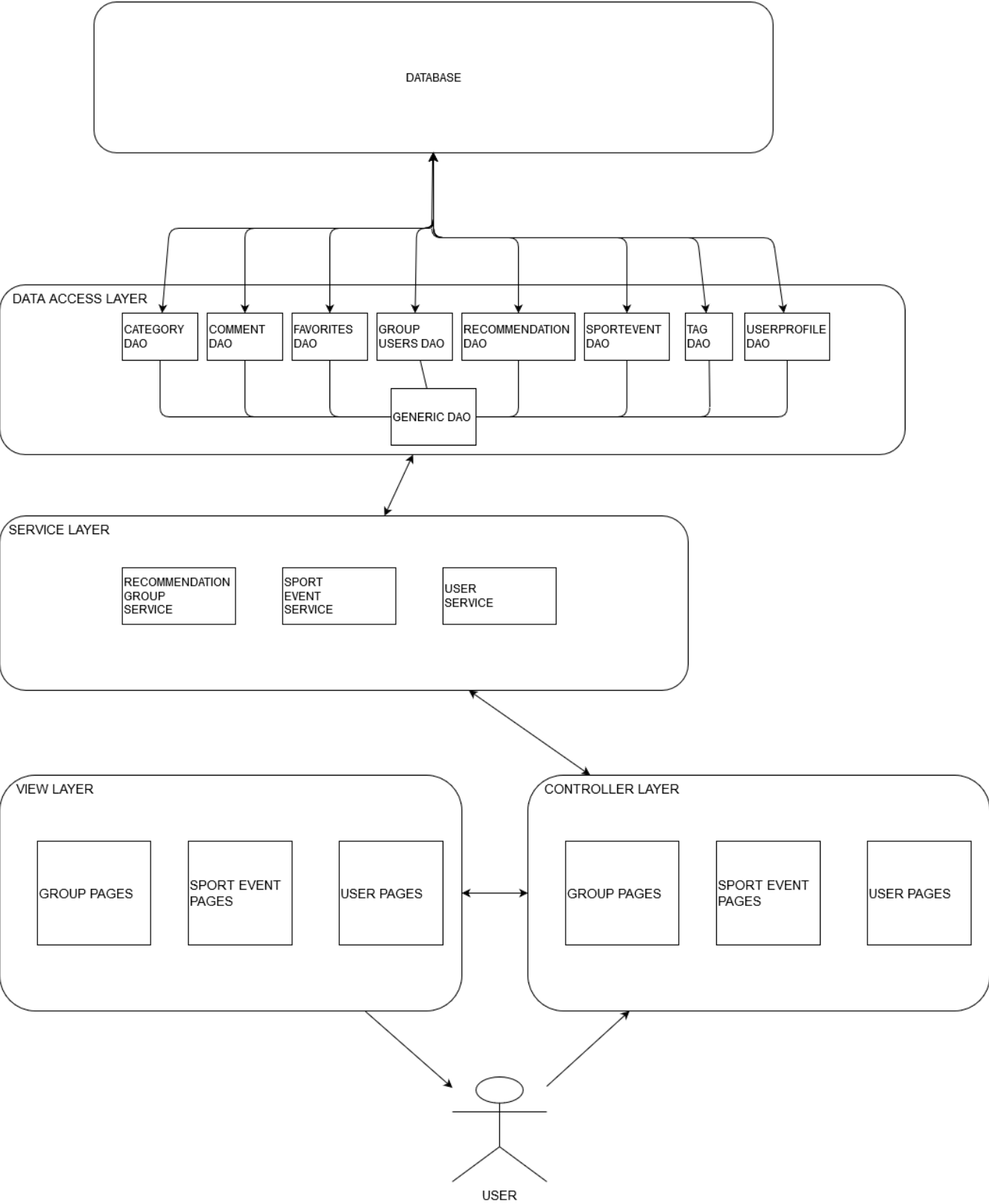


Sitio Web de comentarios sobre eventos deportivos

1.Arquitectura global:



Database: Conjunto de datos y relaciones lógicas entre ellos suponen el almacenamiento de la información sobre usuarios, eventos, grupos y demás información que se tiene que guardar de la aplicación. Esta formada por entidades, que almacenan esta información y se relacionan entre ellas.

