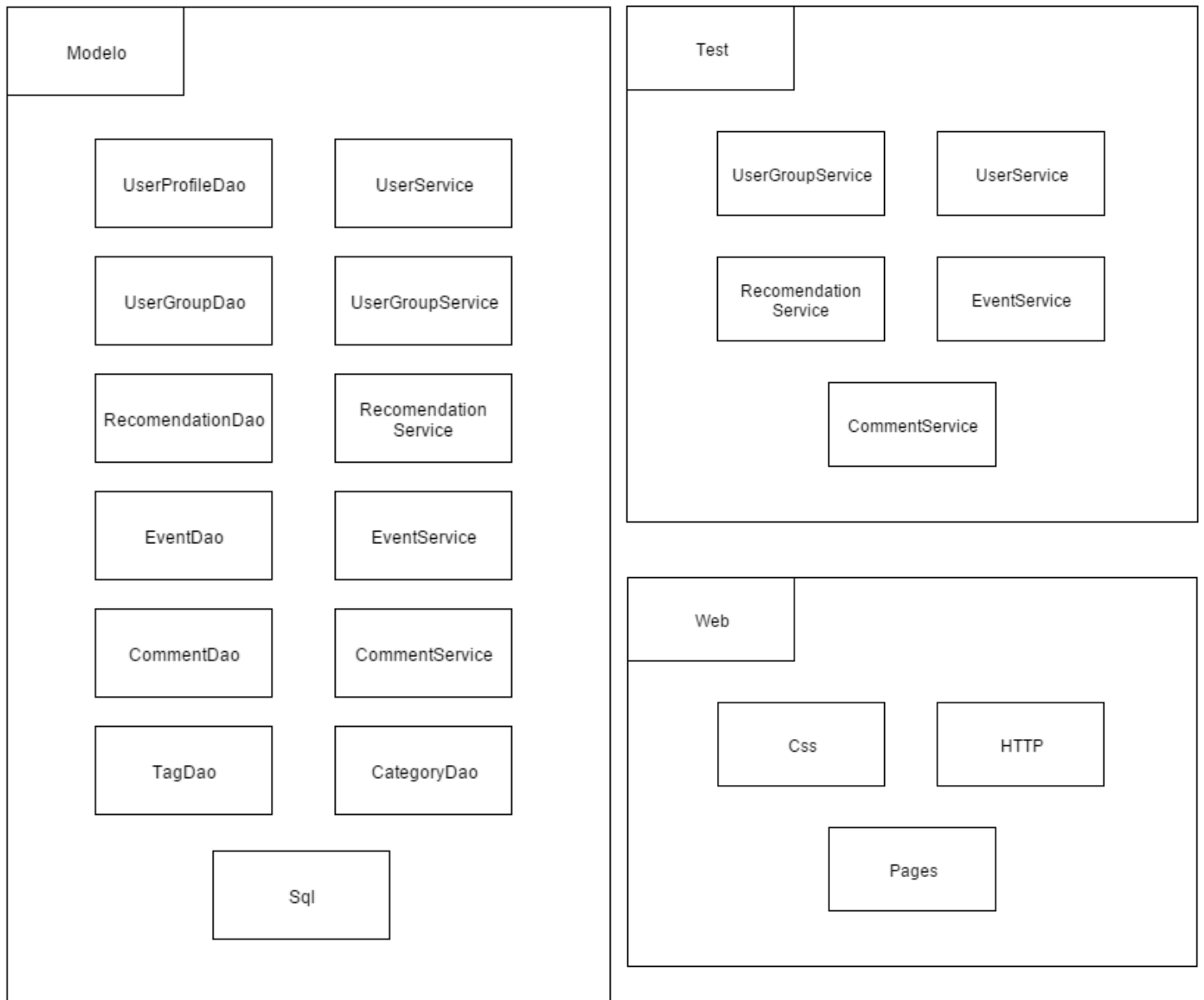


Práctica MaD

MEMORIA

IAGO SANTOS DOMÍNGUEZ | JOSÉ ANDY QUINTERO MELO

1. Arquitectura Global

