

DIGITAL MONEY HOUSE

Digital Money House

Nuestra billetera digital

QA Sign Off

EQUIPO N° 9 - PLAN B SOFTWARE SOLUTIONS

Testing manual: Catalina Forgione

Testing automation: Maria Niveiros

Índice

Introducción	3
Alcance	4
Exit Criteria	6
Resumen de resultados	
• Diseño de pruebas	7
Ejecución de pruebas	
→ Manuales	8
→ Automatizadas	
♦ Postman	9
♦ RestAssured	11
• Reporte de defectos	13
Conclusiones	15

Introducción

Este documento es el Informe Final de Pruebas del sistema **Digital Money House**. 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

Digital Money House es la nueva billetera digital. El usuario puede registrarse, crear una cuenta, asociar sus tarjetas, realizar y recibir transferencias entre cuentas mediante el cvu o el alias, y recargar saldo en su cuenta mediante una tarjeta.



Alcance

Dentro del Alcance

Las pruebas ejecutadas durante los 4 Sprints, abarcaron las siguientes

funcionalidades:

- Registro de usuario.
- Login de usuario.
- Crear una cuenta.
- Asociar una tarjeta a una cuenta.
- Cargar saldo a una cuenta desde una tarjeta.
- Realizar transferencias entre cuentas.
- Consultar historial de transacciones.

Fuera de Alcance

Quedaron fuera del alcance de las pruebas ejecutadas la validación de mail, el recupero de contraseña, la aplicación de filtros de búsqueda y la integración con el frontend.

Tipos de Pruebas Ejecutadas

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Prueba funcional	✓	✓	✓	✓
Prueba de Humo	✓	✓	✓	✓
Prueba de Regresión	✗	✓	✓	✓
Prueba de Integración (Postman)	✓	✓	✓	✓
Prueba automatizada (RestAssured)	✗	✓	✓	✓

Enfoque de la Prueba

Al inicio de cada sprint se crearon los casos de prueba en base a los requerimientos especificados en las user stories, haciendo foco en crear casos de prueba positivos y negativos por cada funcionalidad. Ante cada build recibido, se ejecutaban las pruebas previstas y se realizaba el informe de defectos correspondiente. Se ejecutaron las pruebas de integración utilizando Postman y se reportaron los defectos detectados. Posteriormente, se realizaron pruebas de

confirmación para verificar que los defectos reportados como solucionados por el equipo de desarrollo pasen exitosamente las pruebas correspondientes.

A partir del Sprint 2, se diseñaron, desarrollaron y ejecutaron tests automatizados utilizando RestAssured. Elegimos utilizar Allure para autogenerar los reportes al realizar la ejecución de la suite completa, ya refleja los resultados de una forma muy clara y gráfica, que nos ayuda a interpretar los resultados obtenidos rápidamente, pudiendo enviar el reporte de los test fallados al equipo de desarrollo casi de forma automática.

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 los tests de integración con Postman debe tener un 80% de tasa de tests pasados.
- La ejecución de los tests manuales deben tener un 80% de tasa de tests pasados.
- La ejecución de los test automatizados con RestAssured deben tener un 80% de tasa de tests pasados.

Resumen de Resultados

[Link a Planilla de Testing Manual](#)

Diseño de Pruebas

	Test Manuales	Test Postman	Test RestAssured
Usuarios	87	458	72
Cuentas	8	105	21
Tarjetas	12	196	29
Transacciones	18	268	22
TOTAL TESTS	125	1027	144

Resumen de Resultados

Diseño de Pruebas

Testing Manual

	Test Pasado	Test Fallado	Total Tests
Usuarios	87	0	87
Cuentas	7	1	8
Tarjetas	11	1	12
Transacciones	16	2	18
TOTAL TESTS	121	4	125