Data access layer: Capa de acceso a datos. Aquí tenemos los DAOs, estos se encargan de obtener la información de la base de datos realizando consultas, contienen operaciones esenciales (eliminar, actualizar, crear, encontrar) . También implementan consultas mas complejas o personalizadas para la entidad de la que se desea extraer la información.

Los DAOs estan compuestos por una interfaz y una Clase, la interfaz implementa los métodos que opera, la implementación desarrolla la forma en que se realizan estos métodos.

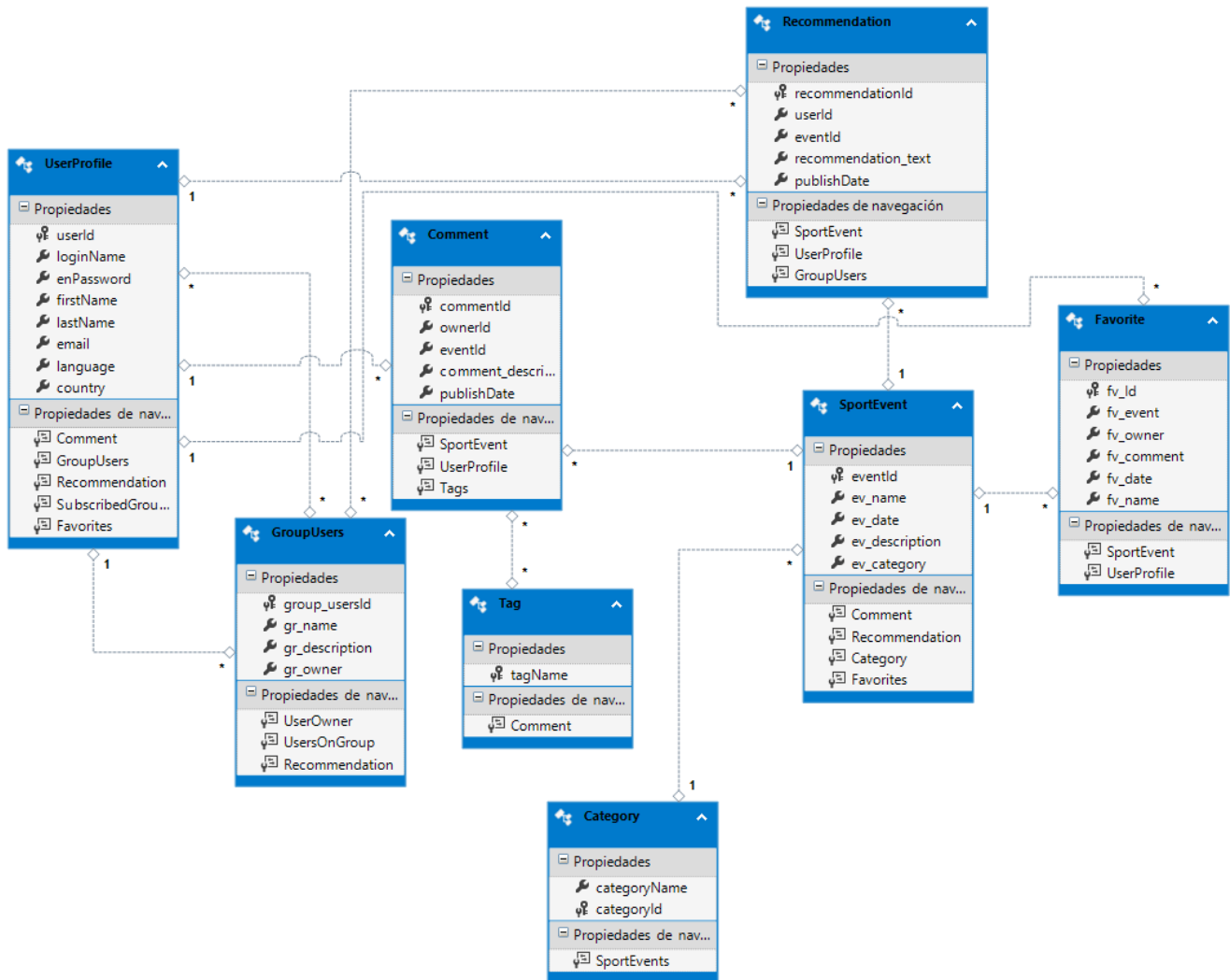
Service layer: Esta capa contiene los paquetes con las funcionalidades de la aplicacion. Estos paquetes son RecommendationGroup, SportEvent y user. Se componen de una interfaz que expone los métodos que permiten ejecutar estas funcionalidades, y de una clase, que implementa dicha funcionalidad. También contienen excepciones propias que lanzan los métodos en caso de que se produzca algún error, como puede ser logearse con un loginName de usuario que no está registrado, para enviar la información se utilizan DTOs para agrupar la información y exponer menos los tipos con los que se implementan estas funcionalidades, asi como por ejemplo las Entidades.