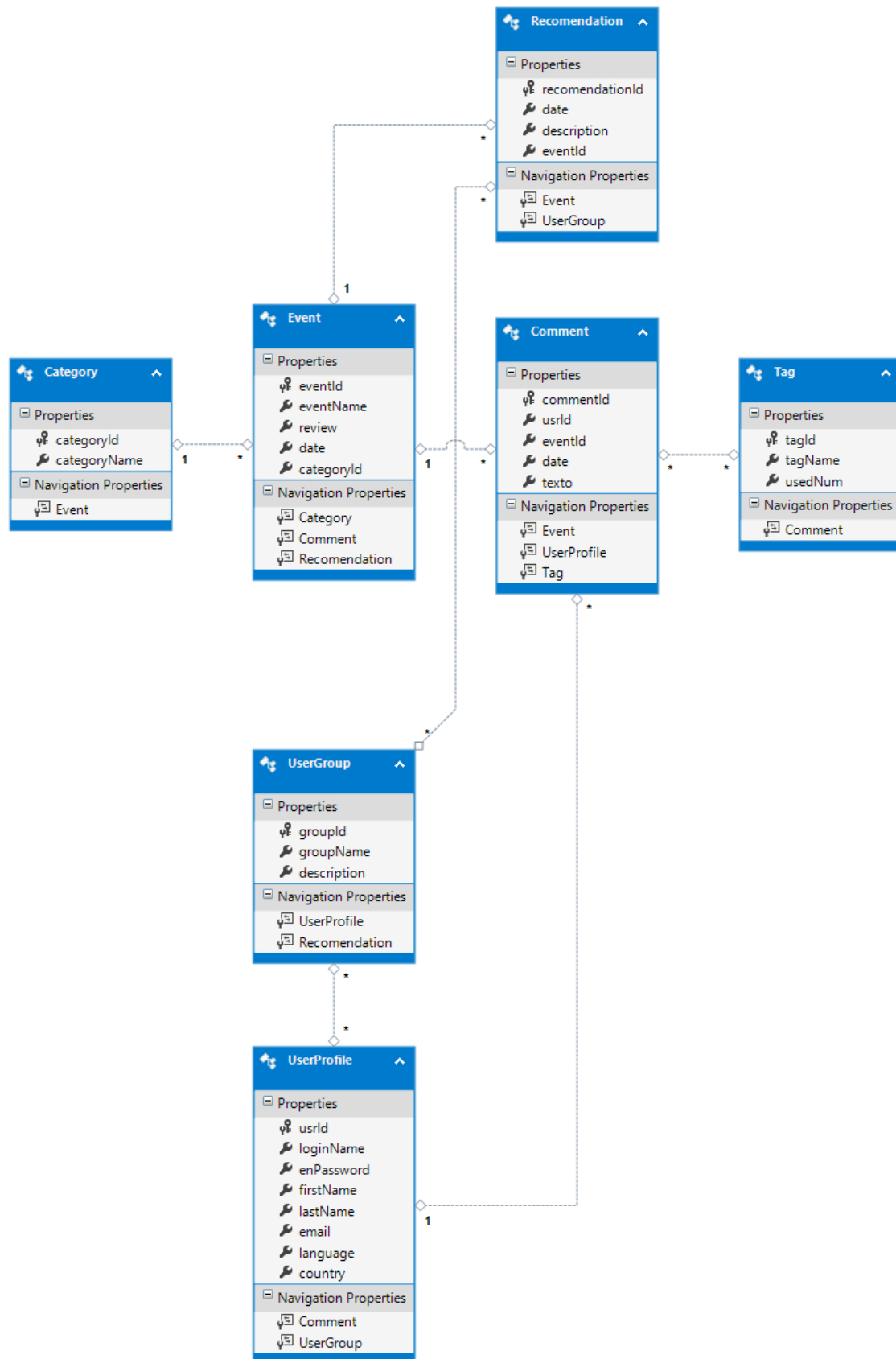


La aplicación se organiza en tres grandes bloques (Modelo, Test y Web) que contienen toda su lógica.

