~MEMORIA~

MARCOS DE DESARROLLO

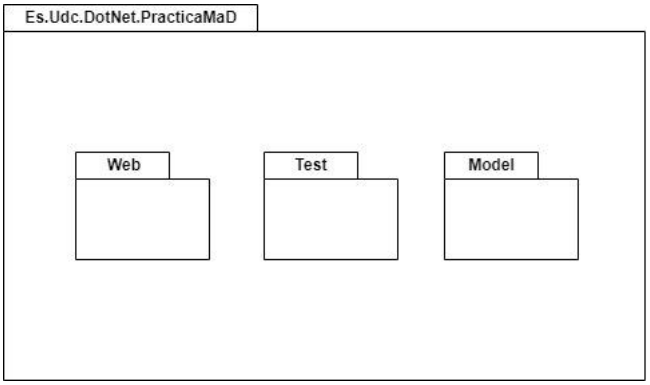
IT-2

Alejandro Viñán Bértoa
Daniel Silva Iglesias
Yago Mira Urdampilleta

MAD - 08

-ÍNDICE:

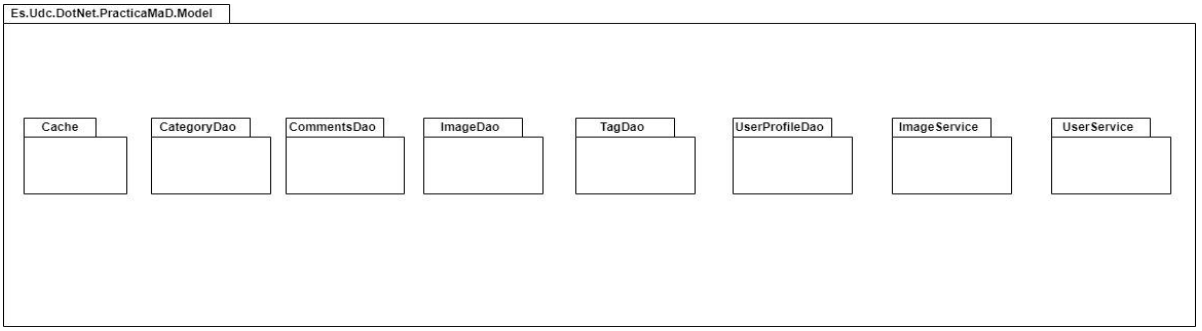
1. Arquitectura global



-Arquitectura global de la aplicación-

-La aplicación tanto en la parte modelo como en la parte web se ha llevado a cabo con la siguiente estructura de paquetes.

-En la parte del **modelo** podemos encontrar:

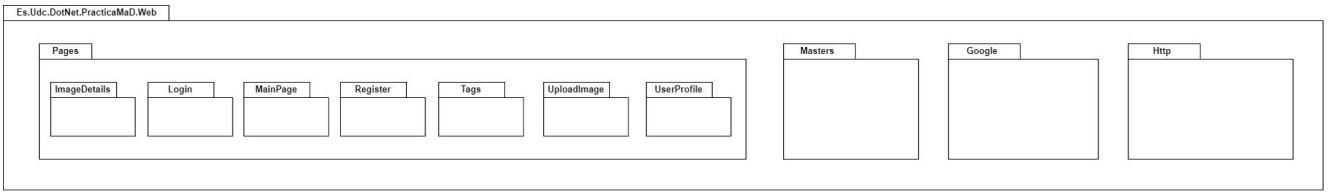


-Arquitectura Modelo-

- **Es.Udc.DotNet.PracticaMaD.Model.Cache:** contiene la clase que nos permitirá tener la caché de imágenes.
- **Es.Udc.DotNet.PracticaMaD.Model.Sql**: contiene los diferentes scripts para crear la base de datos de la aplicación SqlServerCreateDatabase.sql y, cada una de las tablas respecto a las entidades SqlServerCreateTables.sql.
- **Es.Udc.DotNet.PracticaMaD.Model.CategoryDao:** Permite realizar operaciones CRUD sobre categorías y buscar categorías por nombre.
- **Es.Udc.DotNet.PracticaMaD.Model.CommentsDaO:** Permite realizar operaciones CRUD sobre comentarios y buscarlos en función de la imagen.
- **Es.Udc.DotNet.PracticaMaD.Model.ImageDao:** Permite realizar operaciones CRUD sobre imágenes y las busquedas de imagenes necesarias para la aplicación.
- **Es.Udc.DotNet.PracticaMaD.Model.TagDao:** Permite realizar operaciones CRUD sobre tags y algunas búsquedas.
- **Es.Udc.DotNet.PracticaMaD.Model.UserProfileDao**
- **Es.Udc.DotNet.PracticaMaD.Model.ImageService:**
    - **Es.Udc.DotNet.PracticaMaD.Model.ImageService.Exceptions:** contendrá las excepciones relacionadas con el servicio ImageService (paginación, …).
- **Es.Udc.DotNet.PracticaMaD.Model.UserService**

- **Es.Udc.DotNet.PracticaMaD.Model.UserService.Exceptions:** contendrá las excepciones relacionadas con el servicio UserService (contraseña errónea, input validation, …).
- **Es.Udc.DotNet.PracticaMaD.Model.UserService.Util:** contiene clases de utilidad, como el Encrypter que permite encriptar la contraseña de usuario o PropertyValidator, que permite realizar las comprobaciones necesarias para confirmar que los datos introducidos son correctos.

-En la parte del **web** podemos encontrar:



-Arquitectura Web-

- **Es.Udc.DotNet.PracticaMaD.Web.Http:**
  - **Es.Udc.DotNet.PracticaMaD.Web.Cookies**
  - **Es.Udc.DotNet.PracticaMaD.Web.Exceptions:** le muestra al usuario la información de que no se ha encontrado ningún usuario autentificado en la sesión.
  - **Es.Udc.DotNet.PracticaMaD.Web.Session:** podemos diferenciar entre CulturePage que permite establecer la internacionalización (i18n) de la aplicación y SessionManager, que permite realizar todas aquellos casos de uso que requieran de un usuario autenticado.
  - **Es.Udc.DotNet.PracticaMaD.Web.View**

- **Es.Udc.DotNet.PracticaMaD.Web.Pages:**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.ImageDetails**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.Login**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.MainPage**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.Register**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.Tags**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UploadImage**
  - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile.Account**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile.Culture**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile.Email**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages**
      - **.UserProfile.FirstNameLastName**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile.Followers**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile.Follows**
    - **Es.Udc.DotNet.PracticaMaD.Web.Pages.UserProfile.Password**