Controller Layer:

View layer:

2. Modelo

2.1 Clases persistentes:



2.2 Interfaces de los servicios ofrecidos por el modelo:

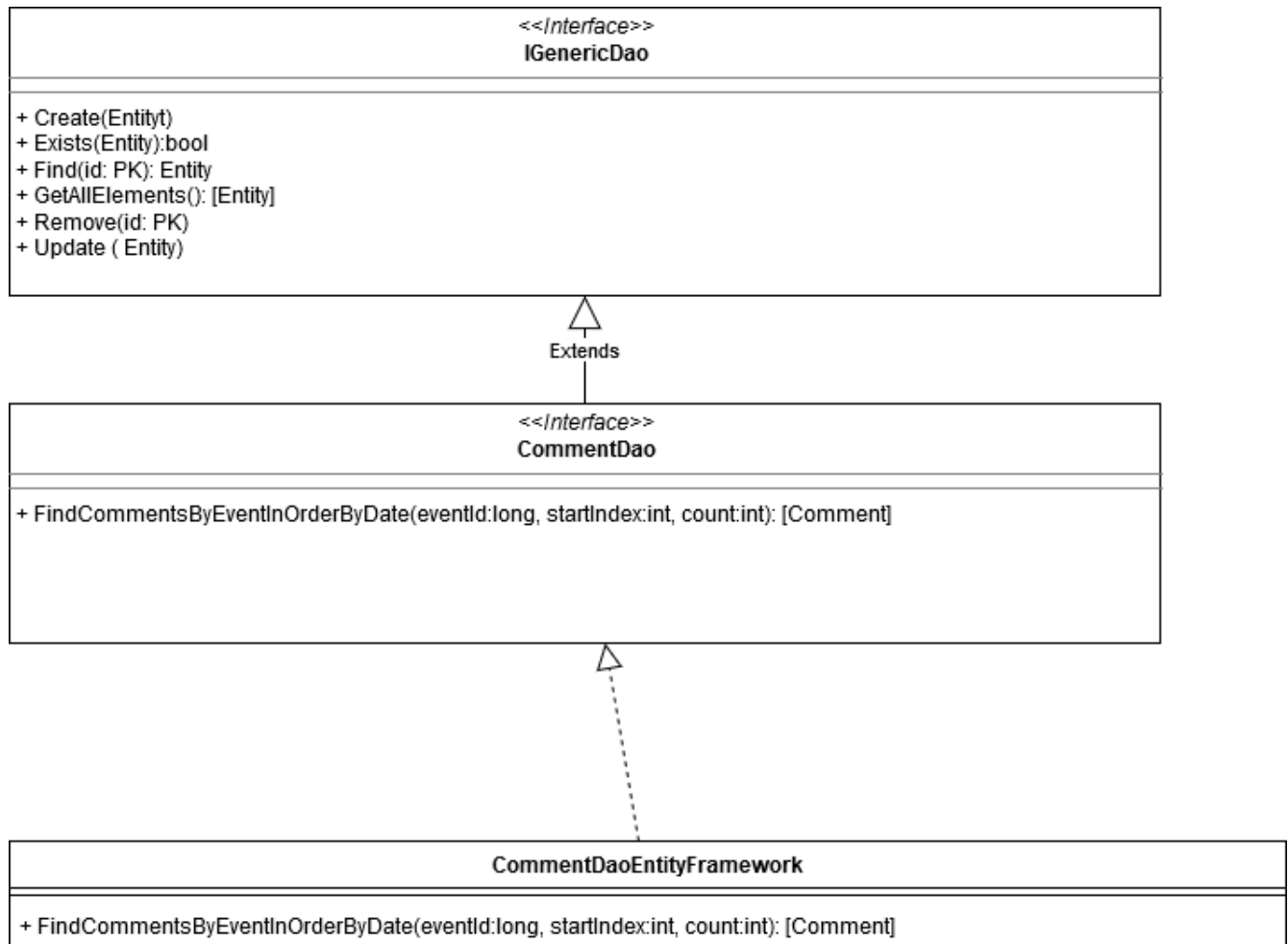
La decisión de agrupamiento de casos de uso en servicios se ha tomado teniendo en cuenta el tipo de caso de uso según con que entidad de mayor importancia que nosotros consideramos estuviese relacionado. Así, tenemos RecomendacionesGrupos, EventosDeportivos y Usuarios