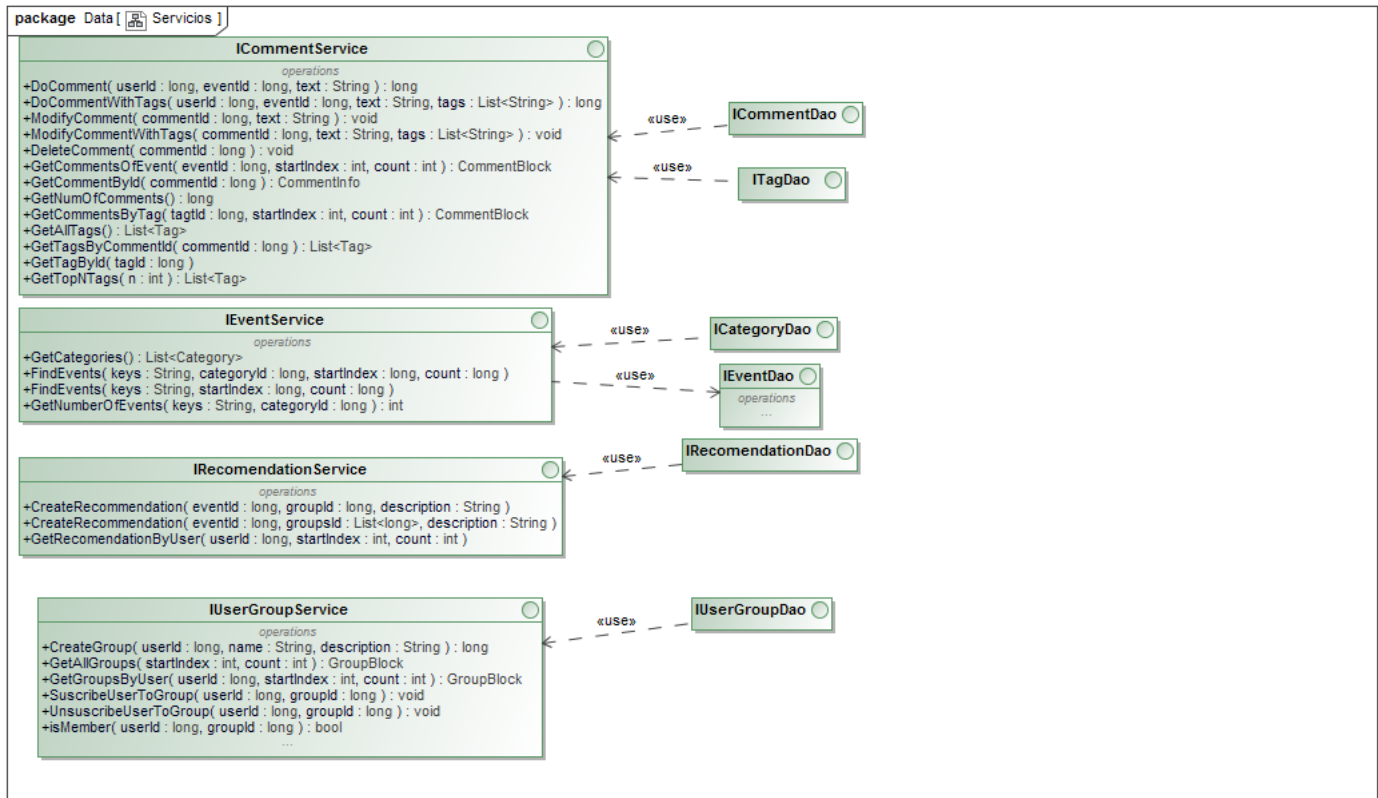
- Modelo: Contiene la lógica de negocio principal *Service y las funcionalidades de acceso a datos *Dao además del paquete Sql que tiene todo lo necesario para crear las bases de datos y las tablas de entidades.
- Test: Contiene todas las pruebas realizadas para probar el correcto funcionamiento de los métodos de cada servicio.
- Web: Contiene la toda la parte grafica de la aplicación, páginas, transiciones entre ellas y estilos.

