

# PROYECTO FINAL

## MASTER SPRING BOOT CON MICROSERVICIOS

### VISION DE ARQUITECTURA

Para abordar y cubrir las necesidades complejas de los sistemas en el sector financiero en este paper se propone una solución de arquitectura de microservicios para Mi Banco con un sistema de gestión de usuarios y cuentas. El objetivo nuestro es el de ofrecer una plataforma segura, escalable y eficiente para sus clientes.

### DECISIONES DE ARQUITECTURA:

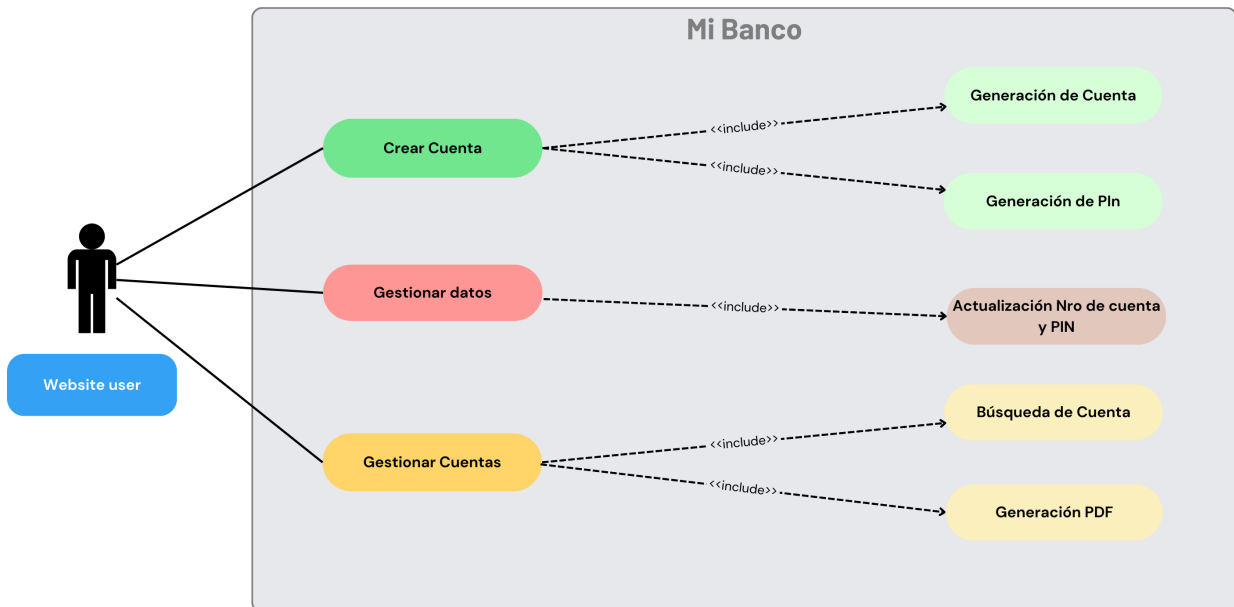
1. **Comunicación entre Microservicios:** Se utilizará el protocolo REST para la comunicación entre los microservicios, permitiendo una integración flexible y ligera. Además, se implementará un sistema de mensajería asíncrona para garantizar la tolerancia a fallos y la escalabilidad.
2. **Seguridad:** La seguridad es una prioridad fundamental en el sistema. Se emplearán tokens JWT firmados para la autenticación y autorización entre microservicios, también implementaremos medidas adecuadas para proteger la integridad de la información con Spring Cloud Config Server.
3. **Escalabilidad y Alta Disponibilidad:** La arquitectura se diseñará para escalar horizontalmente según la demanda, utilizando contenedores Docker y orquestación con Kubernetes. Se implementarán estrategias de división de carga y redundancia para garantizar la disponibilidad del sistema.
4. **Despliegue y Orquestación:** El despliegue se realizará utilizando Docker Compose que nos permitirá tener un mejor control de todos nuestros servicios y para orquestarlos utilizaremos Kubernetes que nos ayudará a gestionar el ciclo de vida de los diferentes microservicios. Sin dejar de lado a Config Server para que cada microservicio obtenga su configuración de manera centralizada y dinámica.
5. **Monitoreo y Métricas:** Se utilizarán herramientas de monitoreo como Prometheus y Grafana para supervisar el rendimiento y la salud de los microservicios. Se definirán métricas clave, como el tiempo de respuesta y la tasa de error, para evaluar el rendimiento del sistema.
6. **Generación de Informes:** El informe a entregar al cliente cuando lo solicite se generará con Thymeleaf.

### MICROSERVICIOS Y FUNCIONALIDAD:

- **Microservicio de Gestión de Usuarios:** Responsable de la creación y actualización de los usuarios, así como el almacenamiento seguro de información personal.
- **Microservicio de Gestión de Cuentas:** Permite el usuario visualizar la información de su cuenta, y genera un informe en PDF.

- **Microservicio de Autenticación**: Encargado de manejar el proceso de inicio de sesión y la generación de tokens JWT para la autorización.

## Registro y gestión de usuarios



### Instrucciones para la ejecución local de los microservicios con Docker.

No se finalizó el funcionamiento como tal de la aplicación, he tenido problemas con la aplicación de Prometheus y Mysql pero he dejado el consumo inicial para la bienvenida

### Rutas de API, parámetros y ejemplos de uso. Sugerencia: Swagger

### Políticas de seguridad y autenticación - Opcional

### PRUEBAS

Tenía la intención de aplicar pruebas de validación de datos para verificar el documento ingresado y validación de datos requeridos como usuario, contraseña y cédula.