



UNIVERSIDADE DA CORUÑA

MEMORIA

Venta Online: Tienda de comercio electrónico

Grupo : mad22

Adrián Bouza Naveira

Yeray Méndez Romero

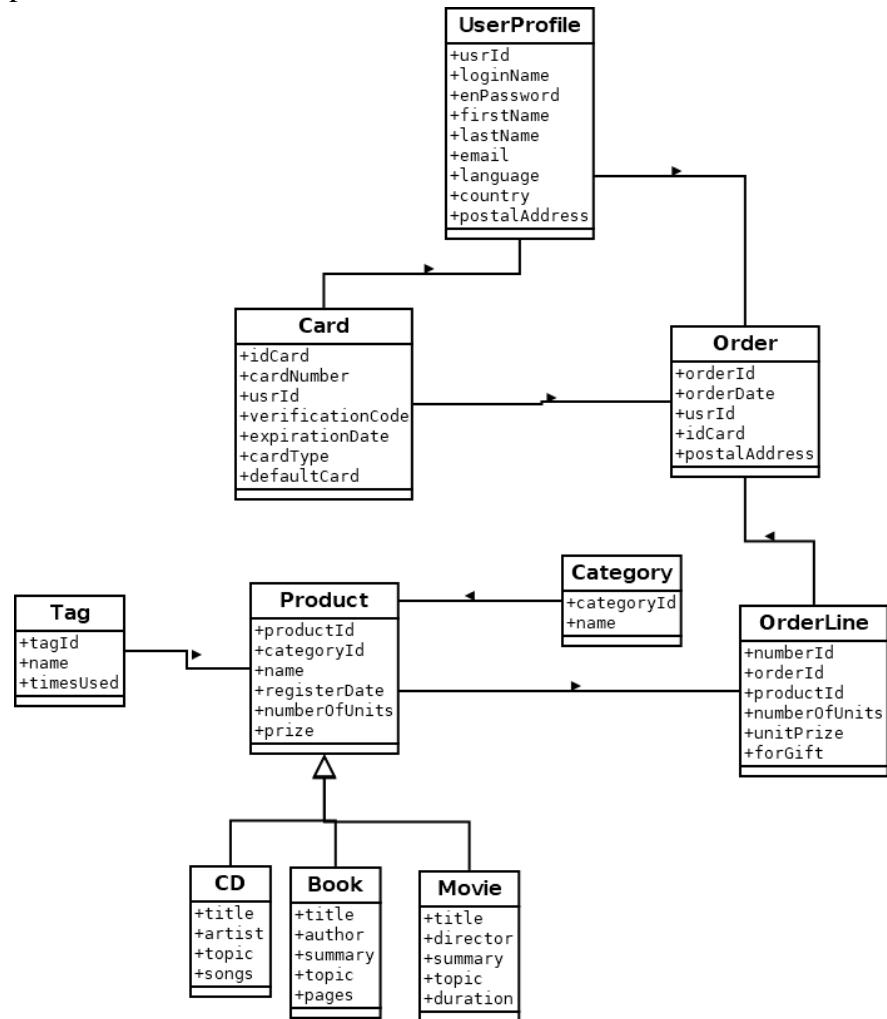
1. Arquitectura global

La aplicación está formada por 3 proyectos:

- a) - Model: Aquí se encuentra la parte modelo de la aplicación, cuenta con el Entity Data Model de Entity framework para acceso a datos además de los DAOs para los diferentes tipos de información que se manejan (tarjetas, categoría, pedido, línea de pedido, producto, tag y perfil de usuario). También cuento con los DTOs para comunicarse con la vista. Los DTOs son los ficheros que terminan en details p.ej: CardDetails.cs, ProductDetails.cs... Por último están los ficheros del servicio. Hay un servicio diferente para cada tipo de datos mencionados anteriormente, excepto para pedido y línea de pedido que están relacionadas
- b) - Test: Este proyecto cuenta con tests para cada servicio mencionado anteriormente, excepto para pedido y línea de pedido que comparten test.
- c) - WebApplication: Está formada por una carpeta con ficheros de estilos css, una carpeta HTTP que contiene las utilidades necesarias para la comunicación con el modelo, gestión del idioma, carrito ... Una carpeta Pages que contiene todas las páginas de la aplicación divididas por tipos.

