



# TESTING KICK OFF

## PLAN DE PRUEBAS

## Análisis del Producto

Digital Money House es una billetera virtual en la que los usuarios podrán enviar y recibir dinero, realizar pagos, gestionar sus tarjetas de crédito y débito, entre otras operaciones disponibles.

El backend de esta aplicación está desarrollado en lenguaje Java. Desarrollar una aplicación con POO (programación orientada a objetos) es mucho más sencillo y también contribuye a mantener un sistema modular, flexible y extensible.

## Diseño de la estrategia de pruebas

### **Alcance de la prueba:**

- En el Sprint 1, se realizarán las pruebas sobre el módulo USERS, más específicamente en los endpoints “/user/register”, “user/login” y “user/logout”, para comprobar el registro de usuario nuevo, el login de un usuario registrado y el logout de un usuario registrado. No se realizarán pruebas al front end de la aplicación.

### **Tipo de prueba:**

- Sprint 1: En base a los requerimientos se redactarán los casos de prueba en la planilla TESTING los cuales se ejecutarán de forma manual mediante el uso de Postman. Los casos de prueba que componen la suite de humo creada en este Sprint, serán clasificados como Funcionales o No Funcionales.
- Todos aquellos defectos encontrados, serán informados mediante una planilla BugReport al equipo de desarrollo, especificando la relevancia y la prioridad de cada defecto.

### **Definición de objetivos**

- Sprint 1: Se probarán los endpoints “user/register”, “user/login” y “user/logout”, para poder registrar un nuevo usuario, loguearse mediante un usuario registrado y cerrar la sesión de dicho usuario respectivamente y que cada prueba retorne el valor esperado. Se verificarán en caso de error que retornen el código de error correcto.

## Establecer los criterios de prueba

### **Definición de un caso de Prueba:**

Un caso de prueba efectivo debe ser claro, conciso y completo. Debe incluir los siguientes elementos:

- **Identificador único:** Un número o código único para identificar el caso de prueba.
- **Descripción:** Una descripción breve del objetivo o funcionalidad que se está probando.
- **Pasos:** Una lista ordenada de pasos detallados que el probador debe seguir para ejecutar el caso de prueba.
- **Datos de prueba:** Cualquier dato específico que se requiera para ejecutar el caso de prueba.
- **Resultado esperado:** La salida o comportamiento esperado cuando se ejecuta el caso de prueba.
- **Estado:** El estado actual del caso de prueba (No ejecutado, Aprobado, Fallido, etc.).

### **Criterios de inclusión de un caso de prueba en una suite de humo**

La suite de humo está diseñada para evaluar rápidamente si el sistema está básicamente funcional después de cambios significativos. Los criterios para incluir un caso de prueba en una suite de humo son:

- **Escenarios críticos:** Casos de prueba que cubren las funcionalidades esenciales y críticas del sistema.
- **Cobertura mínima:** Casos de prueba que abarcan las principales características y flujos de trabajo del sistema.
- **Rendimiento rápido:** Casos de prueba que se ejecutan rápidamente para proporcionar una indicación rápida de la estabilidad general del sistema.
- **Resultados claros:** Casos de prueba que tienen resultados fácilmente interpretables y que permiten identificar rápidamente cualquier problema grave.

### **Criterios de inclusión de un caso de prueba en una suite de regresión**

La suite de regresión se utiliza para asegurar que las nuevas modificaciones o actualizaciones del sistema no introduzcan nuevos defectos ni afecten a las funcionalidades existentes. Los criterios para incluir un caso de prueba en una suite de regresión son:

- Funcionalidad crítica
- Funcionalidad afectada

### **Criterios de suspensión:**

- **Sprint 1:** En caso de que el 50% de las pruebas ejecutadas fallen, se suspenderán automáticamente todas las pruebas, informando al equipo de desarrollo y a la espera de la solución de todos los defectos para poder continuar con las pruebas establecidas para este Sprint.

**Criterios de salida:**

- Sprint 1:
  - En caso que el 80% o más de las pruebas ejecutadas en el sprint deben ser calificados como exitosos para considerar que este criterio se cumple satisfactoriamente.
  - No deben registrarse defectos en estado abierto que cuenten con relevancia crítica.

## Planificación de la configuración del entorno de pruebas

En Postman, se crea un Workspace denominado Digital Money House, con acceso abierto a todo el equipo de testers..

Creamos una variable de entorno llamada localhost, cuyo valor será `http://localhost:8080`.

En el caso de Web, la variable de entorno Host tendrá como valor el dominio de nuestra aplicación.

Esto nos permite reutilizar los tests en cualquiera de los ambientes creados.