**Documentacion del servicio lb-source-svc**

**Nombre: Ricardo Salvador Rivas Franco**

1. **Introduccion**

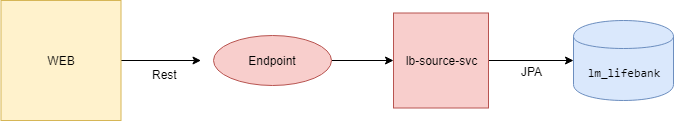
Se solicito la creacion de una plataforma online para una nueva entidad bancaría llamada LifeBank la cual se encargara de:

* Consulta de estado de productos bancarios
* Transferencia entre cuentas bancarias propias
* Transferencia a terceros
* Pago tarjeta de crédito propia
* Pago tarjeta de crédito a terceros
* Pago de préstamo bancario
* Pago de préstamo bancario a terceros

Para ello procedio a diseños de la solucion haciendo uso de las siguientes tecnologias:

* Lenguage de programacion Java
* Arquitectura de microservicios
* Framework Spring boot
* GitHub como repositorio
* Git Kraken para llevar un control de cambio en el repositorio
* Postman para realizar pruebas del servicio.
* Base de datos PostgreSQL

1. **Diagrama Arquitectonico**

****

El diseño de la solucion consta de la siguiente manera, el cliente accedera al sitio web que a su vez este accedera al servicio lb-source-svc a traves de llamados de tipo REST a los siguientes endpoint:

**login**: Autentica al usuario y generarle un toke para poder nevegar a traves de la plataforma online.

**products**: Obtiene los productos que el cliente a contratado con el banco.

**transactions:** Obtiene el listado de transacciones que el cliente ha realizado con uno determinado producto.  
**transaction-p:** Realiza transacciones bancarias entre productos propios de un cliente, recibe cuenta origen(cuenta de ahorros) y cuenta destino (cuenta de ahorro, prestamo y tarjeta de credito) , tipo del producto y el monto a transferir.

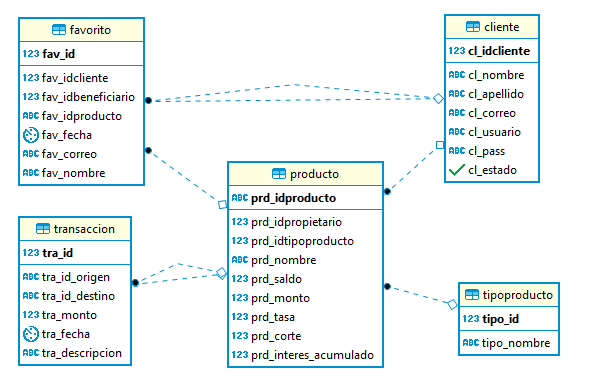
**transaction-t:** Realiza transacciones bancarias a terceros, es decir a productos de otros clientes, siempre y cuando estos esten afiliados, recibe cuenta origen(cuenta de ahorros) e identificador del beneficiario, tipo del producto y el monto a transferir.

**update-mail:** Actualiza el correo electronico de un beneficiario afiliado a un cliente

**add-beneficiary:** Agrega un beneficiario a la lista de un cliente y lo asocia a una cuenta.

**delete-beneficiary:** Elimina un beneficiario de la lista.

1. **Base de datos**



|  |  |
| --- | --- |
| Tabla | Descripción |
| Tipo Producto | Almacena el catálogo de productos que el banco maneja |
| Cliente | Maneja toda la información relacionada a un cliente, su id (número único),nombre, apellido, usuario de acceso y contraseña. |
| Producto | Contiene toda la información relacionada a los productos adquiridos por el cliente, entre la información más relevante está:  Tipo: tipo de producto  Saldo: es el crédito disponible en la cuenta de ahorros o tarjetas de crédito y si la cuenta es de tipo préstamo entonces es el total restante por pagar  Monto: indica el monto total del préstamo realizado y cuando el producto es tipo tarjeta de crédito es el límite asignado a la tarjeta. |
| Transaccion | En esta tabla se almacena cada transacción realizada, es decir cada transferencia entre una cuenta de ahorros a un préstamo o una tarjeta de crédito. Lleva una cuenta origen, cuenta destino, fecha de la transacción y el monto a transferir. |
| Favorito | Lleva el registro de afiliaciones de un cliente, para luego poder hacer transacciones a terceros. Está lleva el identificador de un cliente, el identificador del beneficiario y una número de cuenta (préstamo, tarjeta de crédito o cuenta de ahorros |

1. **Inicializar proyecto**

El proyecto consta de dos elementos el servicio lb-source-svc y dump3-lm-201905121708.backup, los paso a seguir para instalar el servicio es:

1. Restaurar dump3-lm-201905121708.backup en postgreSQL
2. Importar el proyecto lb-source-svc al IDE y descargar dependencias.
3. Cambiar configuraciones de conexión a la base de datos en el archivo ..\src\main\resources\application-local.yml dentro del proyecto importado.
4. Ejecutar proyecto.