

Test Plan

Grupo 3

Control de Versiones

Fecha	Versión	Descripción	Revisor	Aprobador
20/10	0.1	Creación de Test Plan (Kick Off)		

1. Propósito

El propósito del presente manual de pruebas es describir la estrategia general de testing que se aplicará a lo largo del desarrollo de **‘Digital Money House’**, una billetera virtual que busca facilitar y agilizar transacciones de manera accesible. El manual proporcionará un acuerdo compartido entre los integrantes del equipo para llevar a cabo un proceso de testing con metodología y documentación.

Objetivos principales:

- Definir un estándar para testear la calidad del software, estableciendo con claridad los requisitos mínimos que deben cumplirse para su aceptación.
- Establecer criterios para definir defectos y casos de prueba, así como también, niveles y tipos de prueba a aplicar.

2. Estrategia

a. Alcance

El alcance del manual está enfocado a pruebas de interacción con la **GUI** (Interfaz Gráfica de Usuario) y la validación de funcionalidades recibidas según las especificaciones proporcionadas por el cliente en cada sprint.

b. Ciclos - Sprint

El equipo llevará la metodología ágil Scrum a lo largo del proceso de desarrollo de la aplicación.



En este kickOff podremos aproximarnos a plantear un proceso de testing que irá iterando a lo largo del ciclo de vida del producto.

Partiendo de los requerimientos recibidos por parte del cliente, se analizan las historias de usuario correspondientes. En esta primera etapa se conoce de manera general el objetivo del sistema y las funcionalidades que se pondrán a prueba en este primer sprint.

En un siguiente paso (Planeación) podremos analizar más en detalle las condiciones de test necesarias para cubrir cada una de las funcionalidades, por ejemplo si necesitamos conjuntos de datos previos, cuentas registradas, APIs probadas, entre otras.

En el Diseño, luego de entender la funcionalidad, diseñaremos los casos de prueba (tanto positivos como negativos) y procesos exploratorios para test manual.

En desarrollo se codificaremos el plan diseñado para cada test case, esto es: el desarrollo de pruebas unitarias según el código realizado en próximos sprints, (JEST para componentes aislados de la interfaz) y reporte/resolución de defectos (BUG).

La ejecución será en la etapa de Testing: pruebas manuales de los casos de prueba, los test unitarios de componentes y los test exploratorios. Con la ejecución y dependiendo del resultado, se consideran si cumplen los criterios de salida o deben ser re-testeados.

Generalmente, luego de un nuevo sprint, nuevas funcionalidades o nuevos despliegues, las suites de humo (conjunto de test-cases) nos ayudarán a comprobar rápidamente si el flujo principal y básico del sistema no ha sido perjudicado, por tanto, planeamos tenerlos en cuenta antes de ejecutar las pruebas de nuevas funcionalidades.

Finalmente al ir cerrando cada sprint, planearemos pruebas exploratorias (flujo completo de usuario hacia un objetivo), y de integración, en las cuales involucraremos todo lo desarrollado durante el sprint junto con la interacción entre componentes.

3. Casos de Prueba

Un caso de prueba es un escenario que describe una entrada al sistema, una acción o evento y el resultado esperado requerido para evaluar el comportamiento de la característica o componente. Los Casos de Prueba deben cubrir el 100% de los Criterios de Aceptación de las Historias de Usuarios y también validar escenarios negativos y adicionales.

Cada caso de prueba debe seguir los siguientes parámetros:

- **Id:** debe tener un identificador único. Seguir nomenclatura:
 - *TC-001* (TestCase-Número de Test)
- **Pre-condición:** estado en el que debe estar el sistema para poder ejecutar el caso de prueba.
- **Estado:** (*toDo, Fail, Success*)

CASO DE PRUEBA							Estado	Reportado por
Id	Nombre del Caso	Descripción	Precondición	Pasos				
				#	Acción	Resultados esperados		

4. Reporte de Defectos

Luego de las pruebas de software se pueden encontrar **defectos**; si no son reportados antes de que se libere para su uso o pase a producción provocarán fallas, por tanto, es importante en cada ejecución de test, ir reportando siguiendo la plantilla:

- **Id:** debe tener un identificador único. Seguir nomenclatura:
 - *BUG-001* (BUG-Número de Defecto)
- **Pasos para la Reproducción:** debe ser reproducible para ser reportado como defecto. Adicionar toda información necesaria para ayudar al asignado a reproducirlo.
- **Estado:** (*Abierto, En Progreso, Cerrado*)
- **Prioridad:** en relación a la urgencia de su solución (regla de negocio) (*Baja, Media, Alta*)
- **Severidad:** en relación a lo técnico y a la experiencia de usuario (*Baja, Media, Alta*)

DEFECTO						Estado	Prioridad	Severidad	Reportado por	Asignado a
Id	Título	Descripción	Pasos para la reproducción	Resultado						
				Esperado	Actual					

Estos defectos deben ser asignados para darles solución, priorizando en primera medida aquellos de perjuicio alto o bloqueante.

5. Tipos de Pruebas

a. Smoke Suite

La suite de humo está compuesta por un conjunto de casos de prueba (Test-Cases) que conforman un **flujo básico y crítico** del sistema. (regla de mínimas de negocio).

El propósito de este tipo de suite es poder verificar la compilación correcta de este flujo básico, por tanto, debe ser una prueba que no demande mucho tiempo; es eficiente para comprobar rápidamente que las funcionalidades significativas no sufrieron fallas luego de algún despliegue.

b. Regression Suite

Esta suite asegura que luego de algún cambio o luego de alguna nueva funcionalidad, los componentes o **interfaces existentes de la aplicación no han sufrido daños**. Teniendo esto en cuenta, la prueba de regresión confirma que lo que antes del despliegue funcionaba, sigue aún funcionando de la misma manera.

6. Exit Criteria

Los criterios de aceptación definidos para dar paso a la finalización de las pruebas ejecutadas, tiene los siguientes principios:

- Casos de pruebas de sprint pasados totalmente cubiertos.
- Defectos cerrados de severidad crítica.
- Test de Suite de humo 100% aprobado.
- Test de Suite de regresión 100% aprobado.

Enlace a Plantilla de issues, test Cases, suites y defectos:

[DHMONEY Testing](#)