

Testing Kickoff

PLAN DE PRUEBAS

DIGITAL MONEY

Análisis del Producto	Pág. 03
Diseño de la Estrategia de Pruebas	Pág. 04
Alcance de la Prueba	Pág. 04
Tipo de Prueba	Pág. 04
Riesgos y Problemas	Pág. 05
Logística de las Pruebas.....	Pág. 05
Testers	Pág. 05
Responsabilidades	Pág. 05
Herramientas	Pág. 05
Issues a Probar en Sprint 1	Pág. 05
Issues a Probar en Sprint 2	Pág. 06
Issues a Probar en Sprint 3	Pág. 06
Issues a Probar en Sprint 4	Pág. 06
Definición de Objetivos	Pág. 08
Criterios de Pruebas	Pág. 09
Criterios de Entrada	Pág. 09
Criterios de Suspensión/Reanudación	Pág. 09
Criterios de Salida	Pág. 09
Criterio para incluir un caso de prueba en una Suite de Humo	Pág. 10
Criterio para incluir un caso de prueba en una Suite de Regresión	Pág. 10
Casos de Prueba	Pág. 11

Tabla modelo para documentación de Casos de Prueba	Pág. 11
Tabla modelo para documentación ejecución de Casos de Prueba	Pág. 11
Reporte de Defectos	Pág. 12
Entregables	Pág. 13
Reporte de Cubrimiento del Testing	Pág. 13
Métricas relacionadas al Proceso	Pág. 13
Métricas relacionadas al Producto	Pág. 14
Scripts.....	Pág. 14
Adicionales	Pág. 14
Reporte final	Pág. 14

Análisis del Producto

El producto a probar es el MVP de la billetera virtual llamada Digital Money House.

En esta billetera el usuario tendrá las funcionalidades básicas de **registro, inicio y cierre de sesión**, dar de **alta medios de pago como tarjetas de débito o crédito**, **recargar saldo** en su billetera y poder usar este saldo para **hacer envíos de dinero a otros usuarios y/o cuentas**.

El usuario contará con una **Clave Virtual Uniforme (CVU)** única que será la cuenta que le pertenece.

La billetera deberá llevar **registro de todas las transacciones (ingresos y egresos)** que se realizan a través de la billetera (**movimientos o actividad de la cuenta**).

Este desarrollo deberá estar conectado a un frontend desarrollado y provisto por DH.

Estrategia de Pruebas

- **Alcance de la Prueba:**

Realizaremos pruebas funcionales, para evaluar el comportamiento del software y las funciones que éste debe realizar.

Cada una de las pruebas debe estar enmarcada dentro de un flujo.

En el sprint 1, los flujos definidos son:

- Registro de usuario
- Inicio de sesión
- Cierre de sesión

En relación a la seguridad, vamos a validar la integridad del token para cada una de las transacciones, entendiéndose estas como alta de tarjetas de crédito o débito, carga de saldo o transferencias de dinero y acceso a los movimientos de la cuenta; y también la autorización y autenticación de usuarios.

No probaremos requerimientos no funcionales tales como performance, sobrecarga, portabilidad y escalabilidad.

- **Tipo de Prueba:**

Sprint 1	Sprint 2	Sprint 3	Sprint 4
Pruebas manuales Pruebas exploratorias			

- **Riesgos y Problemas:**

Uno de los riesgos es la ambigüedad de los requerimientos, concretamente en cuanto al plan de pruebas y a los tests manuales por parte de los stakeholders. También la falta de asesoramiento técnico en las etapas iniciales del proyecto. Falta de experiencia en los testers sobre algunos temas por ejemplo el uso de herramientas y frameworks para tests y/o vacíos conceptuales no abordados durante el CTD. Manejo inadecuado del tiempo o imprevistos personales.

- **Logística de las Pruebas:**

Los testers serán:

- André Alvarez
- Ana Laura Fidalgo
- Facundo Krivanek
- Hernán Felipe Serna Gómez

Ellos realizarán el diseño de los casos de prueba, y la ejecución de los tests manuales será organizada por flujos (detallados anteriormente en este documento).

La persona que ejecute cada caso en particular será la encargada de reportar los defectos encontrados (si los hubiera) y luego realizar la verificación y consecuente cierre del mismo una vez que haya sido corregido.

Las herramientas que utilizaremos son Postman y Rest Assured.

