UNIVERSITAT DE BARCELONA

PROYECTO INTEGRADO DE SOFTWARE

Modelo de dominio

COINNOTE

AUTORES: — *NIUBS:*

MARÍA ROMÁN MARTÍN — 20222252
MIGUEL HUAYLLAS CHOQUE — 17510710
ELENA DEGRÀCIA JARQUE — 20206863
ALICIA CARRASCO GUARDIOLA — 20150513

ABRIL DE 2021

1 Modelo de dominio

En relación con nuestro modelo de dominio, hemos ido teniendo varias ideas a medida que íbamos realizando e implementando nuestra aplicación. A día de hoy, seguimos teniendo alguna que otra duda en relación a como combinarlo con Firebase, ya que en modelos anteriores de otras asignaturas, no disponíamos de una base de datos como la que estamos utilizando ahora.

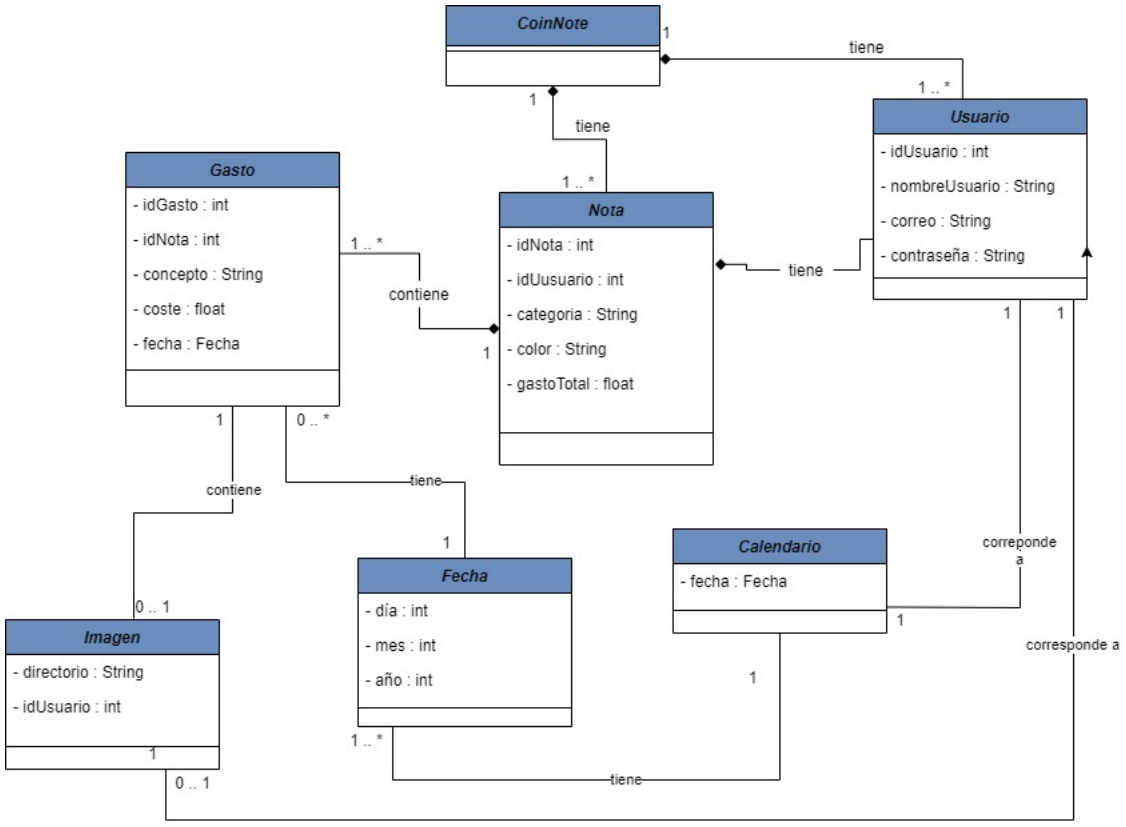

Figura 1: Modelo de Dominio

Actualmente, nuestro modelo consta de siete clases distintas. Primero tenemos la clase coinNote, que es la propia aplicación.

Seguidamente tenemos el usuario. Esta consta de un idUsuario, para poder reconocer a que usuario nos referimos. Este idUsuario se crea automáticamente cuando un cliente se registra en la aplicación. También disponemos en la misma clase de un nombre de usuario (username), un mail y contraseña. Usuario y CoinNote tienen una relación de composición, al igual que Usuario y Nota, es decir, CoinNote no existiría si no existieran usuarios y una nota no existiría si no existe un usuario. Estos dos tienen una relación en la que CoinNote puede tener entre uno e infinito usuarios y usuarios solo puede tener un coinNote. Lo mismo sucede con la clase Nota. Esta tiene la misma relación con coinNote, ya que si no hay notas, no existe la aplicación. Para esta clase tendremos tres atributos. El primero es idNotas. Este nos ayuda a poder reconocer x nota, ya que puede haber dos notas que se llamen iguales. También tendremos un atributo idUsuario, ya que cada usuario tiene sus propias notas. De esta manera podemos saber nosotros que notas mostrar dependiendo del usuario que se ha entrado. Por último tendremos una categoría, un color y un gasto total para cada nota.

Gasto es la cuarta clase que encontramos. Esta tiene un idGasto, un concepto y un coste, un objeto Fecha (explicado posteriormente) y un idNota. Este idNota nos ayudará a reconocer a que nota

pertenece un gasto. La relación que tienen Gasto y Nota es de composición, como hemos visto anteriormente. No puede existir una nota si no existe un gasto.

La clase Fecha la hemos implementado para poder crear fechas reales. Esta está constituida por tres atributos: año, mes y día. Como se ha mencionado anteriormente, la clase gasto tendrá un atributo tipo fecha. Esto significa que a cada gasto le correponde una unica fecha, mientras que una fecha puede aparecer en distintos gastos.

Seguidamente tenemos la clase calendario. Esta tiene como atributo un objeto Fecha. Como se puede observar en el diagrama, la relación que hay entre Fecha y Calendario es que un calendario contiene varias fechas mientras que una fecha solo corresponde a un calendario. También se ve una relación entre usuario y calendario, ya que a cada usuario le corresponde solo un calendario, donde podrá observar sus gastos.

Por último tenemos la clase Imagen. Para esta clase seguimos teniendo alguna duda, ya que las imágenes que tenemos en nuestra aplicación se guardan en la base de datos Firebase. Por lo tanto, hemos decidido que esta clase Imagen tenga como atributo un directorio para poder obtener la imagen pertinente de Firebase. También necesitamos saber de qué usuario queremos obtener la imagen deseada, ya que al guardar una imagen en la aplicación, Firebase crea una carpeta con el idUsuario. Para ello, hemos pensado que esta clase debería tenerlo. Esta clase está relacionada tanto con gasto como con usuario, ya que cada gasto puede tener una imagen y cada usuario puede tener una foto de perfil.