2. Modelo

2.1. Clases Persistentes

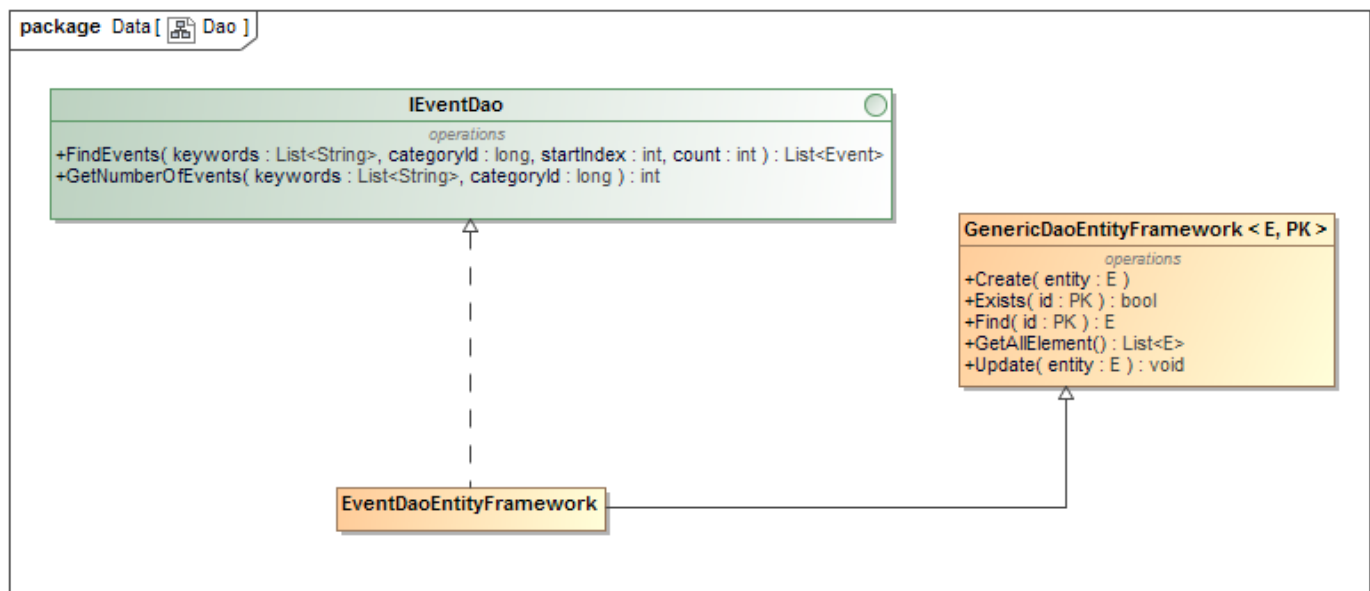


2.2. Interfaces de los servicios ofrecidos por el modelo



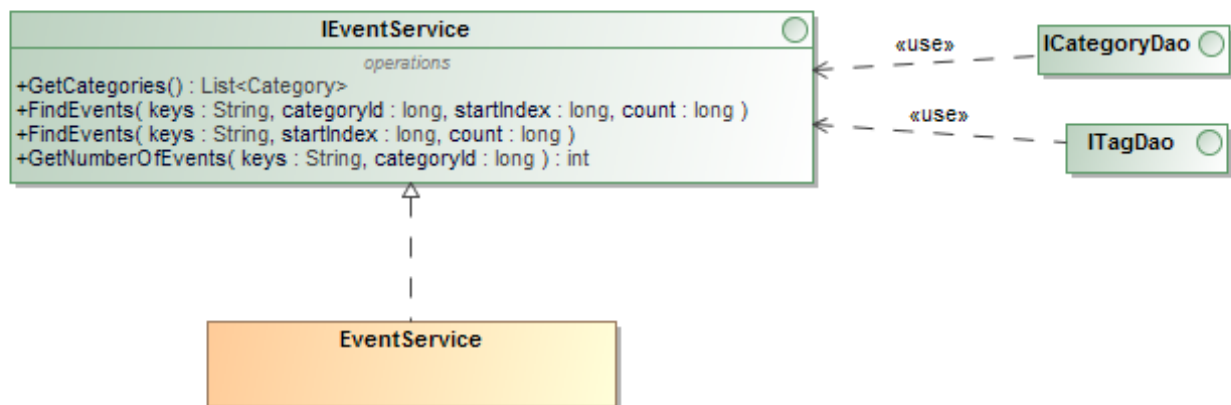
- En un principio decidimos usar un servicio para cada entidad de la aplicación para posteriormente darnos cuenta de que, por ejemplo, categoría (Category) y etiqueta (Tag) no tiene mucho sentido separarlos en un servicio individual, por ese motivo, los métodos referentes a categorías están incluidos en IEventService y los referentes a tags se incluyen en ICommentService.