El proyecto consta de de 4 sprints:

Sprint 1	
Fecha	08/05/2023 al 19/05/2023
Issues a Probar	Como usuario quiero registrarme a Digital Money House para acceder y

	<p>usar los servicios que ofrece.</p> <p>Como usuario quiero acceder a Digital Money House para poder realizar transferencias de fondos.</p> <p>Como usuario necesito poder cerrar sesión en la billetera Digital Money House.</p> <p>Como usuario necesito recuperar mi contraseña vía correo electrónico (opcional).</p> <p>Como usuario me gustaría validar mi correo electrónico cuando me registre a Digital Money House y haga el primer login (opcional).</p>
--	--

Sprint 2	
Fecha	22/05/2023 al 02/06/2023
Issues a Probar	PROXIMAMENTE

Sprint 3	
Fecha	05/06/2023 al 16/06/2023
Issues a Probar	PROXIMAMENTE

Sprint 4	
Fecha	19/06/2023 al 30/06/2023

Issues a Probar	PROXIMAMENTE
-----------------	--------------

Definición de Objetivos

- Evaluar correctamente los “happy path” en cada uno de los flujos
- Registrar y probar la mayor cantidad posible de casos de prueba positivos (que validan el flujo normal del sistema, es decir, los requisitos funcionales) y negativos (que validan flujos no contemplados dentro de los requisitos del sistema)
- Describir y probar cada uno de los casos border establecidos
- Encontrar tantos errores como sea posible y corregirlos para reducir los defectos que se puedan presentar en el código
- Obtener información acerca de la calidad del producto
- Obtener información sobre el proceso de pruebas

Criterios de Prueba:

- **Criterios de Entrada:**

La ejecución de los tests puede comenzar una vez que cada issue esté marcado como finalizado por parte del desarrollador.

- **Criterios de Suspensión y Reanudación:**

Si al ejecutar un caso de prueba se encuentra un defecto, este debe ser reportado y ese caso de prueba particular suspendido hasta que los desarrolladores corrijan el error.

Si los miembros del equipo de control de calidad descubren que el 50% de los casos de prueba han fallado, se suspenderán todas las pruebas hasta que los desarrolladores resuelvan todos los errores identificados hasta el momento.

- **Criterios de Salida:**

Se definieron los siguientes criterios de aceptación para finalizar las pruebas:

- No se debe tener defectos en ESTADO ABIERTO de SEVERIDAD ALTA (crítica y/o bloqueante).
- No se debe tener más de un 20% de defectos en ESTADO ABIERTO de SEVERIDAD MEDIA implicando a los que afectan la funcionalidad del sistema pero no impiden seguir trabajando.
- Se pueden tener defectos en ESTADO ABIERTO de SEVERIDAD BAJA ya que no afectan la funcionalidad del sistema.
- La ejecución de los tests en Postman deben tener un 75% de pass rate

- **Criterio para incluir un caso de prueba en una Suite de Humo:**

Para ser incluido en una Suite de Humo, el caso de prueba debe tratarse de una funcionalidad principal y crítica. Las pruebas de humo son pruebas básicas que sirven para comprobar el funcionamiento esencial de la aplicación. Están concebidas para ejecutarse rápidamente, y su objetivo es ofrecer la seguridad de que las principales funcionalidades del sistema funcionen según lo previsto.

Las utilizaremos justo después de realizar una compilación nueva para decidir si se pueden ejecutar o no pruebas más exhaustivas (y por ende más caras y que llevan más tiempo), o inmediatamente después de una integración para asegurarnos de que la aplicación funciona correctamente en el entorno que se acaba de implementar.

Generalmente las pruebas de humo son un subconjunto que luego serán parte de una batería de pruebas más amplia que es la suite de regresión.

- **Criterio para incluir un caso de prueba en una Suite de Regresión:**

Las pruebas de regresión consisten en un examen profundo y exhaustivo del software para garantizar que cualquier cambio de código reciente no haya afectado negativamente las funciones existentes. Aquí el criterio para incluir una prueba en este conjunto es cuando se realiza cualquier corrección de errores, modificación de código, cambios de requisitos, corrección de defectos o cualquier desarrollo de módulo nuevo.

Incluyen las pruebas de las funciones esenciales del sistema (humo) pero la suite de regresión es una batería de pruebas mucho más extensa y exhaustiva; que consume más tiempo y recursos.