2. Modelo

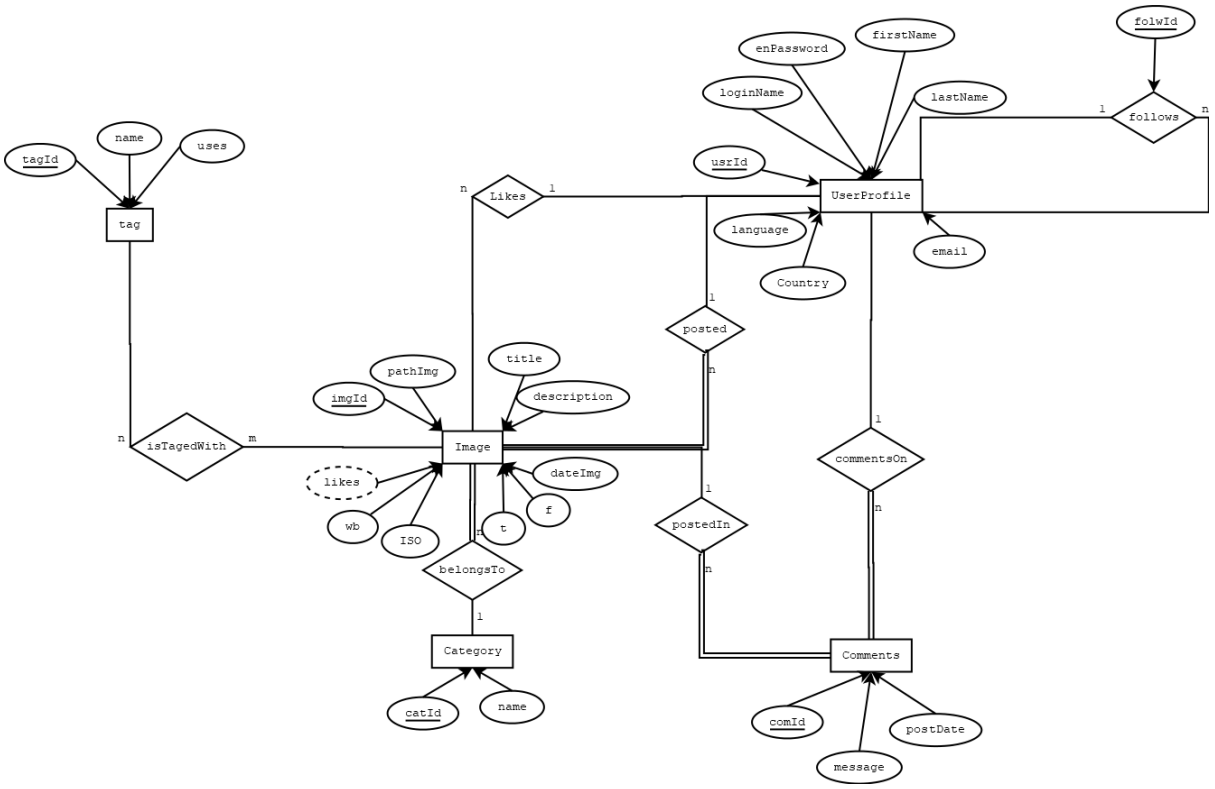
2.1 Clases Persistentes

El dominio a modelar se trata de una base de datos para una web de subida de imágenes. Para ello, se ha elaborado el ER y MR:

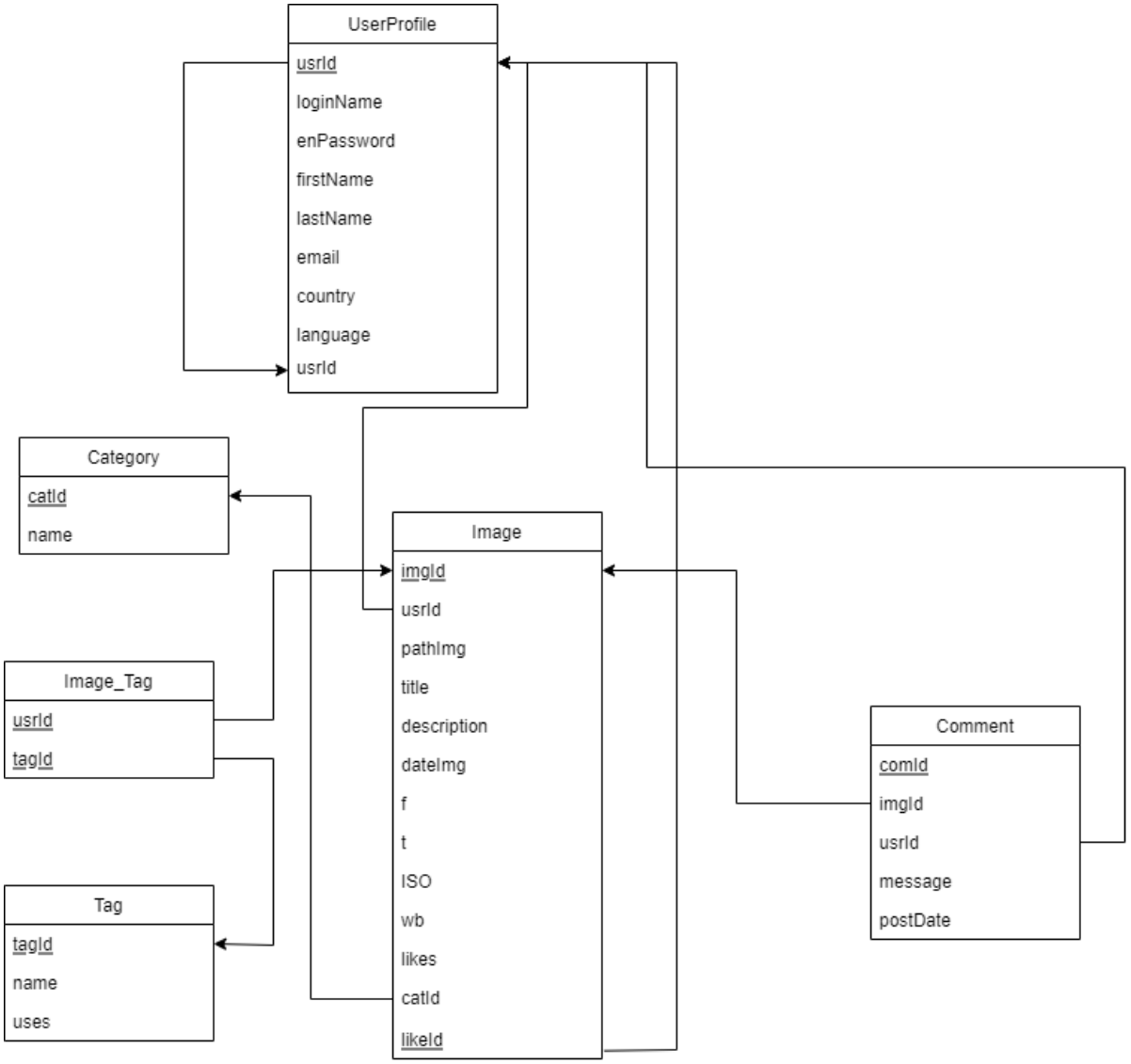
Debido a que la que las búsquedas en bases de datos a través de valores enteros son mucho más rápidas y eficaces que a través de cadenas de caracteres, a la hora de implementar la aplicación hemos añadido una serie de atributos en:

-Category: Se identifica mediante el catId.
-UserProfile: Se identifica mediante el usrId.
-Tag: Se identifica mediante el tagId.

Además del motivo antes mencionado, otra de las razones por las que se ha optado por agregar este nuevo atributo es porque las clases *IGenericDao* y *GenericDaoEntityFramework* proporcionadas para realizar esta práctica están implementadas de tal manera que únicamente aceptan un único identificador como clave primaria.

