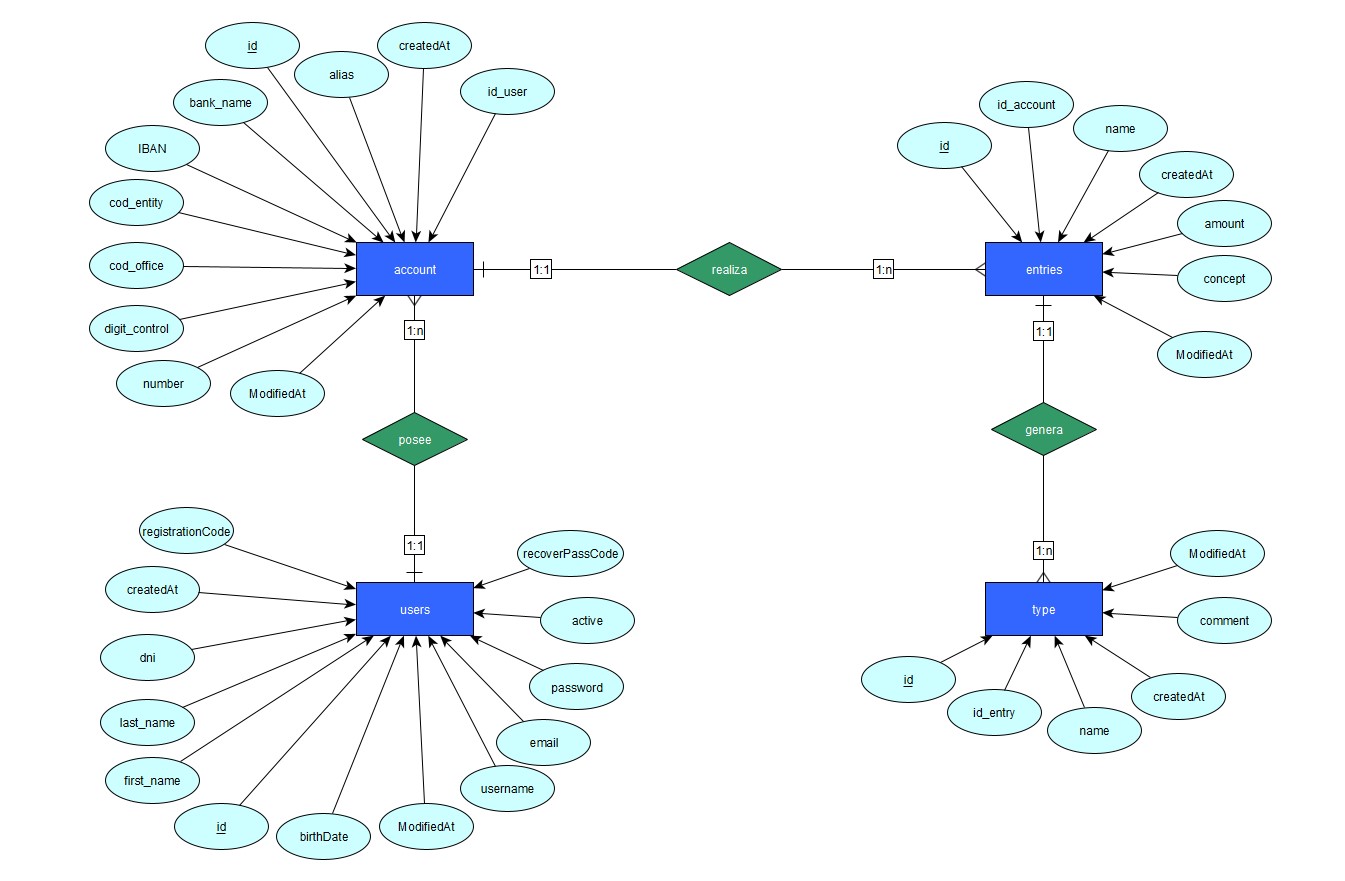
APLICACIÓN SOBRE MOVIMIENTOS DE TU CUENTA

La aplicación se llamará xpaccount, viene del inglés expenses account, gastos de mi cuenta bancaria. Consiste en gestionar los gastos e ingresos que se generan en mis cuentas bancarias.

* La aplicación permitirá añadir, editar y eliminar cuentas.
* Añadir, editar y eliminar movimientos/asientos bancarios de tu cuenta.
* Añadir asientos bancarios, tipos de movimientos según el asiento bancario.
* Indicar si es ingreso o gasto.
* Fecha del asiento.
* Conceptos, observaciones.
* Mostrar el Total de dinero en cuenta.
* Registro de usuario
* Login

# BASE DE DATOS

Para realizar la base de datos hemos utilizado la herramienta yEd Graph Editor, donde hemos creado las entidades con los siguientes atributos:



## ENTIDADES

Users:

* Id: Clave primaria, Identificador único, autoincremental y entero de cada registro.
* username: Tipo texto, VARCHAR con un máximo de 50 caracteres, requerido.
* email: Tipo texto, VARCHAR de un máximo de 100 caracteres, requerido y obligatorio.
* password: Tipo texto, VARCHAR de un máximo de 100 caracteres, requerido y obligatorio.
* recoverPassCode: Código de recuperación de contraseña: VARCHAR (20)
* registrationCode: Código de registro de usuario: VARCHAR (20)
* active: El usuario puede darse de alta pero hasta que no introduce el código de registro no se activa su cuenta. Por defecto vendrá como false. BOOLEAN.
* createdAt: Fecha de creación del usuario. Timestamp y requerido.
* modifiedAt: Fecha de modificación de alguno de los atributos. Timestamp.
* first\_name: Nombre/s del usuario. VARCHAR(100)
* last\_name: Apellido/s del usuario. VARCHAR(200)
* dni: DNI del usuario. VARCHAR(9)
* birthday: Fecha de nacimiento del usuario. Timestamp y requerido. Obligatorio ser mayor de edad.