2.3. Diseño de un DAO

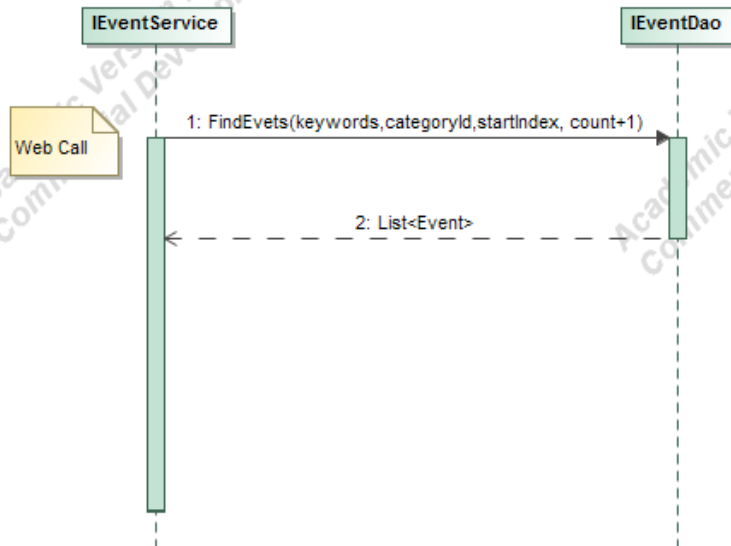


2.4. Diseño de un servicio del modelo

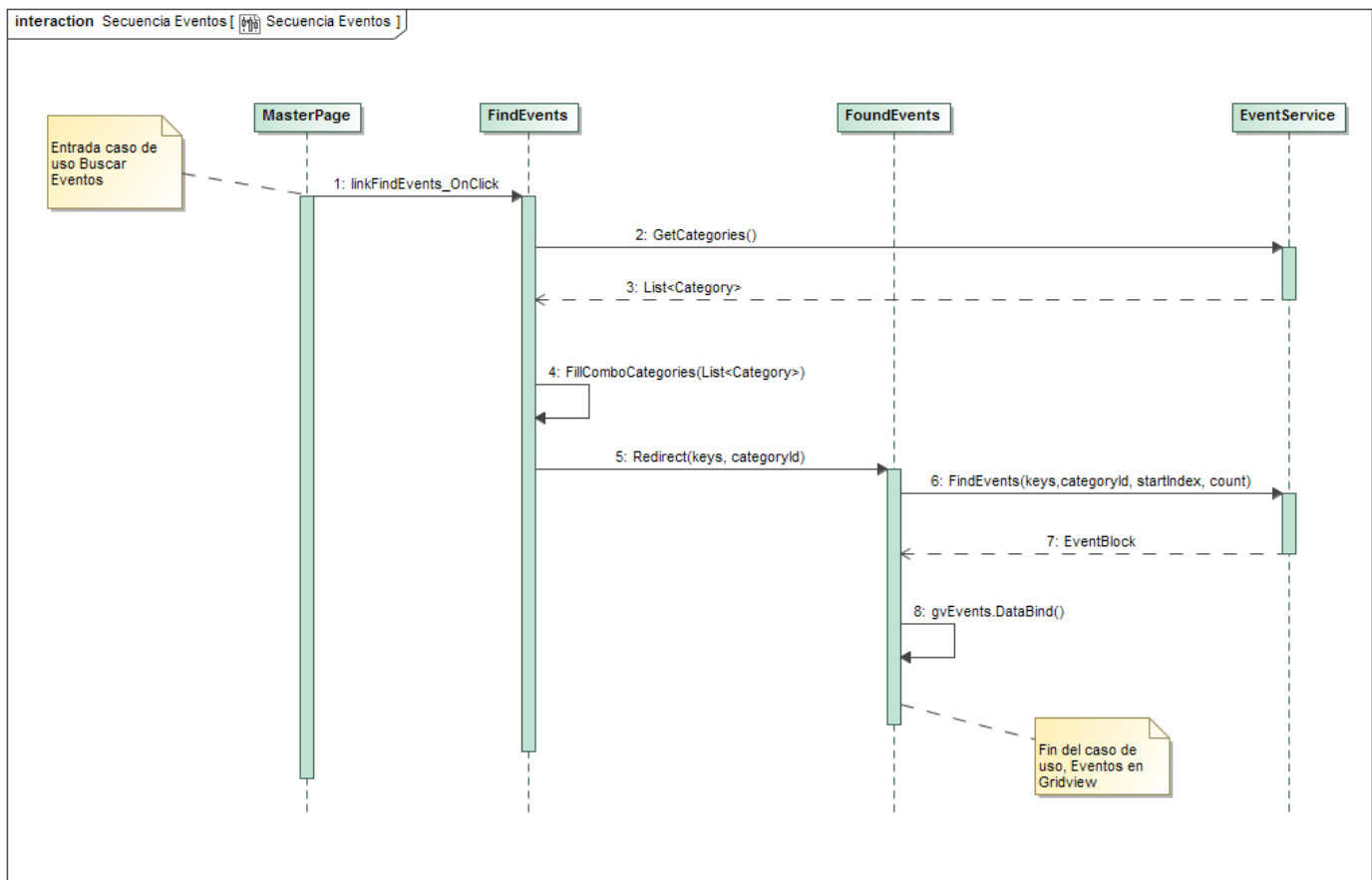
package Data [ Servicio]



