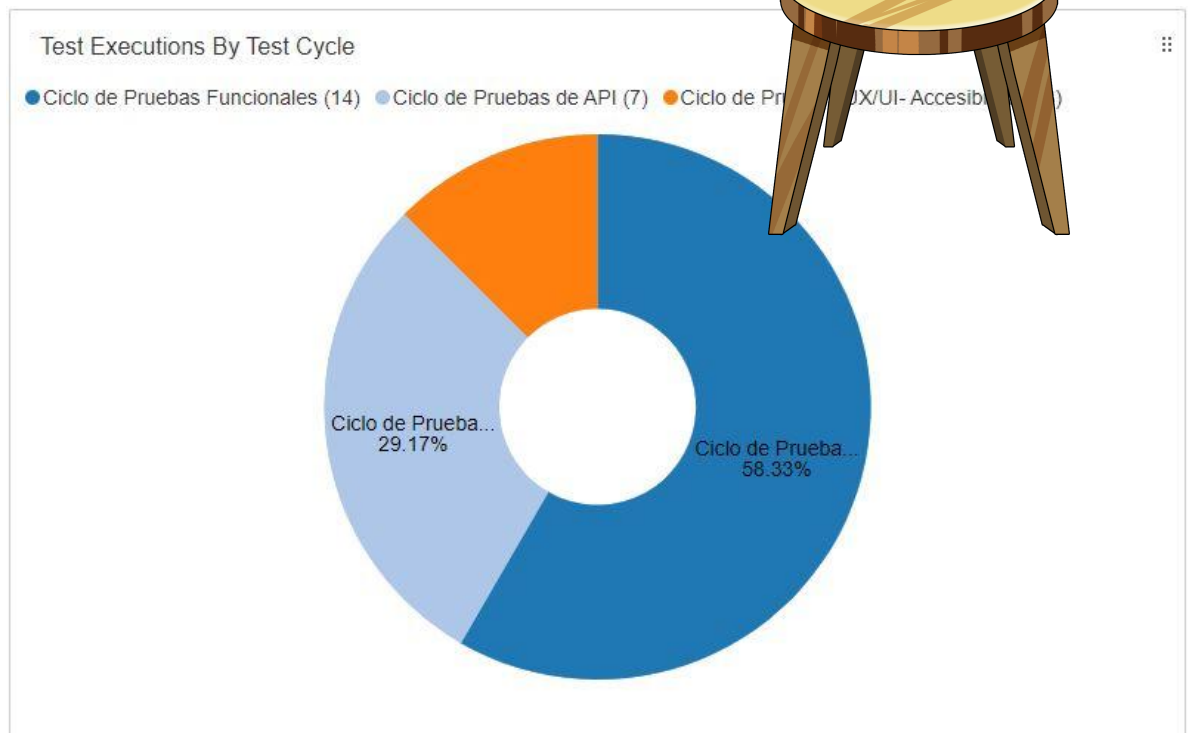


Para el poder ejecutar las casos de prueba de este sprint, hicimos uso de la aplicación **Postman** donde realizamos las distintas peticiones a la API:



Recopilando la información que nos devolvió la aplicación **Zephyr squad**, podemos destacar:

→ Los *Cycle summary* quedaron establecidos de la siguiente manera:



Donde:

- Para el ciclo de pruebas funcionales:

Ciclo de Pruebas Funcionales

Compilación	:	Total Executions	:	14	Start Date	:	2023-03-10
Entorno	:	Cycle Executions	:	14	End Date	:	2023-03-12
Created By	:	Total Executed	:	14	Description	:	View More
Tiempo de ejecución total: 0m		Tiempo registrado total		:	Ejecuciones no controladas: 14		
				:			



- Para el ciclo de pruebas de accesibilidad:

Ciclo de Pruebas UX/UI- Accesibilidad

Compilación	:	Total Executions	:	3	Start Date	:	2023-03-12
Entorno	:	Cycle Executions	:	3	End Date	:	2023-03-12
Created By	:	Total Executed	:	3	Description	:	View More
Tiempo de ejecución total: 0m		Tiempo registrado total		:	Ejecuciones no controladas: 3		
				:			

- Para el ciclo de pruebas de API'S:

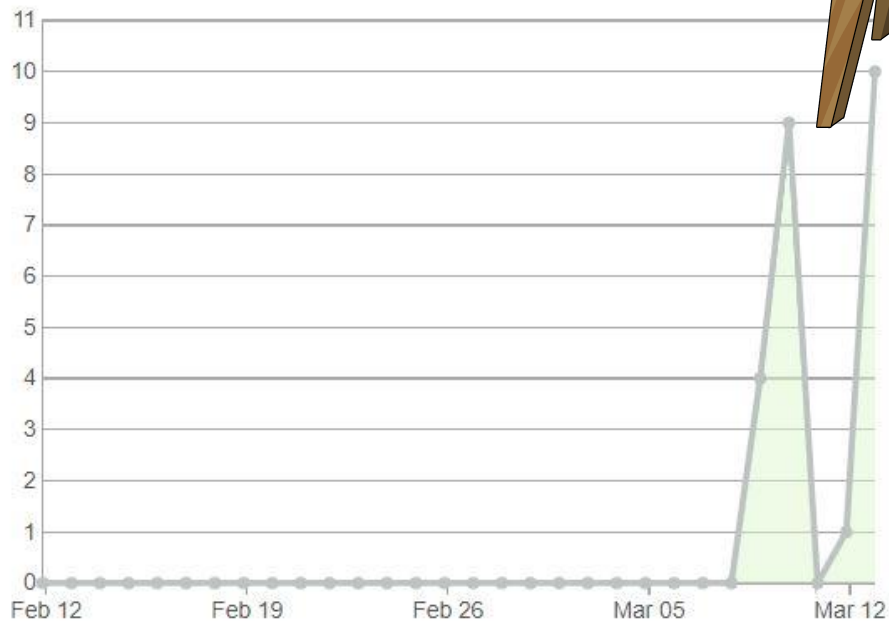
Ciclo de Pruebas de API

Compilación	:	Total Executions	:	7	Start Date	:	
Entorno	:	Cycle Executions	:	7	End Date	:	
Created By	:	Total Executed	:	7	Description	:	
Tiempo de ejecución total: 0m		Tiempo registrado total		:	Ejecuciones no controladas: 7		
				:			

→ Respecto a la duración total de la prueba:

- Creación de la prueba:

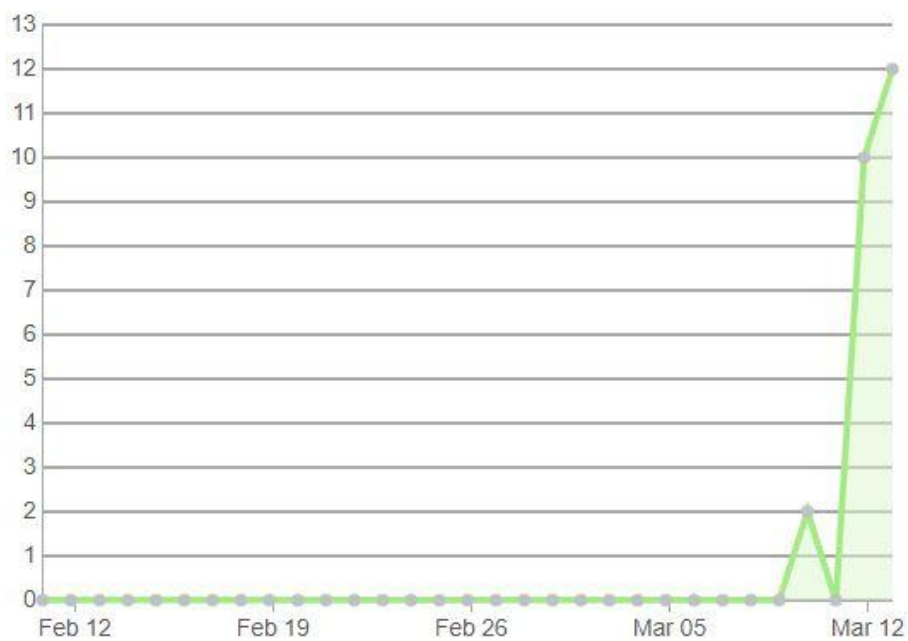
CREACIÓN DE LA PRUEBA: 30 DAY RESUMEN



Pruebas creadas: 24 30 {2} over last {3} days

- Ejecución de la prueba:

EJECUCIÓN DE LA PRUEBA: 30 DAY RESUMEN



Pruebas ejecutadas: 24 30 {2} over last {3} days

→ El diagrama de flujo acumulado:

