

UNIVERSIDADE DA CORUÑA

Marcos de Desarrollo

Primera iteración

Grupo 1.2

2021/2022

**Jose Manuel Turnes Pazos
Miguel Quiroga Rodriguez
Rita Cernadas Tubío**

**jose.turnes
miguel.quiroga
rita.cernadas**

Índice

1. Arquitectura global.	2
2. Modelo.	3
2.1. Clases persistentes.	3
2.2. Interfaces de los servicios ofrecidos por el modelo.	4
2.3. Diseño de un DAO.	5
2.4. Diseño de un servicio del modelo.	5

1. Arquitectura global.

La estructura global de la aplicación está formada por 3 paquetes principales: Model, ModelTest y Web.

El paquete Model, el primero de todos los paquetes, está formado tanto por la creación de la base de datos como la creación de las tablas, el EDMX con el contenido de las entidades y su relación entre ellas, los Dao's de la aplicación (CommentDao, ImageUploadDao, PublicationDao, UserProfileDao, TagDao y CategoryDao) y por último pero no por ello menos importante los Servicios necesarios para la elaboración de la aplicación (CommentService, ImageUploadService, PublicationService, UserProfileService y TagService).

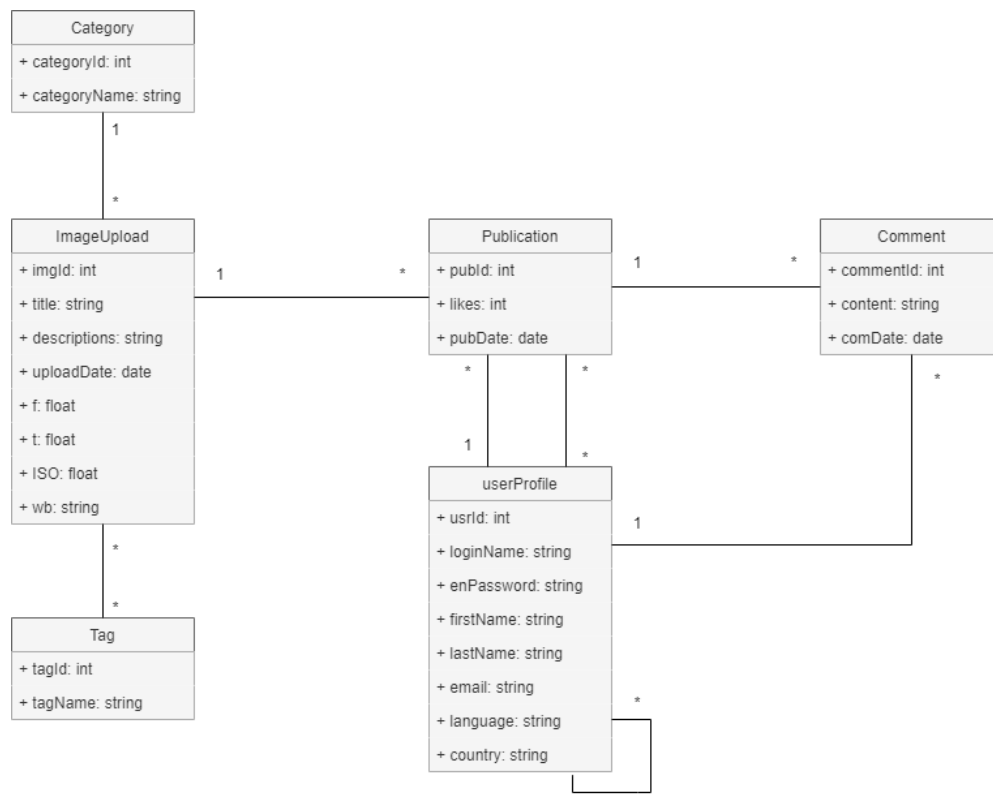
El paquete ModelTest, se encarga de comprobar si todas las funcionalidades implementadas en los Servicios de la capa Model están funcionando correctamente. Para ello, se elabora para los Servicios más relevantes los test necesarios (CommentService, ImageUploadService, PublicationService, UserProfileService). También contamos con las pruebas de NinjectXML.

Y por último contamos con el paquete Web que será desarrollado en la iteración 2.

2. Modelo.

2.1. Clases persistentes.

En este apartado se muestran todas las clases persistentes y la relación entre ellas.



2.2. Interfaces de los servicios ofrecidos por el modelo.

Para agrupar los distintos casos de uso en los cinco Servicios ofrecidos decidimos seguir un criterio basado en la relación entre ellos, es decir, agrupamos los casos de uso según con lo que estuviesen relacionados. Un ejemplo sería, todos los casos de uso relacionados con los usuarios, estarán en el servicio de UserProfile, mientras que los casos de uso relacionados con los comentarios estarán en el servicio de Comment.

El servicio de UserProfile, engloba a las casos de uso de “Registro de usuarios”, “Autenticación y salidas” y “Seguimiento de usuarios”. Concretamente se desarrollan las funcionalidades de cambiar contraseña, buscar un usuario, loguearse, registrarse, actualizar el perfil, comprobar si existe un usuario y seguir un usuario.

IUserService
+ ChangePassword(userProfileId: long, oldClearPassword: String, newClearPassword: String): void + FindUserProfileDetails(userProfileId: long): UserProfileDetails + Login(loginName: String, password:String, passwordIsEncrypted:Boolean): LoginResult + RegisterUser(loginName: String, clearPassword: String, userProfileDetails: UserProfileDetails): long + UpdateUserProfileDetails(userProfileId: long, userProfileDetails: UserProfileDetails): void + UserExists(loginName: string): bool + follow(followedLogin: string, followerLogin: string): void

El servicio de Comment, engloba a la funcionalidad “Añadir comentario”, en la que podemos distinguir el añadido, actualización y borrado de un comentario, como se especifica en el caso de uso.

ICommentService
+ AddComment(pubId: long, comment: String, userId: long): long + UpdateComment(commentId: long, content: String): void + RemoveComment(commentId: long): void + ShowComments(pubId: long, startIndex: int, count: int): List<Comment>

El servicio de ImageUpload, engloba a parte de la funcionalidad “Subida de imágenes”.

IImageUploadService
+ UploadImage(img: ImageUploadDetails, tags: List<string>, category: string): long + SearchByKeywords(keywords: String, startIndex: int, count: int): List<ImageUpload>

El servicio de Tag, engloba al caso de uso adicional de “Etiquetado de imágenes”. En él creamos las etiquetas y las recuperamos.

ITagService
+ CreateTag(name: String): long + GetAllTags(): List<Tag>

Y por último el servicio de Publication, este podemos decir que es uno de los servicios más importante, ya que es donde se gestionan la mayoría de las funcionalidades. En él se implementan los casos de uso relacionados con “Búsqueda de imágenes” , “Indicar “Me gusta” y parte de “Subida de imágenes”. Concretamente, se desarrollan los casos de uso de subir, modificar, eliminar, buscar y dar “me gusta” a una publicación.

IPublicationService
+ UploadPublication(userId: long, imgId: long): long + UpdatePublication(pubId: long, publicationDetails: PublicationDetails): void + RemovePublication(pubId: long): void + FindPublication(keyword: String, category: String): PublicationDetails + LikedPublication(pubId: long, userId: long): void

2.3. Diseño de un DAO.

En este apartado se incluye el diagrama de clases del CommentDao de la aplicación.

CommentDao
+FindByPubIdOrderByDateAsc(pubId: int, startIndex: int, count:int): List<Comment>

2.4. Diseño de un servicio del modelo.

En este apartado se incluye el diagrama de clases del CommentService de la aplicación.

CommentService
+ ShowComments(pubId: long, startIndex: int, count: int): List<Comment> + AddComment(pubId: long, comment: string, userId: long): void + UpdateComment(commentId: long, content: string): void + RemoveComment(commentId: long): void

También se incluye el diagrama de secuencia que muestra la ejecución de AddComment en la capa modelo.

