# Documento del Proyecto

## 1. Objetivos del Proyecto

El objetivo de este proyecto es desarrollar una API REST para una billetera virtual llamada **Digital Money House**, proporcionando funcionalidades clave para la gestión de saldo, pagos y transacciones de los usuarios. Se busca construir un producto mínimo viable (**MVP**) con un enfoque en seguridad, escalabilidad y facilidad de uso.

Los principales objetivos incluyen:

* Permitir a los usuarios registrarse, iniciar y cerrar sesión de manera segura.
* Gestionar el saldo disponible en la cuenta del usuario.
* Registrar ingresos de dinero y realizar transferencias.
* Administrar medios de pago y realizar pagos desde la billetera virtual.
* Ofrecer una interfaz clara para consultar actividad y movimientos financieros.
* Implementar autenticación y autorización robusta mediante JWT.

## 2. Planificación y Descripción de Actividades (Backlog)

El desarrollo se dividirá en cuatro sprints, de dos semanas cada uno, enfocados en implementar una funcionalidad clave del sistema.

### Sprint 1: Registro, Inicio y Cierre de Sesión (Semana 1 - 2)

* Implementar el **registro de usuario**.
* Desarrollar el **inicio y cierre de sesión** con JWT.
* Crear la base de datos en MySQL.
* Implementar pruebas unitarias y de integración.
* Desplegar el servicio para validación inicial.

### Sprint 2: Mi Perfil - Registro de Tarjetas - Ingreso de Dinero (Semana 3 - 4)

* Implementar endpoints para **consulta y edición de datos del usuario**.
* Crear el **CRUD para tarjetas** asociadas a la cuenta.
* Permitir **ingreso de dinero** desde una tarjeta.
* Aplicar autenticación: solo usuarios logueados pueden acceder.
* Mejorar pruebas automatizadas.

### Sprint 3: Transferencias y Dashboard (Semana 5 - 6)

* Implementar **transferencias de dinero** entre usuarios.
* Crear el **Dashboard** con:
  + Saldo disponible
  + CVU y alias del usuario
  + Últimos movimientos
* Integrar autenticación JWT en todos los endpoints.
* Optimizar el rendimiento de consultas a la base de datos.

### Sprint 4: Historial de Actividad y Movimientos (Semana 7 - 8)

* Implementar **historial de transacciones** (ingresos y egresos).
* Posibilidad de **filtrar transacciones** por tipo y fecha.
* Mejorar seguridad y manejo de errores.
* Desplegar en AWS y documentar infraestructura.

## 3. Informes de Entrega

Cada sprint finalizará con una revisión donde se evaluará:

* Funcionalidades completadas.
* Cobertura de pruebas.

## 4. Infraestructura Utilizada

El proyecto se desplegará utilizando:

* **Backend:** Java con Spring Boot, Spring Data y Hibernate.
* **Base de Datos:** MySQL.
* **Autenticación:** JWT.
* **Pruebas Automatizadas:** RestAssured.
* **Infraestructura:**
  + **Frontend:** Desplegado en Vercel.
  + **Almacenamiento:** AWS EC2 para el backend.
  + **Base de datos:** Instancia en AWS RDS.
  + **Docker** para contenerización y despliegue local.

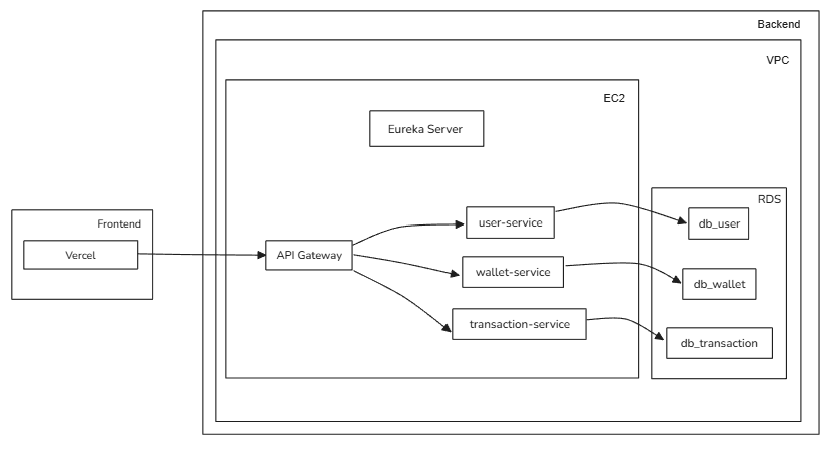
## 5. Retro Personal y Lecciones Aprendidas

Durante el desarrollo del proyecto, se recopilarán experiencias y mejoras en buenas prácticas en el desarrollo de microservicios, optimizaciones y estrategias.