Testing Automation Con Postman

▶	POST	Registro exitoso Copy	2	14	×
▶	POST	Registro fallido - Faltante Nombre Copy	3	0	
▶	POST	Registro fallido - Faltante Apellido Copy	3	0	
▶	POST	Registro fallido - Faltante Dni Copy	3	0	
▶	POST	Registro fallido - Faltante email Copy	3	0	
▶	POST	Registro fallido - Faltante Username Copy	3	0	
▶	POST	Registro fallido - Faltante telefono Copy	3	0	
▶	POST	Registro fallido - Faltante contrasenia Copy	3	0	
▶	POST	Registro fallido - Campo Nombre null Copy	3	0	
▶	POST	Registro fallido - Campo Apellido Null Copy	3	0	
▶	POST	Registro fallido - Campo Dni null Copy	3	0	
▶	POST	Registro fallido - Campo email null Copy	3	0	
▶	POST	Registro fallido - Campo username null Co...	2	0	
▶	POST	Registro fallido - Campo phonenumber nul...	3	0	
▶	POST	Registro fallido - Campo contraseña null C...	2	0	
▶	POST	Registro fallido - Campo Apellido empty C...	3	0	
▶	POST	Registro fallido - Campo Nombre empty C...	3	0	
▶	POST	Registro fallido - Campo Dni empty Copy	3	0	
▶	POST	Registro fallido - Campo email empty Copy	3	0	
▶	POST	Registro fallido - Campo username empty ...	2	0	
▶	POST	Registro fallido - Campo phonenumber em...	3	0	
▶	POST	Registro fallido - Campo contraseña empt...	2	0	
▶	POST	Registro fallido - Username existente Copy	3	0	
▶	POST	Registro fallido - email existente Copy	3	0	
▶	POST	Registro fallido -User null Copy	2	0	
▶	POST	Registro fallido -User empty Copy	2	0	
▶	POST	Login exitoso Copy	8	0	
▶	GET	Get All Users Copy	14	0	
▶	GET	Get All Users con Token Inválido Copy	2	0	
▶	GET	Get All Users con Token Vacío Copy	2	0	
▶	GET	Get User By ID Copy	14	0	
▼	GET	Get User By ID - User inexistente Copy	1	2	×

▶	POST	Login fallido - Contraseña null Copy	3		0	
▶	POST	Login fallido - Contraseña vacío Copy	3		0	
▶	POST	Agregar una tarjeta Copy	20		0	
▶	POST	Agregar una tarjeta - token invalido Copy	2		0	
▶	POST	Agregar una tarjeta - token vacío Copy	2		0	
▶	POST	Agregar una tarjeta sin Account ID Copy	3		0	
▶	POST	Agregar una tarjeta con Account ID = null ...	3		0	
▶	POST	Agregar una tarjeta sin Card Number Copy	2		0	
▶	POST	Agregar una tarjeta - card number null Copy	2		0	
▶	POST	Agregar una tarjeta sin Type Copy	2		0	
▶	POST	Agregar una tarjeta con Type = null Copy	2		0	
▶	POST	Agregar una tarjeta sin Onwer Copy	2		0	
▶	POST	Agregar una tarjeta Owner null Copy	2		0	
▶	POST	Agregar una tarjeta sin Security Number C...	2		0	
▶	POST	Agregar una tarjeta Security Number null ...	2		0	
▶	POST	Agregar una tarjeta Balance null Copy	2		0	
▶	POST	Crear una transaction date empty Copy	15		0	
▶	POST	Crear una transaction sin detail Copy	15		0	
▶	POST	Crear una transaction detail null Copy	15		0	
▶	POST	Crear una transaction detail empty Copy	15		0	
▶	POST	Crear una transaction sin type Copy	15		0	
▶	POST	Crear una transaction type null Copy	15		0	
▶	POST	Crear una transaction type empty Copy	15		0	
▶	PUT	Modificar una transaction Copy	1		14	×
▶	PUT	Modificar una transaction - token invalido ...	2		0	
▶	PUT	Modificar una transaction - token vacío C...	2		0	
▶	PUT	Modificar una transaction - account origin...	1		14	×
▶	GET	Get transactions por ID Copy	14		1	×
▶	GET	Get transactions por ID -token invalido Co...	2		0	
▶	DELETE	Eliminar una transacción por Id -token inv...	2		0	
▶	DELETE	Eliminar una transacción por Id - empty to...	2		0	
▼	DELETE	Eliminar una transacción por Id - id invalid...	1		1	×