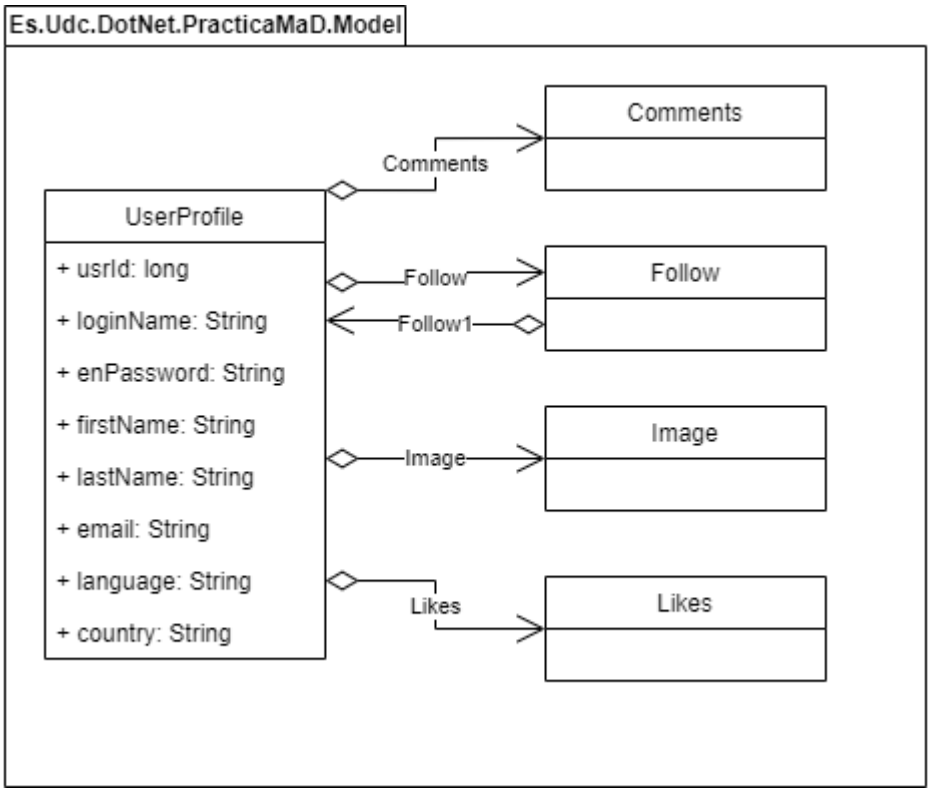


-MODELO E/R-

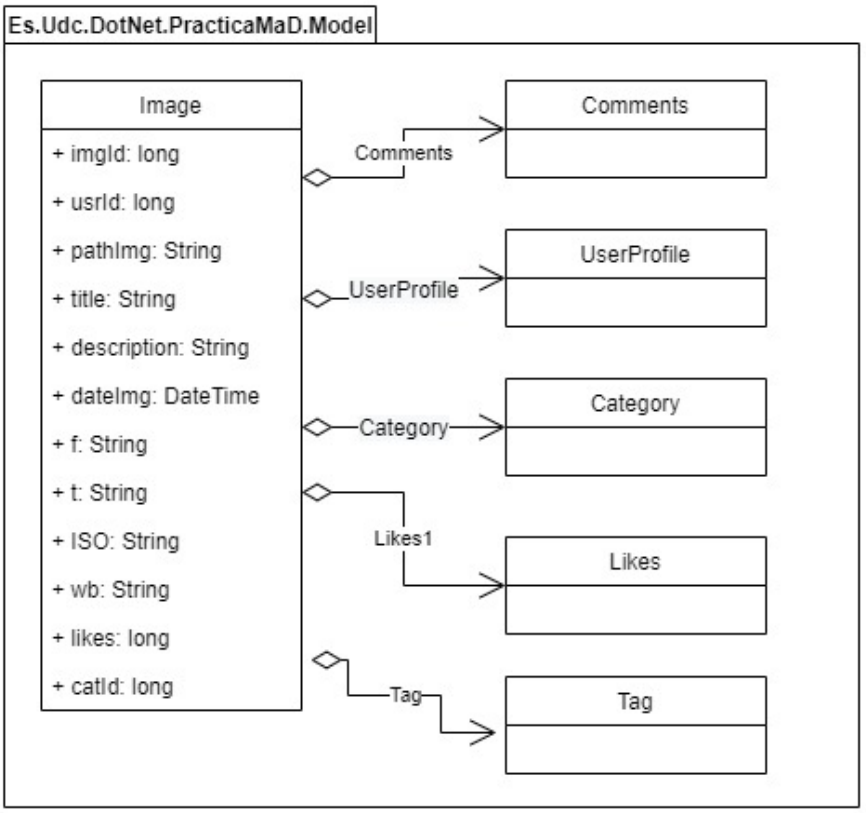


-MODELO RELACIONAL-

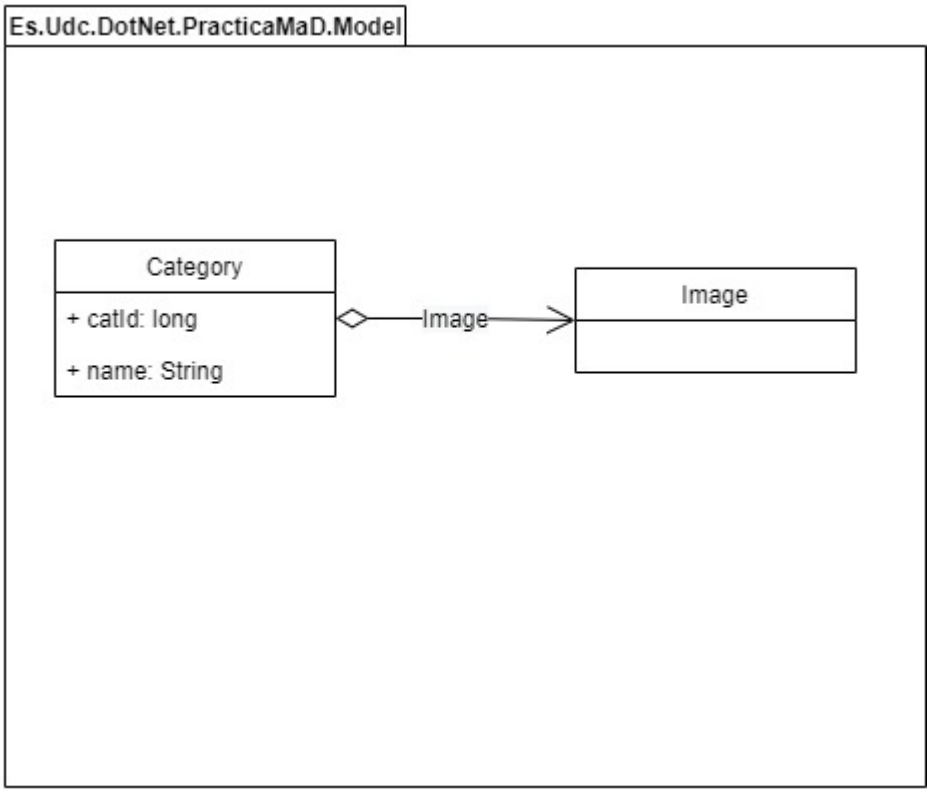
2.1.1 UserProfile



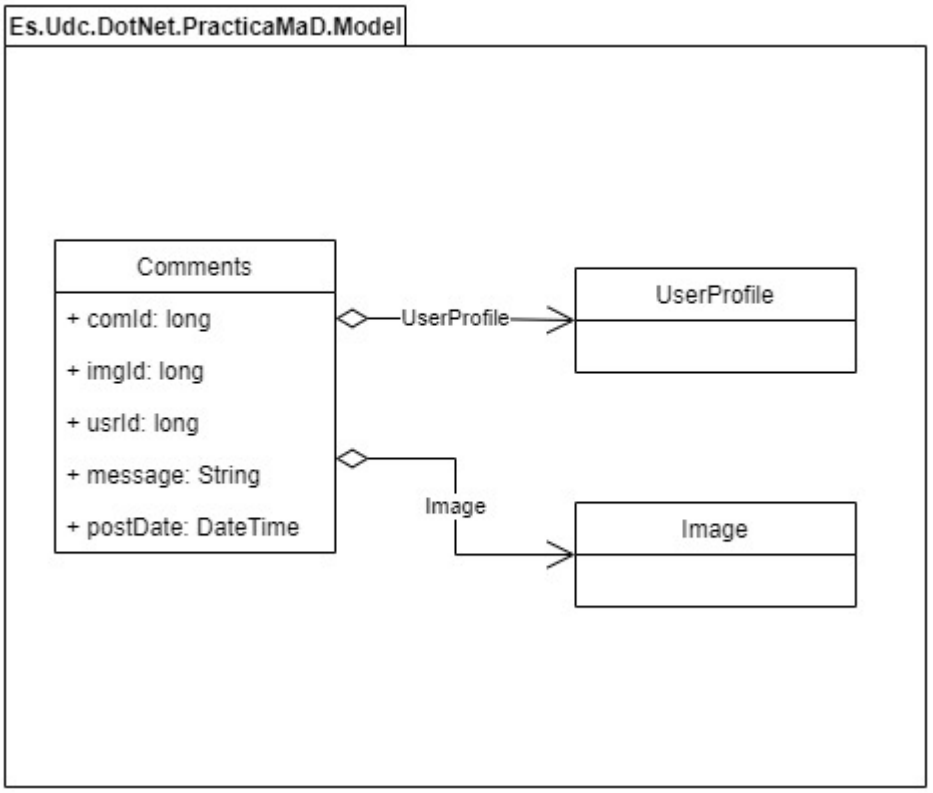
2.1.2 Image


Es.Udc.DotNet.PracticaMaD.Model
Image
+ imgId: long
+ usrId: long
+ pathImg: String
+ title: String
+ description: String
+ dateImg: DateTime
+ f: String
+ t: String
+ ISO: String
+ wb: String
+ likes: long
+ catId: long
Comments
Comments
UserProfile
UserProfile
Category
Category
Likes1
Likes
Tag
Tag

