|  |
| --- |
| ASSIST |
| Prueba técnica |
| Desarrollo de un sistema de gestión de cuentas bancarias utilizando Lambda conPython, DynamoDB y microservicios con Java Spring Boot |

|  |
| --- |
| **ESPINOSA CAMILO ANIBAL**  22-6-2023 |

**ESTRUCTURA DE DATOS**

**Diccionario de datos**

|  |  |
| --- | --- |
| **User** | |
| **Atributo** | **Descripción** |
| idUsuario | Identificador único |
| cedula | Identificación personal |
| Nombre | Nombre del usuario |
| Apellido | Apellido del usuario |
| Direccion | Dirección del usuario |

|  |  |
| --- | --- |
| **CuentaBancaria** | |
| **Atributo** | **Descripción** |
| idCuenta | Número de cuenta |
| idUsuario | Identificador del usuario (TBUser) |
| saldo | Saldo de la cuenta |
| estado | Estado de la cuenta (1,0) |

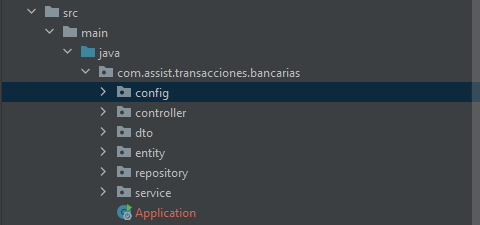
|  |  |
| --- | --- |
| **Transaccion** | |
| **Atributo** | **Descripción** |
| idTransaccion | Identificador único |
| idCuenta | Cuenta asociada a la transacción |
| usuario | Datos del usuario que hizo la transacción |
| origen | Identificador de el origen de la transacción |
| tipo | Tipo de origen de la transacción (1-deposito, 2-retiro, 3-transferencia) |
| valor | Monto de la transacción |

**PROCESO DE DESARROLLO, DECISIONES DE DISEÑO Y TECNOLOGÍAS UTILIZADAS.**

Para el desarrollo de la prueba técnica, además de las tecnologías mencionadas en el documento, se implementó el proyecto bajo la tecnología Maven ya que ofrece ventajas como la gestión de dependencias, un ciclo de vida de proyecto definido, acceso a un repositorio de biblioteca central, una estructura de directorios estándar y la capacidad de personalizar y ampliar el proyecto, Maven simplifica la construcción y gestión de proyectos Java, facilitando el desarrollo y asegurando que estén disponibles las dependencias necesarias.

Se implementó java 17 para manejar las ultimas actualizaciones donde se evidencia estandarización de la API de tiempo y actualizaciones de seguridad. Estas mejoras en Java 17 permiten escribir código más conciso, legible y eficiente, y proporcionan un mejor rendimiento y estabilidad en las aplicaciones. La elección de la versión de Java a utilizar depende de los requisitos y compatibilidad específicos del proyecto.

El modelo elegido para la organización de proyecto fue el basado en capas donde Cada capa tiene su propia carpeta, y dentro de ellas se agrupan las clases relacionadas.



Fueron utilizadas librerías de springboot tales como lombok, spring-data-dynamodb, aws-java-sdk-dynamodb, spring-boot-starter-web, aws-java-sdk-bom para facilitar el desarrollo y manipulación de datos en el proyecto en cuanto al lambda se utilizó la librería de boto3 para el acceso a la base de datos.

Para la integración de los microservicios con la lambda se creó un api Gateway que hace la conexión entre el proyecto en springboot hacia aws para procesar los datos de las transferencias entre cuentas bancarias.