**Documentación de Arquitectura de Software**

**1. Descripción General del Sistema**

El presente documento describe la arquitectura de software de un API que proporciona funcionalidades para realizar conversiones de tipo de cambio, así como también permite el registro, actualización y búsqueda de tipos de cambio. La autenticación se realiza a través de JWT y se registra la auditoría funcional para cada solicitud de tipo de cambio.

**2. Arquitectura de Alto Nivel**

La arquitectura del sistema se basa en un enfoque de microservicios, utilizando Spring Boot como framework principal. Se implementa una arquitectura reactiva para manejar eficientemente las solicitudes concurrentes. La persistencia de datos se realiza utilizando H2, una base de datos en memoria para simplificar el desarrollo y las pruebas. El sistema se asegura mediante JWT para la autenticación de usuarios. Las solicitudes al API pueden realizarse a través de Postman.

**3. Componentes del Sistema**

**Controlador de Tipo de Cambio**

* Descripción: Este componente maneja las solicitudes HTTP relacionadas con los tipos de cambio, incluyendo la conversión, registro, actualización y búsqueda.
* Tecnologías Utilizadas: Spring WebFlux para la gestión de solicitudes HTTP reactivas.

**Servicio de Tipo de Cambio**

* Descripción: Este componente contiene la lógica de negocio para realizar las operaciones de tipo de cambio, como la conversión y la gestión de registros en la base de datos.
* Tecnologías Utilizadas: Spring Data para la interacción con la base de datos H2

**Seguridad JWT**

* Descripción: Este componente proporciona la seguridad mediante la autenticación de usuarios utilizando tokens JWT.
* Tecnologías Utilizadas: Spring Security con JWT para la autenticación y autorización.

**Base de Datos H2**

* Descripción: Esta base de datos en memoria se utiliza para almacenar la información de los tipos de cambio y los registros de auditoría.
* Tecnologías Utilizadas: H2 Database Engine para la persistencia de datos.

**4. Patrones y Decisiones Arquitectónicas**

La arquitectura del sistema sigue el patrón de microservicios para la modularidad y escalabilidad. Se utiliza la programación reactiva para manejar eficientemente las solicitudes concurrentes. La seguridad se implementa mediante JWT para la autenticación de usuarios.

**5. Interfaces y Comunicación**

El API expone endpoints HTTP para interactuar con los servicios de tipo de cambio. Las solicitudes se realizan utilizando JSON para el intercambio de datos.

**6. Seguridad**

La seguridad se gestiona mediante la autenticación JWT, que proporciona tokens seguros para la autorización de usuarios.

**7. Versionamiento y Control de Cambios**

Se utiliza un sistema de control de versiones como Git para gestionar y documentar los cambios en el código y la arquitectura del sistema.