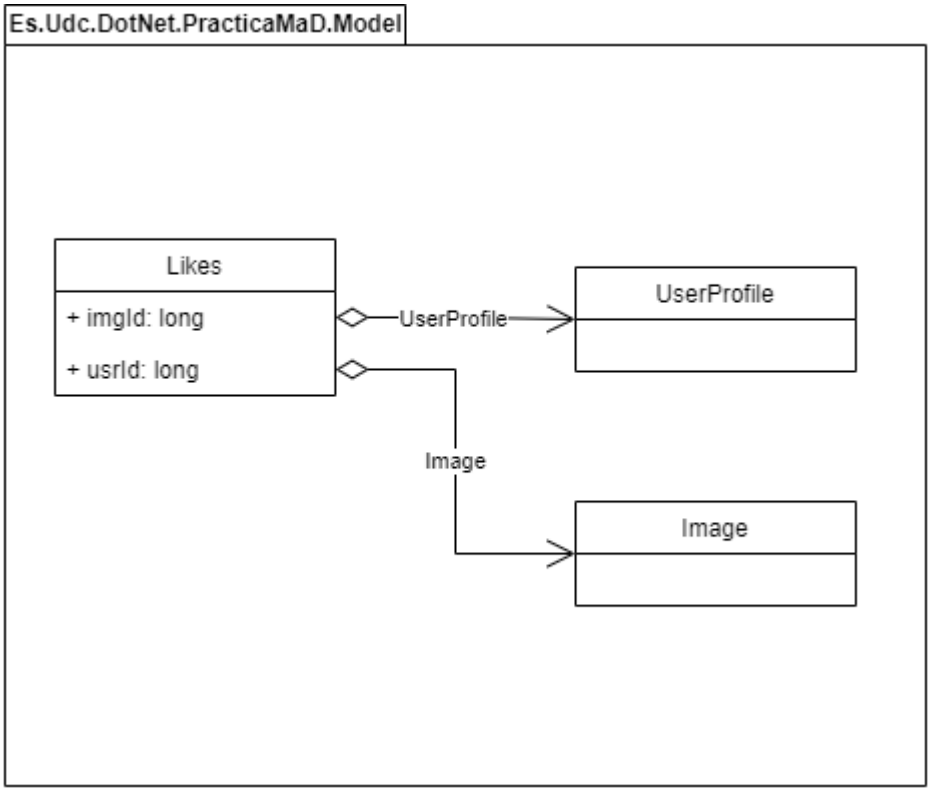
2.1.3 Category



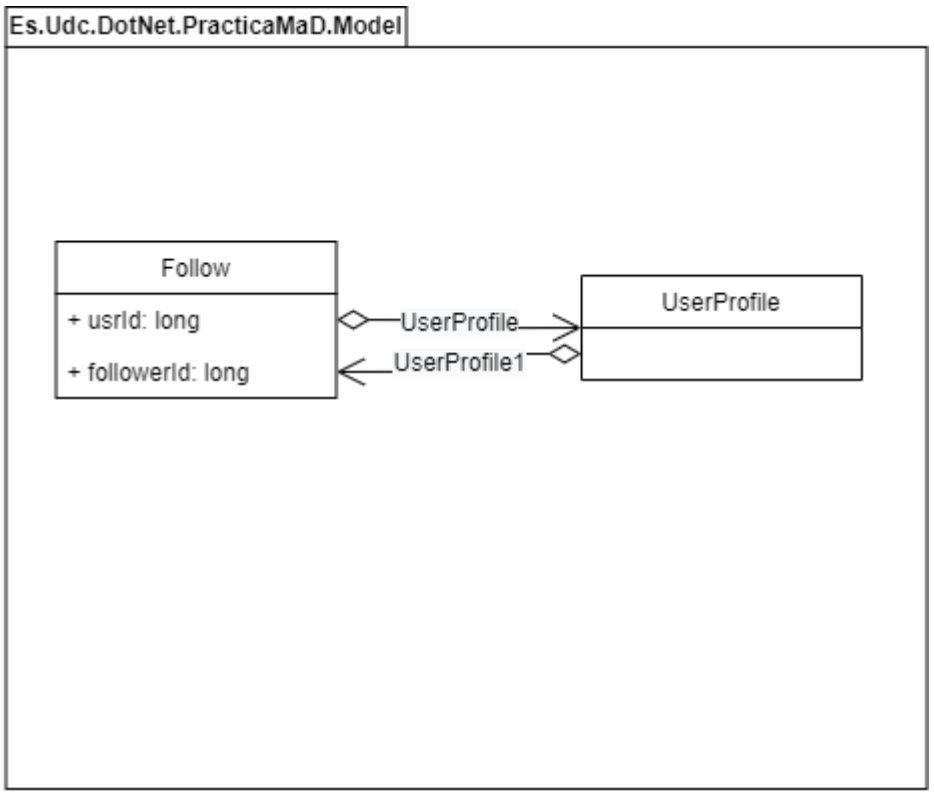
2.1.4 *Comments*



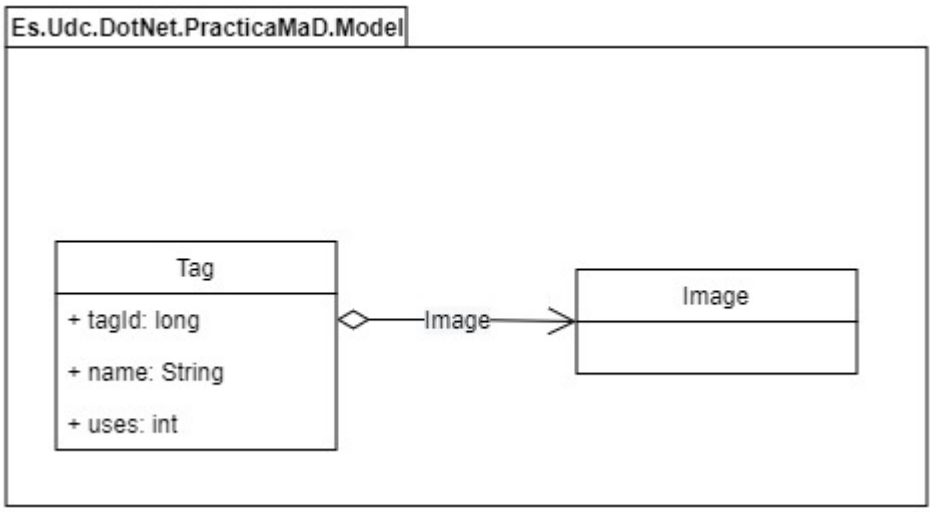
2.1.5 *Likes*



2.1.6 *Follow*



2.1.7 *Tag*



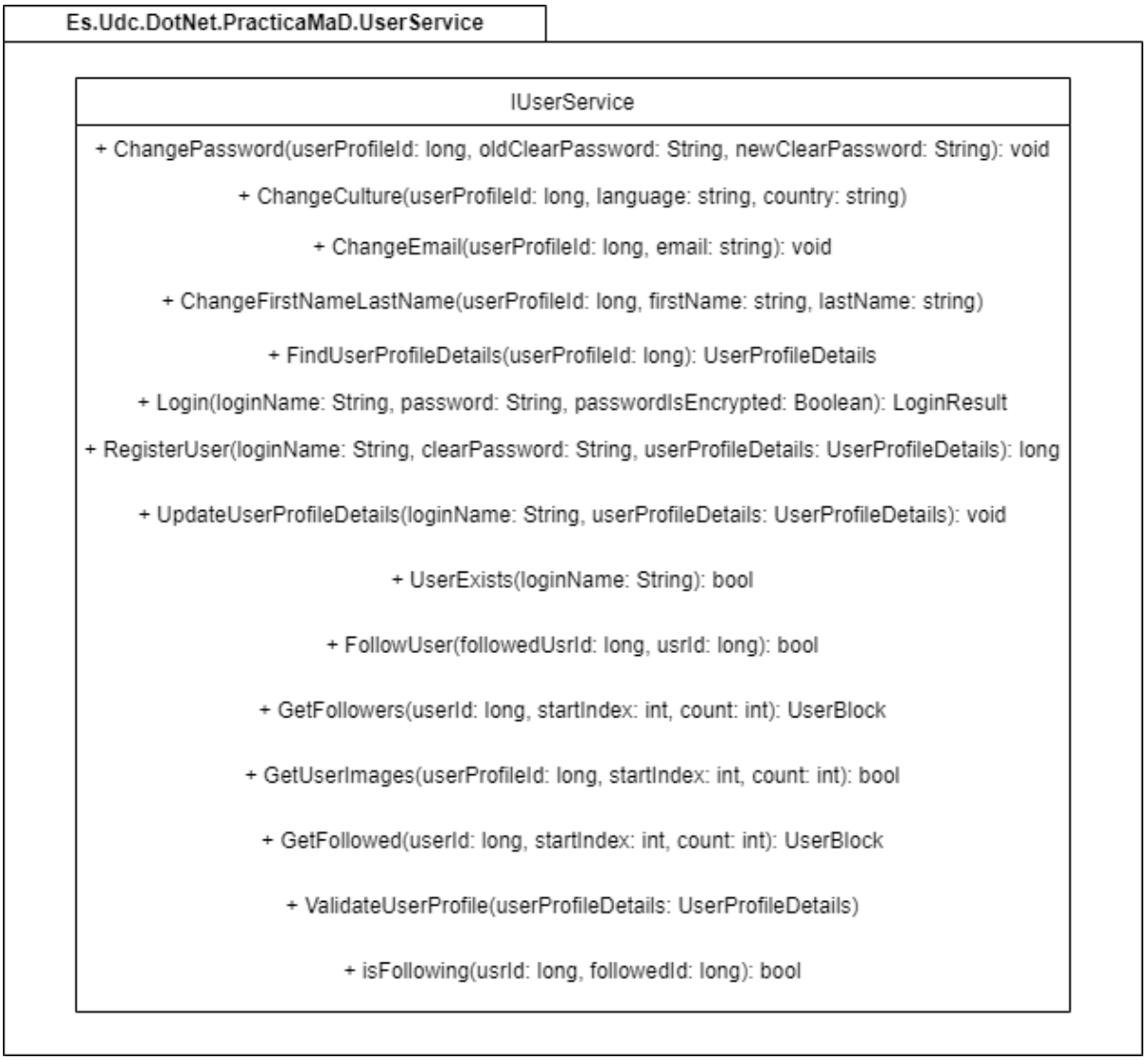
2.2 Interfaces de los servicios

2.2.1 ImageService

-Este servicio agrupa todos aquellos casos de uso que permiten manejar la información pertinente a las imágenes, junto con los tags, comentarios,... y todo aquello que permita mostrar/obtener información de una o varias imágenes.