<<Interface>> UserService
+ UserProfileDao: IUserProfileDao + field2: Type
+ RegisterUsers(loginName:string, password:string): long + Login(loginName:string, password:string, isEncrypted:bool): LoginResult + FindUserProfileDetails(userProfileId:long): UserProfileDetails + UpdateProfileDetails(userProfileId:long, userProfileDetails:UserProfileDetails) + ChangePassword(userProfileId:long, oldPassword:string, newPassword:string)

<<Interface>> UserService
+ UserProfileDao: IUserProfileDao + SportEventDao: ISportEventDao + CommentDao: ICommentDao + CategoryDao: ICategoryDao + TagDao: ITagDao + FavoritesDao: IFavoritesDao
+ GetCategories() + FindComments(eventId:long, startIndex:int, count:int):[DTOComment] + FindEvents(keyword: string, startIndex: int, count: int, category: string): SportEventBlock + AddComment(userLogin: string, sportEventId: long, commentDescription: long):long + UpdateTagComment(userLogin: string, commentId: long, tags: [String]) + AddToFavorites(userLogin: string, name:string, comment:string) + DeleteFromFavorites(userLogin: string, sportEventId: long) + ListFavorites(userLogin:string):[DTOFavorite] + UpdateComment(commentId:long, comment_text:string) + getComment(commentId:long):DTOComment

<<Interface>> RecommendationGroupService
+ UserProfileDao: IUserProfileDao + SportEventDao: ISportEventDao + GroupUserDao: IGroupUserDao + RecommendationDao: IRecommendationDao
+ AddUserToGroup(loginName:string, groupId:long) + CreateGroup(groupName:string, description:string, loginName:string): long + AbandonGroup(loginName:string, groupId:long) + DeleteGroup(loginName:string, groupId:long) + ShowAllGroups(string loginName):[DTOGroupUser] + AddRecommendation(loginName:string, eventId:long, recommendation:string):long

2.3 Diseño de un DAO:



2.4 Diseño de un servicio del Modelo:

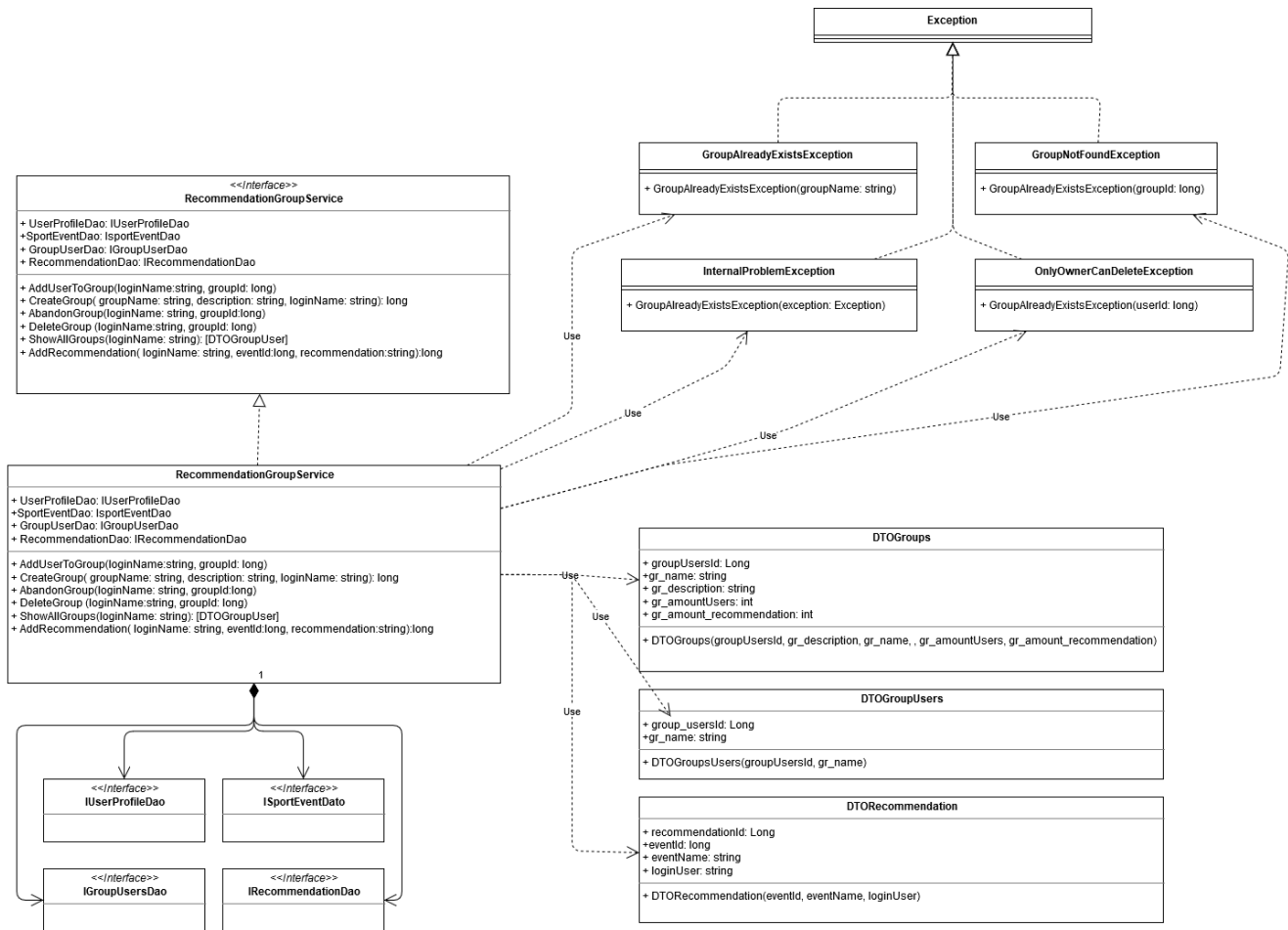
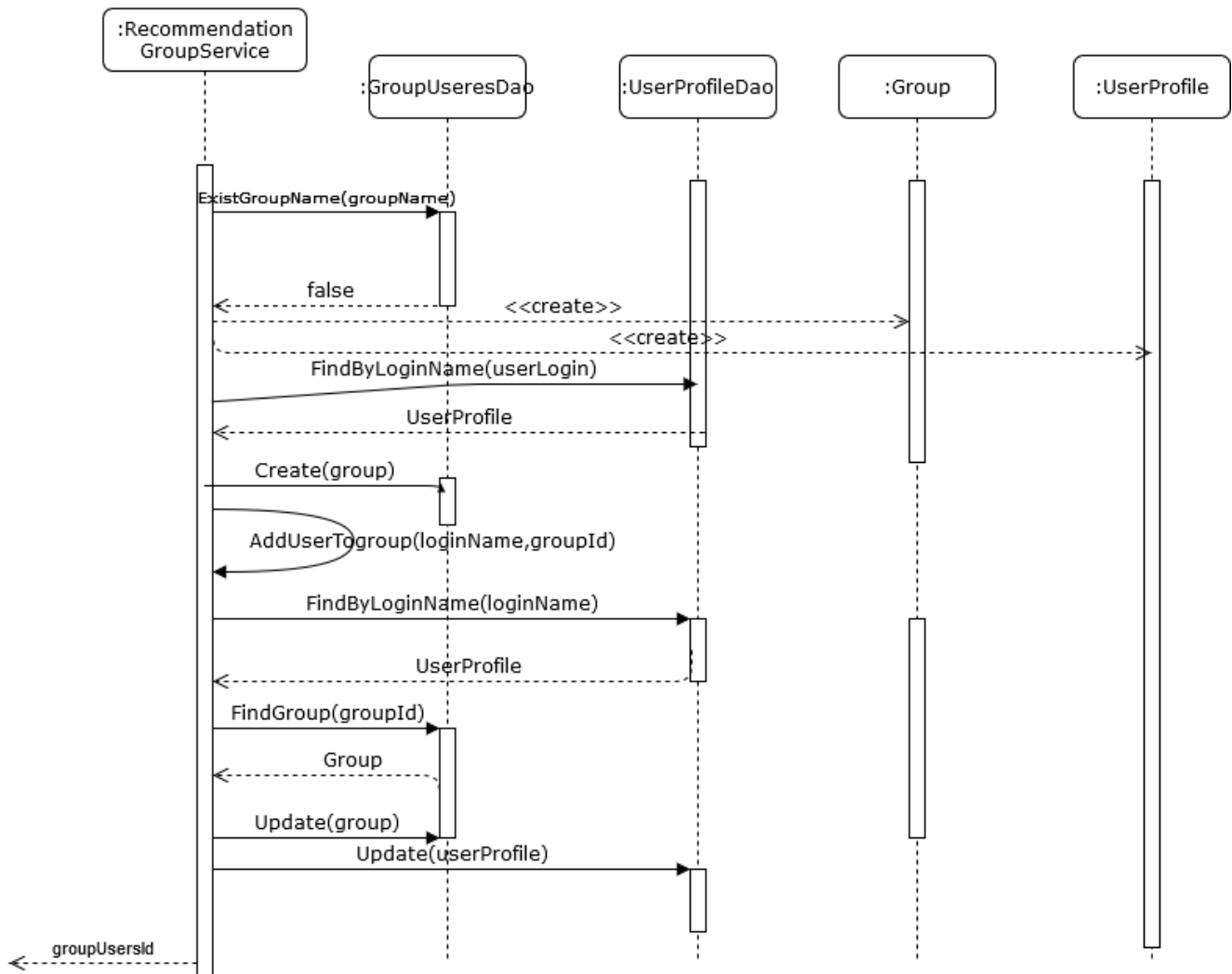


Diagrama secuencia:

CREAR GRUPO



3. Interfaz Grafica

4. Parte adicional

5. Compilacion e instalaci3n de la aplicaci3n

Compilaci3n y ejecuci3n mediante visual studio 2019 ,ejecucion de scripts de creaci3n de bases de datos sobre SQL Server .

Establecemos la soluci3n web como proyecto de inicio y ejecuci3n con Ilexpress