## CUENTAS

account:

* Id: Clave primaria, Identificador único, autoincremental y entero de cada registro.
* idUser: Clave foránea de la tabla user y número entero.
* createdAt: Fecha de creación del usuario. Timestamp y requerido.
* modifiedAt: Fecha de modificación de alguno de los atributos. Timestamp.
* Alias: Nombre de la cuenta bancaria. VARCHAR (50).
* Bank\_name: Nombre del banco. VARCHAR (100)
* IBAN: International Bank Account Number code. Se compone de 2 letras y 2 números. VARCHAR(4). Obligatorio.
* Cod\_entity: Código de la entidad bancaria. Se compone de 4 dígitos. VARCHAR(4). Obligatorio.
* Cod\_office: Código de la oficina bancaria. Se compone de 4 dígitos. VARCHAR(4). Obligatorio.
* digit\_control: dígitos de control. 2 dígitos. VARCHAR(2). Obligatorio.
* Number: Número de cuenta compuesto por 10 dígitos. VARCHAR(10). Obligatorio.

## ASIENTOS

entries:

* Id: Clave primaria, Identificador único, autoincremental y entero de cada registro.
* idAccount: Clave foránea de la tabla account y número entero.
* createdAt: Fecha de creación del asiento. Timestamp y requerido.
* modifiedAt: Fecha de modificación de alguno de los atributos. Timestamp.
* name: Nombre del asiento bancario. VARCHAR (50). Obligatorio ENUM
* Amount: Cantidad asentada (ingreso o gasto). DECIMAL(9,2). Obligatorio.
* Concept: Concepto del asiento. VARCHAR(100).

## TIPOS DE ASIENTOS

type:

* Id: Clave primaria, Identificador único, autoincremental y entero de cada registro.
* idEntry: Clave foránea de la tabla entries y número entero.
* createdAt: Fecha de creación del asiento. Timestamp y requerido.
* modifiedAt: Fecha de modificación de alguno de los atributos. Timestamp.
* name: Nombre del tipo de asiento bancario. VARCHAR (50). Obligatorio ENUM
* comment: Observaciones. DECIMAL(9,2). Obligatorio.

## CREACIÓN DE TABLAS EN MYSQL WORKBENCH

Las tablas se crearán dentro de la base de datos llamada xpaccount que, a su vez, se incluirá dentro del proyecto dentro del archivo bbdd/initDB.js:

* + - CREATE DATABASE IF NOT EXISTS xpaccount;
    - USE xpaccount;

Se eliminan las tablas de forma inversa a como se han creado

* + - DROP TABLE IF EXISTS types;
    - DROP TABLE IF EXISTS entries;
    - DROP TABLE IF EXISTS accounts;
    - DROP TABLE IF EXISTS users;
    - CREATE TABLE IF NOT EXISTS users (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(100) UNIQUE NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(100) NOT NULL,

birthday TIMESTAMP,

firstName VARCHAR(100),

lastName VARCHAR(200),

dni VARCHAR(9),

registrationCode VARCHAR(100),

recoverPassCode VARCHAR(20),

active BOOLEAN DEFAULT false,

createdAt TIMESTAMP NOT NULL,

modifiedAt TIMESTAMP);

* + - CREATE TABLE IF NOT EXISTS accounts (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

idUser INT UNSIGNED NOT NULL,

FOREIGN KEY (idUser) REFERENCES users(id),

alias VARCHAR(50),

bankName VARCHAR(100),

ibanCode VARCHAR(4) NOT NULL,

entityCode VARCHAR(4) NOT NULL,

officeCode VARCHAR(4),

digitControl VARCHAR(2),

number VARCHAR(10),

createdAt TIMESTAMP NOT NULL,

modifiedAt TIMESTAMP);

* + - CREATE TABLE IF NOT EXISTS entries (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

idAccount INT UNSIGNED NOT NULL,

FOREIGN KEY (idAccount) REFERENCES accounts(id),

name VARCHAR(100) NOT NULL,

amount DECIMAL(9,2) NOT NULL,

concept VARCHAR(100),

createdAt TIMESTAMP NOT NULL,

modifiedAt TIMESTAMP);

* + - CREATE TABLE IF NOT EXISTS entries (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

idEntry INT UNSIGNED NOT NULL,

FOREIGN KEY (idEntry) REFERENCES entries(id),

name VARCHAR(100) NOT NULL,

comment VARCHAR(200),

createdAt TIMESTAMP NOT NULL,

modifiedAt TIMESTAMP);