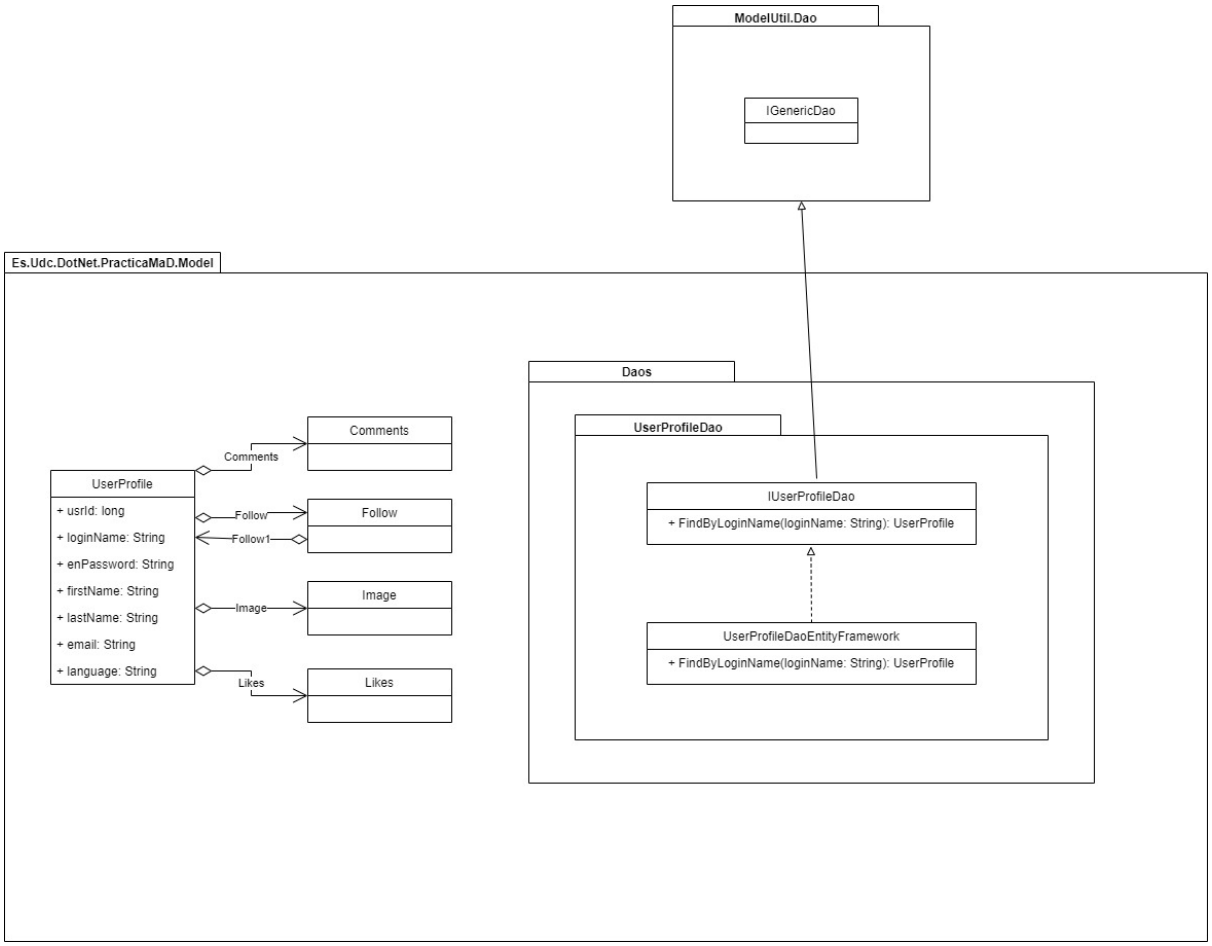
2.2.2 UserService

-Este servicio agrupa todos aquellos casos de uso que permiten manejar la información pertinente a un usuario dentro de la aplicación (datos personales, preferencias de idioma, etc). Así como los casos de uso que permiten registrarse e iniciar sesión. Además, también aquellos casos de uso relacionado con el seguir a otros usuarios.