Automation Testing

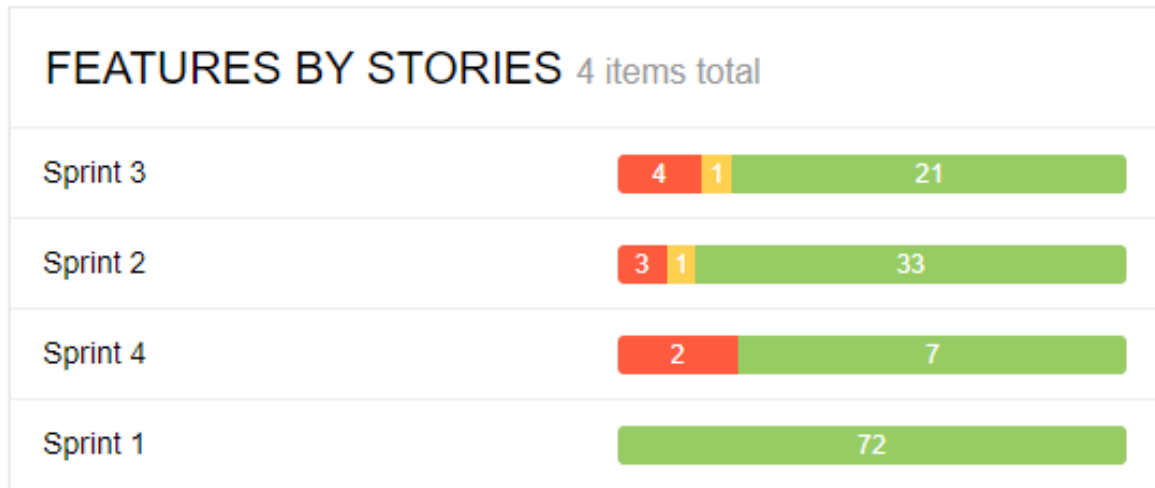


Para automatizar los tests utilizamos RestAssured, que es una tecnología de código abierto muy utilizada para las pruebas de automatización de API REST junto con TestNG, un framework para pruebas y testing que trabaja con Java y está basado en JUnit pero introduciendo nuevas funcionalidades que los hacen más poderosos y fáciles de usar. Para nuestros reportes elegimos Allure, ya que nos permite visualizar los resultados de la ejecución de un proyecto automatizado de una forma más clara y gráfica.

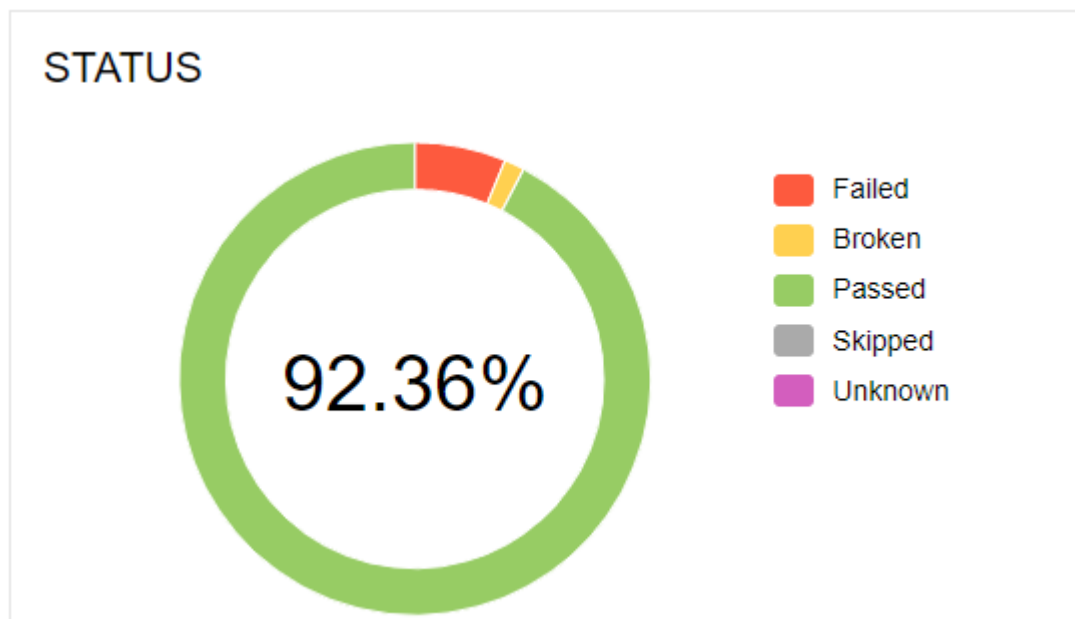
[Link al proyecto RestAssuredTests](#)

La siguiente sección muestra información obtenida del dashboard que nos proporciona Allure con los resultados de los tests al finalizar la ejecución de la suite de pruebas en RestAssured.