interaction Untitled1 [ Untitled1]



3. Interfaz Gráfica



4. Parte Opcional

- En la parte opcional hemos realizado la parte de etiquetado de comentarios para lo cual comenzamos por añadir la entidad Tag a la base de datos con las relaciones pertinentes (se muestran en el apartado 2.1).
- A continuación creamos lo necesario para acceder a ellos, TagDao y, en lugar de crear un servicio propio, añadimos los métodos necesarios en el servicio de comentarios (CommentService).
- En lo referente a la vista, creamos 5 clases Css que cambian el color y el tamaño de las etiquetas usando el atributo CssClass que proporciona ASP.NET para los WebControls y creamos un top 20 de enlaces consecutivos mostrados en todo momento en la plantilla de todas las páginas, MasterPage.

5. Compilación e Instalación

- Ejecutar los scripts de creación de la base de datos (SqlServerCreateDatabase) y el de creación de tablas (SqlServerCreateTables) contenidos en la carpeta Sql del Modelo.

- En caso de querer ejecutar los tests habrá que ejecutar el script de creación de la base de datos para tests (SqlServerCreateTables.Test) que se encuentra dentro del paquete Tests.
- **Puede cambiarse el valor de @Default_DB_Path en todos los scripts para indicar el lugar de creación de la base de datos.*
- Para ejecutar la interfaz hemos creado un script que inserta datos en la base de datos y se encuentra en la carpeta Sql del modelo (InsertsSQLMaD) este script creara por defecto cuatro categorías y eventos para cada una de ellas.
- Una vez ejecutado todos los scripts podemos proceder a ejecutar los tests: Test > Windows > Test Explorer > Run All
- Una vez ejecutados todos los scripts podemos proceder a ejecutar la interface: Click derecho encima del paquete web > Set as StartUp Project y procedemos a su ejecución.

6. Problemas Conocidos

- Deberíamos usar Server.Transfer(url) en vez de Response.Redirect(url) debido a que la primera opción es más eficiente al cambiar de páginas en el contexto local, pero lo hicimos con redirect desde el principio y no tuvimos tiempo de arreglarlo.
- El hashcode de un par de entidades no está redefinido porque nos fallaban los tests y no entendíamos el error para poder corregirlo.