2.3 Diseño de un DAO

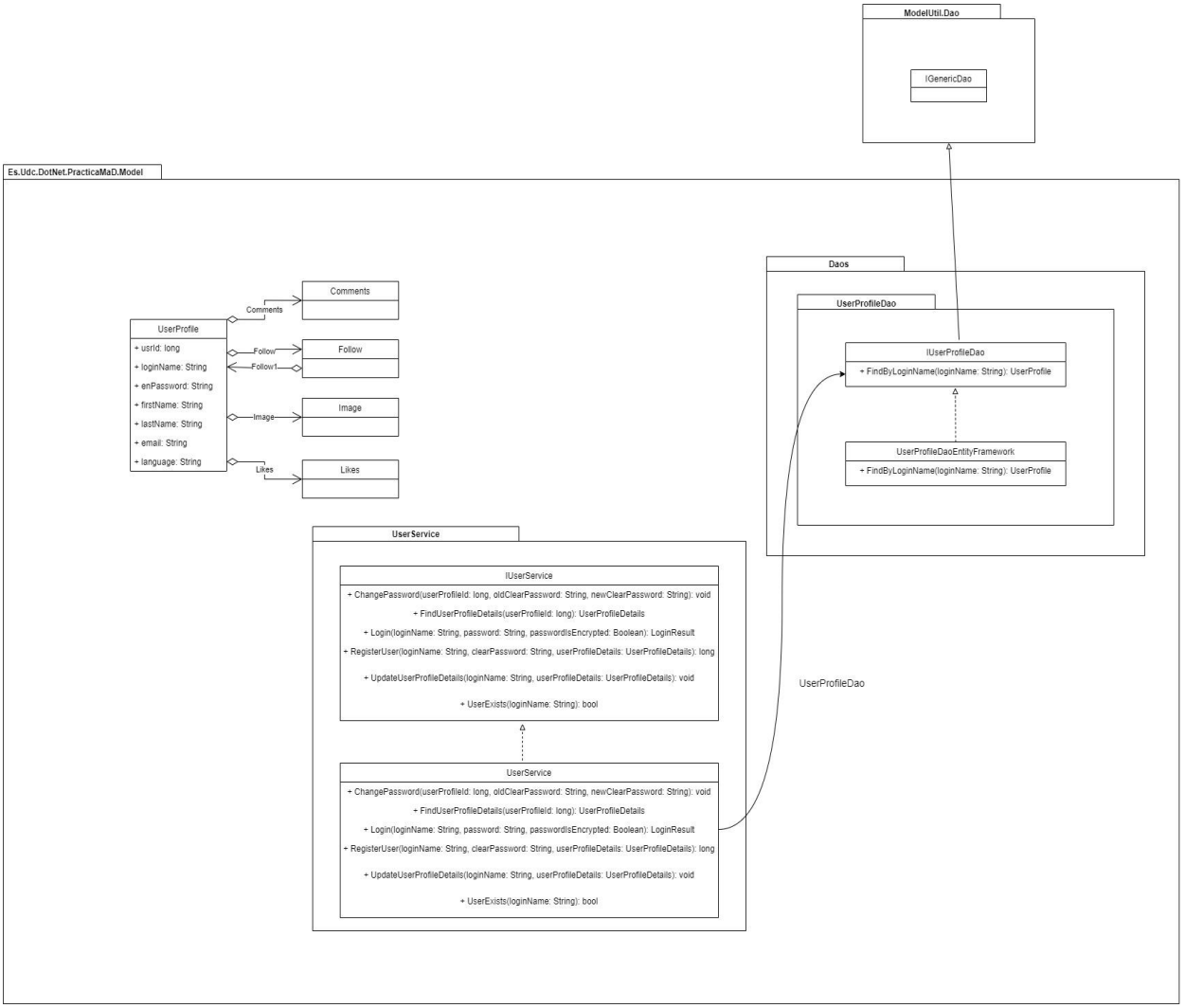
2.3.1 UserProfileDao



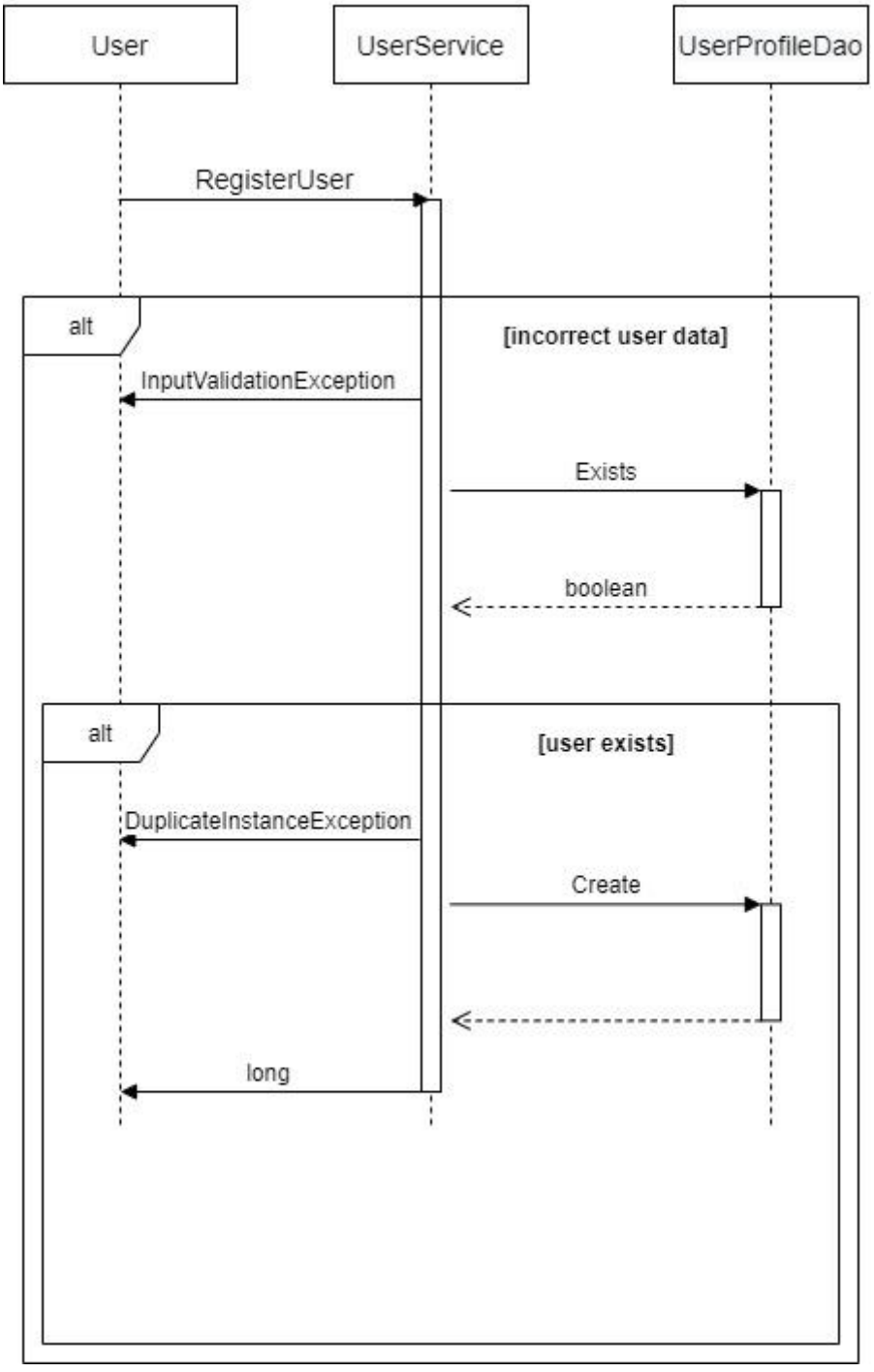
2.4 Diseño de un servicio del modelo

2.4.1 UserService