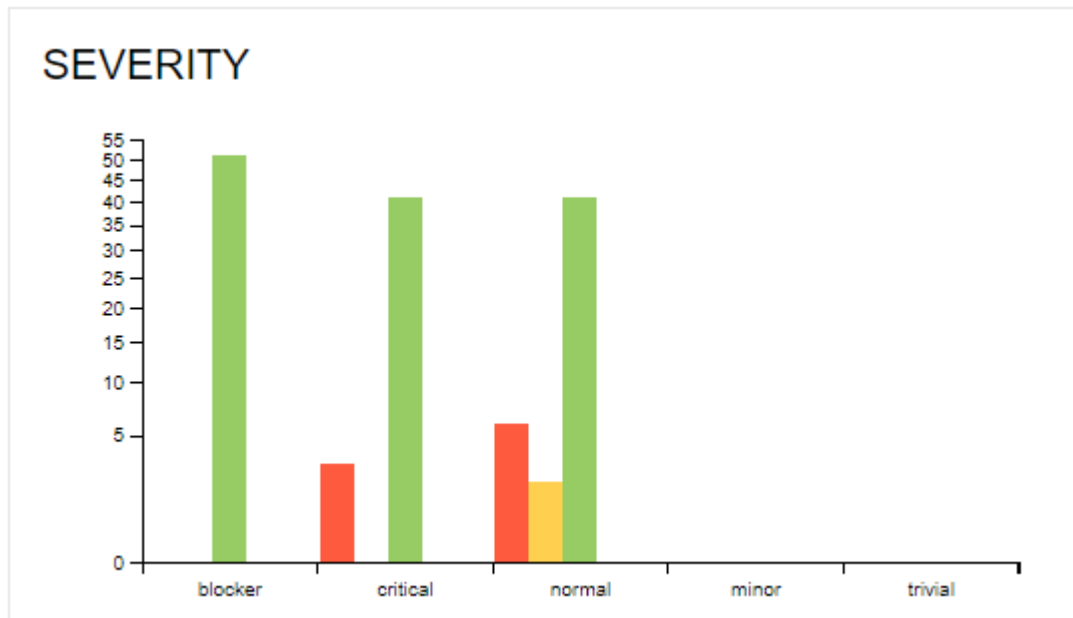
Tests por Sprint



Tests por estado



Tests por severidad



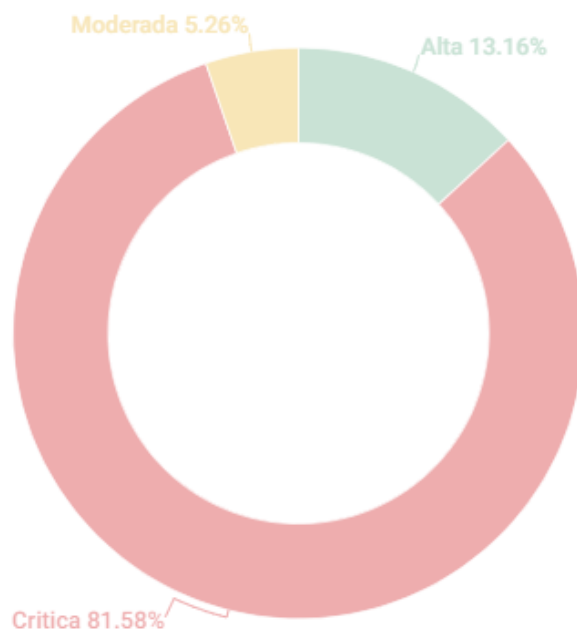
Reporte de Defectos

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

Tests manuales ejecutados por Sprint



Sobre 76 defectos encontrados

Defectos por Sprint



Defectos Abiertos

La siguiente sección muestra información con respecto al número total de defectos que permanecen abiertos al final de la fase de pruebas. No se registran defectos abiertos al final de la fase de pruebas.

Lecciones Aprendidas / Conclusión

Durante este proceso de aprendizaje e implementación de los conocimientos adquiridos a lo largo de la cursada, pudimos conocer más en profundidad las tareas que realiza un QA.

Ampliamos nuestro dominio de las herramientas que habíamos utilizado anteriormente como Postman para realizar el Api Testing y RestAssured, en el caso de automation. Mejoramos la forma de documentar nuestras planillas de casos de prueba, reportes de defectos e informes de pruebas.

Toda esta experiencia nos permitió poner a prueba nuestro sistema, hacerlo fallar, buscar defectos y solucionarlos, haciendo de nuestro producto un sitio mucho más estable y confiable. Aún sabiendo que la prueba exhaustiva es imposible, procuramos abarcar la mayor cantidad de casos, a fin de minimizar los defectos existentes.