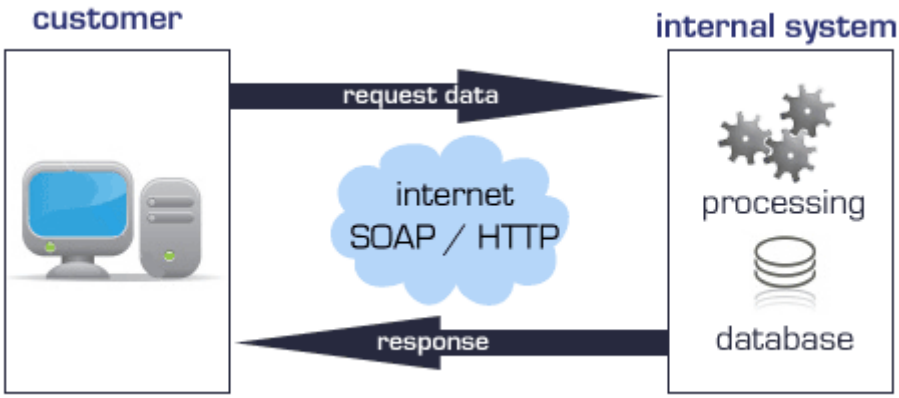
ARQUITECTURA ORIENTADA A SERVICIOS

PROFESORA: MARCELA CASTRO



David Feijoo y Mouctar Diallo

17 de diciembre de 2018

1) **Diseñar y desarrollar un servicio web ReST que se utilice para gestionar el repositorio de datos relativa a la a.**

   Grupo 1 (Mouctar/David): Gestión de ingresos y egresos a una cuenta bancaria.

2) **Definir el modelo de operaciones, indicando cada una de las operaciones, sus parámetros de entrada y de salida y una breve descripción. La definición no debería limitarse a la dificultad de la implementación, sino que se debe indicar a su criterio que funciones API se requerirán para la gestión del repositorio de datos.**

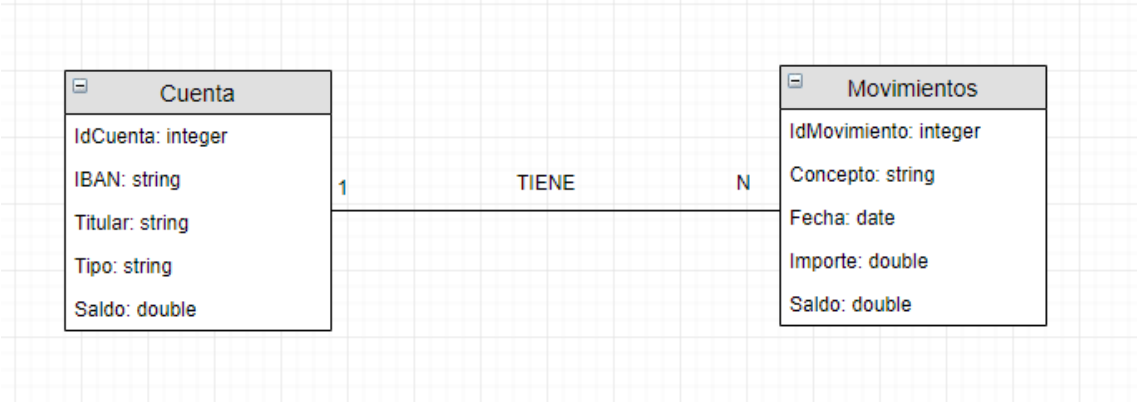
   Nuestro modelo de operaciones es el siguiente:

Descripción	Operación/URL	Parámetros de entrada	Salida OK
Alta de una cuenta	POST /cuenta	IBAN, Titular, Tipo, Saldo	IdCuenta
Modificar los atributos de una cuenta	PUT /cuenta	IBAN, Titular, Tipo, saldo	IdCuenta + IBAN, Titular, Tipo, Saldo
Listar movimientos De una cuenta	GET /cuenta/{id cuenta}/movimientos	NONE	IdCuenta, Concepto, Fecha, Importe, Saldo actual
Añadir un registro de ingreso/egreso	POST / cuenta/{idCuenta}/movimientos	Concepto, fecha, importe, saldo actual	Id_movimiento, + concepto, fecha, importe, saldo actual
Eliminar una cuenta	DELETE / cuenta/	Idcuenta	200 OK

En esta tabla de operaciones tenemos operaciones tanto de cuenta como de movimiento. En la práctica hemos implementado la clase cuenta con sus atributos y podemos realizar operaciones como dar de alta una cuenta, con la comanda POST, modificar atributos de las cuentas, eliminar una cuenta y listar los movimientos de una cuenta. También hemos implementado la relación One to many para poder crear registros de movimientos y guardarlos dentro de cuenta en una lista de movimientos.

3) Definir el modelo de entidades para implementar el servicio. Indicar las entidades y las relaciones entre ellas indicando si son relaciones 1 a 1 (OnetoOne), 1 a N (OneToMany), N a 1 (ManyToOne) o ManyToMany (N a N).

Nuestro modelo de entidades es el siguiente. Como vemos tenemos dos entidades (Cuenta y Movimientos) y tiene una relación de 1 a N (OnetoMany), es decir una cuenta tiene varios movimientos y un movimiento pertenece a una cuenta.



4) Implementar las operaciones utilizando Spring con JPA y base de datos Mysql. Si las operaciones definidas son muchas y/o complejas, se acordará con el profesor cuales implementar.

Nuestro proyecto tiene una clase Cuenta y una clase Movimientos. Cada clase tiene su controller donde definimos todas las apis que se pueden realizar y también tenemos la interface repository de cada una de las clases que es la que llamaría a las clases controller.

5) Probar las operaciones implementadas. Desarrollar y entregar un script basado en "Curl" que permita dar de alta varios registros, modificar y borrar datos.

Hemos realizado un script donde están todas las operaciones que hemos implementado. Hemos añadido nuevas cuentas, hemos modificado atributos de las cuentas y hemos eliminado cuentas. También hemos añadido registros de movimientos a una cuenta para comprobar que la relación Onetomany funcionaba de forma correcta. El script se encuentra dentro del proyecto.

Conclusión: Con esta práctica hemos aprendido a usar las relaciones para juntar clases y poder interactuar con sus apis.