-DIAGRAMA DE CLASES-



-DIAGRAMA DE SECUENCIA-

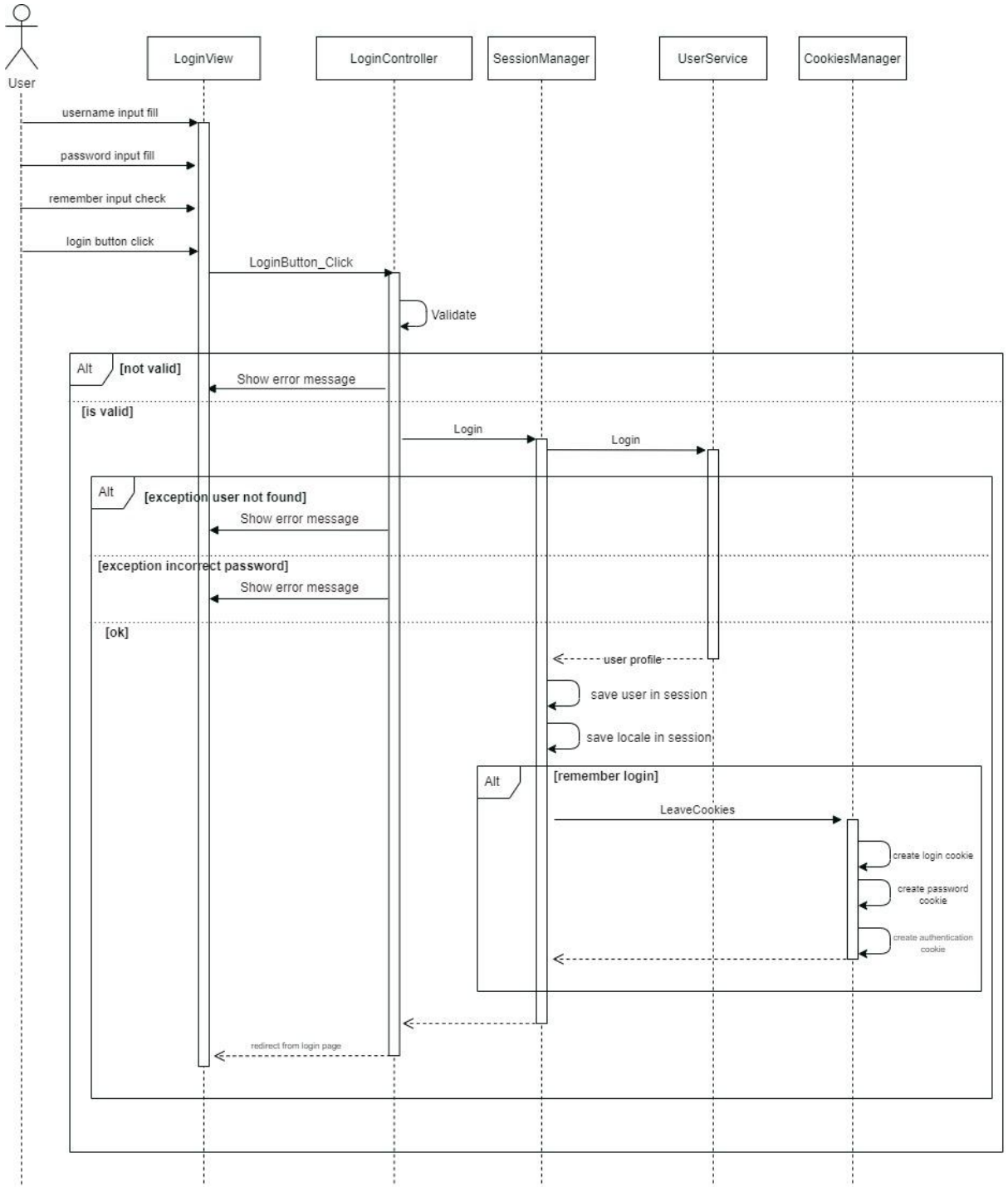


-

2.5 *Otros aspectos*

-No ha habido otros aspectos relevantes que comentar en este apartado.