**6. Diseño de la infraestructura necesaria**

Para el correcto funcionamiento del proyecto en arquitectura de microservicios, se requiere la siguiente infraestructura lógica:

* API Gateway: Punto único de entrada para las peticiones, que enruta las solicitudes hacia los microservicios correspondientes.
* Eureka Server (Service Discovery): Servicio de descubrimiento que permite a los microservicios registrarse y descubrir a otros.
* Microservicios independientes:
  + user-service: Maneja la autenticación y datos de usuario.
  + wallet-service: Gestión de cuentas y saldos.
  + transaction-service: Procesa depósitos, transferencias y movimientos.
* Base de datos por microservicio: Cada servicio mantiene su propia base de datos (MySQL), siguiendo el principio de base de datos descentralizada.
* Docker: Contenedores para cada componente, permitiendo su despliegue y ejecución consistente en cualquier entorno.
* Docker Compose: Orquestador para levantar todos los servicios localmente con un solo comando.



**INFORMES**

**SPRINT 1**

**Informe de entrega:**

Pude completar todas las funcionalidades pautadas en el backlog para este sprint, es decir, registro de usuario, inicio de sesión y cierre de la misma. Además, pude ejecutar los casos de prueba creados y lograr una aprobación del 100%, lo que me permite concluir que el sistema cuenta con estabilidad en esta primera etapa.

**Informe de retrospectiva:**

Analizando este primer sprint, llegué a la conclusión de que siempre debo tener presente que lo más importante es avanzar de a pasos. Al intentar hacer todo de una, terminé malgastando tiempo. Esto me pasó por querer empezar a lo grande en vez de avanzar, probar y ajustar.

### SPRINT 2

**Informe de entrega:**En este sprint logré implementar el acceso seguro a endpoints mediante JWT, la protección de rutas críticas, el inicio de pruebas con tokens válidos y la integración con la lógica de logout. También se completó la creación de tarjetas vinculadas a cuentas y se automatizaron los casos de prueba correspondientes.

**Informe de retrospectiva:**Me di cuenta de la importancia de validar no solo los flujos "felices" sino también los errores. Comencé a probar escenarios extremos y errores comunes, y eso ayudó a robustecer mi backend. Además, entendí que la documentación técnica clara me ahorra tiempo cuando retomo funcionalidades.

### SPRINT 3

**Informe de entrega:**

Este sprint estuvo centrado en conectar los microservicios e integrar la lógica necesaria para el dashboard del usuario, incluyendo la visualización de saldo, alias, CVU y últimas transacciones. También se sumó la edición del perfil, ingreso de dinero y pruebas exploratorias. Se cubrieron todos los requerimientos y se documentaron los resultados de forma exhaustiva.

**Informe de retrospectiva:**

Sentí mayor fluidez al trabajar, ya que había consolidado las bases del proyecto. Aprendí a delegar responsabilidades entre servicios y mantener una arquitectura limpia. El test exploratorio fue clave para detectar pequeños detalles, y reforcé la importancia de probar desde el punto de vista del usuario final.

### SPRINT 4

**Informe de entrega:**

Este sprint se centró en la revisión final de los requerimientos del proyecto y en la validación del cumplimiento de los mismos, se ejecutaron test (todos aprobados) y se pulieron los documentos finales.

**Informe de retrospectiva:**

Durante el proyecto comprendí la importancia de una clara documentación, incluso a nivel de frontend. Es fundamental que antes de comenzar un proyecto se dejen claras las necesidades a cubrir y cómo se hará.