**Inicio de Testing – Plan de Pruebas Kick Off**

**Producto: Digital Money House**

**Digital Money House** es una aplicación de tipo billetera virtual que permite a los usuarios realizar pagos de servicios, enviar dinero a otras personas, asociar tarjetas de crédito o débito, y consultar sus movimientos financieros, entre otras funcionalidades.

**Objetivo del Sprint 1 – Alcance de Pruebas**

En esta primera iteración, el foco está en validar las funciones básicas de acceso del usuario a la aplicación. Las funcionalidades bajo prueba son:

* Registro de usuario: http://localhost:8084/user/register
* Inicio de sesión: http://localhost:8084/user/login
* Recuperación de contraseña: http://localhost:8084/user/{username}/forgot-password
* Cierre de sesión: http://localhost:8081/user/logout

Estas pruebas se realizarán manualmente utilizando la herramienta **Postman**, con un enfoque de **pruebas de humo**. La documentación de los resultados se llevará a cabo en una planilla de pruebas manuales.

*Nota: Se recomienda reemplazar los endpoints locales (localhost) por variables de entorno como {{baseUrl}} para facilitar la ejecución en distintos ambientes (test, QA, staging, etc.).*

**Estructura de un Caso de Prueba**

Cada caso de prueba será redactado de acuerdo con la siguiente plantilla:

* **ID**: Identificador único del caso (ejemplo: TC01\_REGISTER\_SUCCESS)
* **Nombre del Caso**: Título breve que describa el objetivo del test.
* **Descripción**: Detalle del flujo que se probará y el propósito de la validación.
* **Entorno**: Ambientes en los que se ejecutará el caso (dev/test/staging).
* **Precondiciones**: Estado inicial o configuración requerida para ejecutar el caso.
* **Pasos de Ejecución**:
  + Enumeración clara de las acciones a realizar.
  + Resultado esperado por cada paso.
* **Tipo de Prueba**: Humo, regresión, validación, etc.
* **Estado del Diseño**: Indica si el caso está en borrador, en revisión o finalizado.
* **Severidad**: Impacto que tendría el fallo de esta funcionalidad.
* **Prioridad**: Nivel de urgencia con el que se debe atender este caso o sus defectos.

**Reporte de Defectos**

Cualquier desviación respecto al comportamiento esperado se documentará en una hoja de control de errores (Excel o similar), que incluirá:

* Funcionalidad afectada.
* Resultado obtenido vs. esperado.
* Gravedad del defecto.
* Prioridad de resolución.
* Estado actual del defecto (nuevo, en análisis, resuelto, etc.).

🔎 *Se considerarán errores aquellos comportamientos que no cumplan con la funcionalidad esperada total o parcialmente.*

**✅ Criterios para la Suite de Pruebas de Humo**

La suite de **smoke testing** debe permitir una verificación rápida de la estabilidad mínima del sistema tras cada despliegue. Para incluir un caso de prueba en esta suite, se deben cumplir los siguientes criterios:

* **Flujos de uso principal**: Validación de caminos críticos (ej. login, registro).
* **Requisitos básicos**: Deben representar funcionalidades imprescindibles para el funcionamiento general del sistema.
* **Velocidad de ejecución**: Deben ser rápidos de ejecutar, sin necesidad de validaciones complejas.
* **Poca complejidad**: Se privilegian pruebas simples, que permitan detectar fallos rápidamente.
* **Alta frecuencia de uso**: Se priorizan funciones que los usuarios utilizan regularmente.

**Criterios para la Suite de Regresión**

La suite de regresión busca garantizar que nuevos cambios no generen errores en funcionalidades ya existentes. Los casos incluidos deben cubrir:

* **Zonas del sistema modificadas recientemente**: Toda funcionalidad directamente impactada por los cambios en código.
* **Procesos críticos para el negocio**: Funciones clave para el éxito del producto y la satisfacción del cliente.
* **Errores corregidos anteriormente**: Se incluirán pruebas para escenarios que previamente generaron errores.
* **Reutilización efectiva**: Siempre que los casos de prueba ya existentes sigan siendo válidos, se reutilizarán evitando redacción innecesaria de duplicados.

**Cobertura y Tipos de Pruebas**

Durante el Sprint 1 se abordarán principalmente:

* Pruebas funcionales (positivo y negativo).
* Validación de status HTTP, estructura de respuesta y mensajes de error.
* Casos con datos válidos e inválidos.
* Pruebas exploratorias sobre flujos no documentados.

**Herramientas y Datos**

* **Herramienta principal**: Postman.
* **Gestión manual de resultados**: planilla Excel compartida.
* **Datos de prueba**: usuarios ficticios creados para la validación. En fases posteriores se prevé automatizar la generación y limpieza de datos.

**Planificación**

Se estima una duración de pruebas de **X días** (dependiendo del volumen de casos). Las pruebas comenzarán una vez completada la entrega funcional del equipo de desarrollo. El equipo de QA asignado será responsable de mantener actualizada la documentación y de comunicar resultados en los dailys.