3. *Interfaz Gráfica*



-Diagrama de secuencia para iniciar sesión en la interfaz-

4 Parte adicional

4.1 Tags

La funcionalidad de añadir tags a las imágenes requería agregar los siguientes de casos de uso:

-Etiquetar una imagen.
-Desetiquetar una imagen.
-Añadir una tag.
-Ver las etiquetas más usadas.
-Ver las imágenes etiquetas con un tag

Y debido con la proximidad con el servicio de imágenes se optó por incorporar estos nuevos casos de uso dentro de este servicio (ImageService). Pero estos nuevos casos de uso requerirían de una clase persistente en la base de datos (Tag), la cual será manejada a través de un DAO (ITagDao). Siendo utilizado este DAO en el ImageService.

Además, tendremos una nube de Tags en nuestra aplicación, donde mostraremos los diferentes Tags usados en la app, modificando el tamaño de los tags en la nube en función de la vez que hayan sido usados (esto lo haremos contando el número de relaciones que tendrá los tags en la bd con la tabla de Imágenes).

4.2 Cacheado

Para el cacheado de las búsquedas primero implementamos una clase general ImageCache en el paquete Cache en la que implementamos una MemoryCache de la librería System.Runtime.Caching. Esta clase es una clase estática que expone las funcionalidades de la MemoryCache a toda la aplicación.

La caché almacena datos como si fuese un mapa con los campos clave valor. En nuestro caso el valor serían las imágenes resultantes de la búsqueda y para la clave decidimos hacer una combinación de las palabras clave introducidas por el usuario y la categoría seleccionada.

Para que una caché sea efectiva tiene que tener un tamaño relativamente pequeño o sino sería más eficiente buscar los datos directamente en la base de datos, para ello limitamos el tamaño máximo de la caché a 5 búsquedas, en caso de querer añadir otro elemento a la caché el elemento más viejo en la caché se borra como si tuviese el comportamiento de una cola FIFO.

4.3 OAuth y Bootstrap

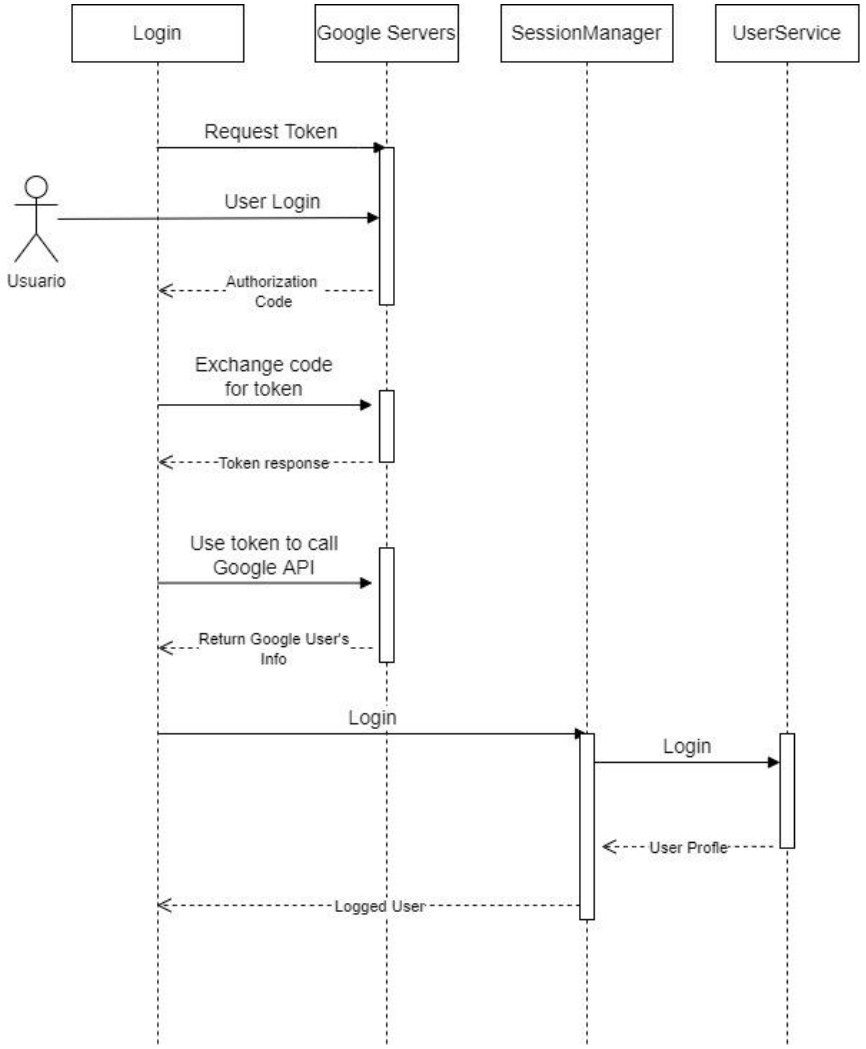
Se ha implementado la librería Bootstrap con el objetivo de mejorar la apariencia de nuestra aplicación. Más concretamente, instalamos todas las hojas de estilos, scripts, … a través del Manejador de Paquetes de Nuget. Se instalarán todos los scripts en el propio folder /Scripts. Posteriormente referenciamos en la sección “Script” de nuestro .master a los archivos JS de BootStrap. Y ocurriría lo mismo con los archivos css.

Para el desarrollo de los elementos visuales de la aplicación tomamos como referencia los ejemplos dados en la página de Bootstrap.

La funcionalidad de OAuth ha sido añadida a través de las clases del frontend relacionadas con el apartado de “Login” en donde hemos definido los valores obtenidos respecto a las credenciales de la api de OAuth donde previamente hemos creado un proyecto (https://console.cloud.google.com/).

Una vez el usuario intente autenticarse a través de esta opción, se llamará a la api de OAuth y se retornarán los valores pertinentes (email, nombre, …) los cuales almacenaremos en nuestra BD, por un lado, si el usuario no se había autenticado previamente, se registrará con estos nuevos datos, en caso contrario, se autenticará como un usuario normal.
Se ha seleccionado el servicio de **Google** para esta funcionalidad.

-DIAGRAMA DE SECUENCIA DE OAUTH-



5. Compilación e instalación de la aplicación

Lo primero clonar el proyecto de ***https://github.com/DanielS-source/mad-practica***

Es necesario tener una carpeta con la ruta C:\Software\DataBase para el almacenamiento de las bases de datos de la aplicación. En caso de querer cambiar la ruta es necesario cambiar la ruta en los siguientes ficheros a la que se prefiera: *SqlServerCreateDatabase.sql* (Modelo, linea 15), *SqlServerCreateTables.sql* (Test, linea 15), *ImageService.cs* (Modelo, Linea 45, 48 y 51)

Ahora ya se pueden ejecutar los scripts de creación de la base de datos. Para crear la base de datos es necesario ejecutar los scripts que se encuentran en el paquete Sql del Modelo llamados *SqlServerCreateDatabase.sql* y *SqlServerCreateTables.sql* en ese orden. Si se quiere ejecutar los tests es necesario ejecutar también el script *SqlServerCreateTables.sql* del paquete Sql de Test.

6. Problemas Conocidos

-No ha habido otros aspectos relevantes que comentar en este apartado.