## Microservicios con Spring y Maven

En este espacio, buscamos mostrar lo más similar a un paso a paso sobre cómo fuimos planeando y llevando a cabo este proyecto de finanzas.

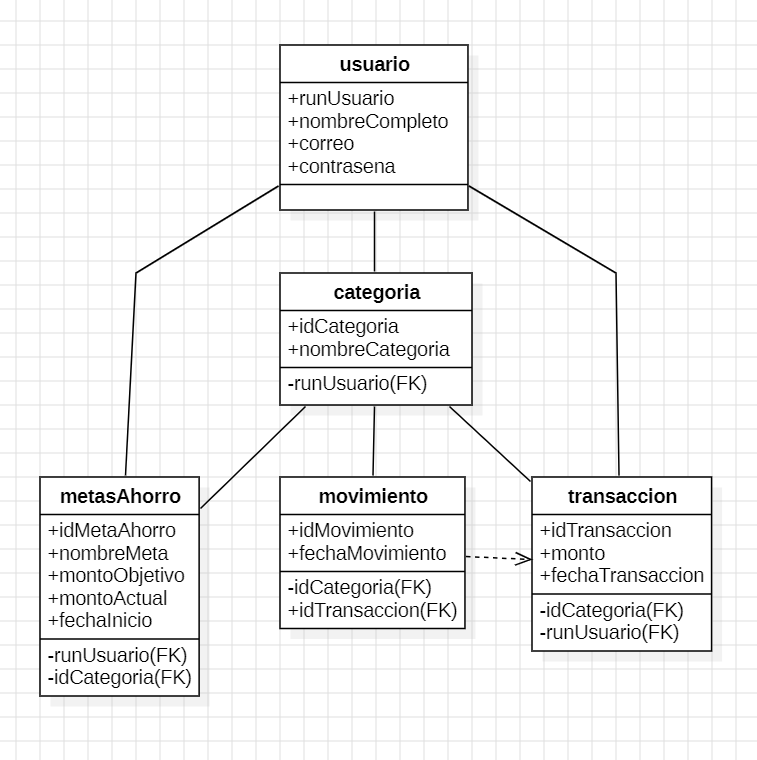
### A. Implementación de Microservicios:

En los microservicios que creamos se encuentran y destacan la existencia de «categorías» que permite seleccionar, opción que se encarga de presentar las acciones y disponibilidades de nuestro sistema; el poder efectuar una «transacción», lugar que como su nombre lo dice, es donde podemos transaccionar dinero enviándolo a cualquier meta de ahorro o persona estimada que uno quiera y requiera; el historial de «movimientos», en este lugar somos capaces de ver cada uno de los movimientos que ha tenido el dinero de los usuarios dentro del sistema, pudiendo conocer la fecha, estimación y monto utilizado en el movimiento monetario requerido; y por último, la opción de crear y organizar «metas de ahorro», que como su nombre lo dice, permite crear y personalizar tanto como se pueda cualquier meta de ahorro que uno quiera, donde uno puede delimitar la cantidad de dinero que uno espera acumular.

La comunicación entre ellas utilizando APIs RESTfull fue posible gracias a la existencia de métodos como Get, dónde obtenemos datos almacenados en la base de datos; con el método Post enviamos datos para que sean almacenados; Put es es para modificar «todo» el contenido relacionado a un objeto, en cambio Patch funciona de forma similar pero solo para modificar una pequeña parte sobre el objeto elegido; delete permite eliminar completamente los datos de un objeto almacenado en la base de datos.

### **B. Base de Datos:**

Cada microservicio creado tiene su propia base de datos, lo que garantiza su independencia al momento de comenzar el funcionamiento de cada uno de ellos:



Garantizamos la seguridad de los datos de forma que utilizamos sentencias JPA (ej: @entity, @id, etc.) para garantizar que los datos estén completos y correctamente antes de ser almacenados.

### **C.** Configuración de las bases de datos**:**

acá calvo?