Casos de Prueba

Como hemos mencionado anteriormente, los casos de prueba van a estar organizados por flujos. Se documentarán tanto casos positivos como negativos.

Tabla modelo para documentación de Casos de Prueba:

Id	Título o Nombre	Descripción	Precondición	Input	Resultado Esperado	Estado de Diseño	Diseñado Por	Suite	Automatización
Formato: TC_nombreflujo_0001 Ejemplo: TC_login_0001 Numeración secuencial Unico	Sintético y explicativo	Más detalle sobre el caso de prueba		Curl de la request en Postman		En Diseño Diseñado No Ejecutado Pasado Fallado Diferido Bloqueado	Nombre del QA responsable	Humo o Regresión	Si o No

Tabla modelo para documentación de ejecución de Casos de Prueba:

Fecha	Id	Estado De Ejecución	Datos de Prueba	Ejecutado Por	Id Defecto	Nota	Referencias
Formato: DD/MM/YYYY	Id del caso de prueba a ejecutar	No Ejecutado Pasado Fallado Diferido Bloqueado		Nombre del QA responsable	Id del defecto que reportaremos si el caso ha fallado	Información extra	Información extra

Reporte de Defectos

Deberán reportarse como defecto todos los casos de prueba cuyo resultado no sea el esperado.

Tabla modelo para Reporte de Defectos:

Id	Fecha Creación	Título o Nombre	Descripción	Resultado Actual	Resultado Esperado	Input (Para Reproducción)
Formato: DFF_nombre flujo_0001 Ejemplo: DEF_login_0 001 Numeración secuencial Unico	Formato DD/MM/YYYY	Corto y específico, que se entienda lo que queremos reportar	Desarrollar lo que dejamos afuera en el título lo podríamos explicar acá	Lo que sucedió realmente o lo que nos mostró la aplicación	Lo que esperamos que suceda o muestre la aplicación según los requerimientos de la misma	Info para reproducir el defecto

Fecha Cambio Estado	Estado	Prioridad	Severidad	Reportado Por	Asignado A	Referencias
Formato DD/MM/YY YY	Nuevo Asignado En progreso Corregido En espera de verificación En verificación Verificado Cerrado Diferido Duplicado Rechazado	La prioridad del defecto es definida por el cliente/usuario. Escala: 1. Baja 2. Media 3. Alta 4. Critica	La severidad del defecto es definida por el equipo IT Escala: 1. Baja 2. Media 3. Alta 4. Critica	Nombre del QA que lo ha reportado	Nombre de quien lo verificará	Información adicional, capturas de pantalla, etc

Entregables:

- **Reporte de cubrimiento**

Para **controlar el estado de las actividades de prueba** y el **progreso con respecto al plan de pruebas**, **al finalizar cada sprint se entregará un reporte** con información sobre lo ocurrido durante el periodo de pruebas - un resumen sobre las pruebas realizadas. Este incluirá información sobre:

- **Defectos:** defectos encontrados y corregidos y los resultados de pruebas de verificación
- **Casos de prueba:** casos de prueba planificados y casos de prueba ejecutados, y que porcentaje de los casos de prueba ejecutados han sido superados o no
- **Cobertura de las pruebas:** Cobertura de pruebas de requisitos e historias de usuario, criterios de aceptación, riesgos o código
- **Factores** que están **bloqueando** o han bloqueado el progreso
- **Estado general de la prueba y calidad del producto** con respecto a los criterios de salida

Teniendo en cuenta que en la **calidad del software** es fundamental disponer de métricas para un **correcto seguimiento y control de las pruebas**, para realizar el reporte utilizaremos las siguientes **métricas**:

Métricas relacionadas al Proceso

- **Cobertura de ejecución de pruebas:** Número total de casos de prueba ejecutados / Número total de casos de prueba previstos para la ejecución.
- **Casos de prueba superados:** Número total de casos de prueba aprobados / Número total de casos de prueba ejecutados.
- **Casos de prueba fallidos:** Número total de casos de prueba fallidos / Número total de casos de prueba realizados.

Métricas relacionadas al Producto

→ **Tasa de detección de errores:** (Número total de defectos encontrados / Número total de casos de prueba realizados) *100.

- **Planilla de Casos de Prueba y ejecución de los mismos**
- **Reporte de Defectos**
- **Scripts**
- **Adicionales**
- **Reporte final de testing**