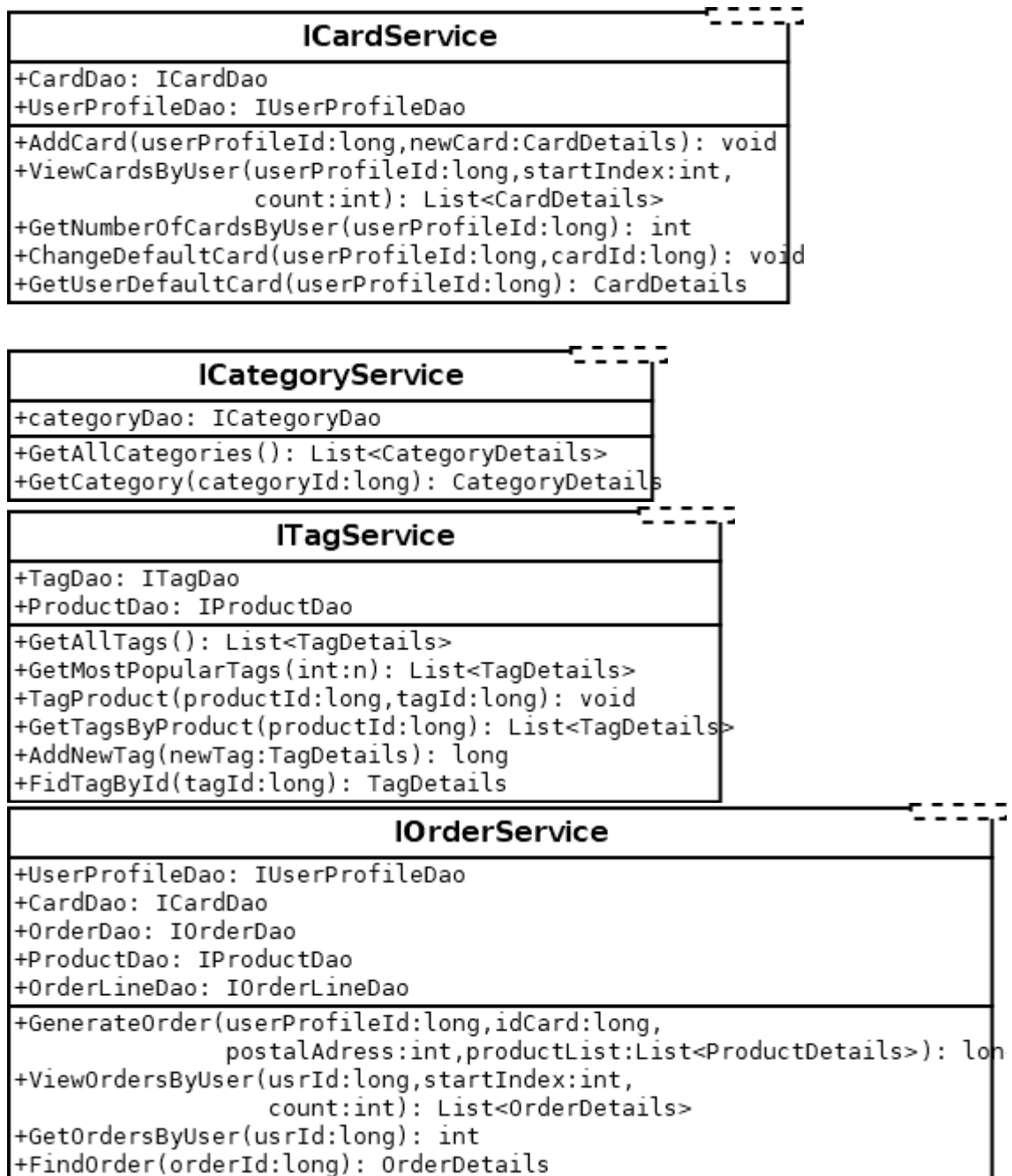
2. Modelo

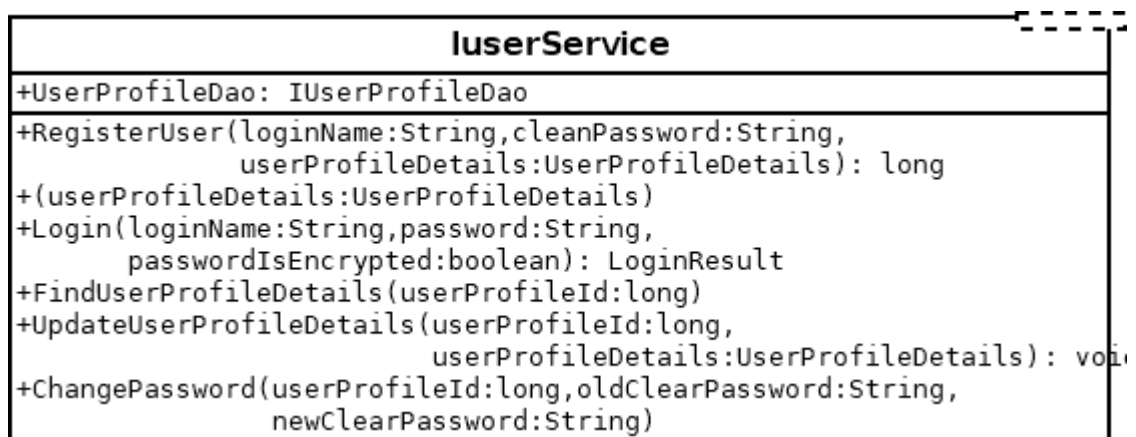
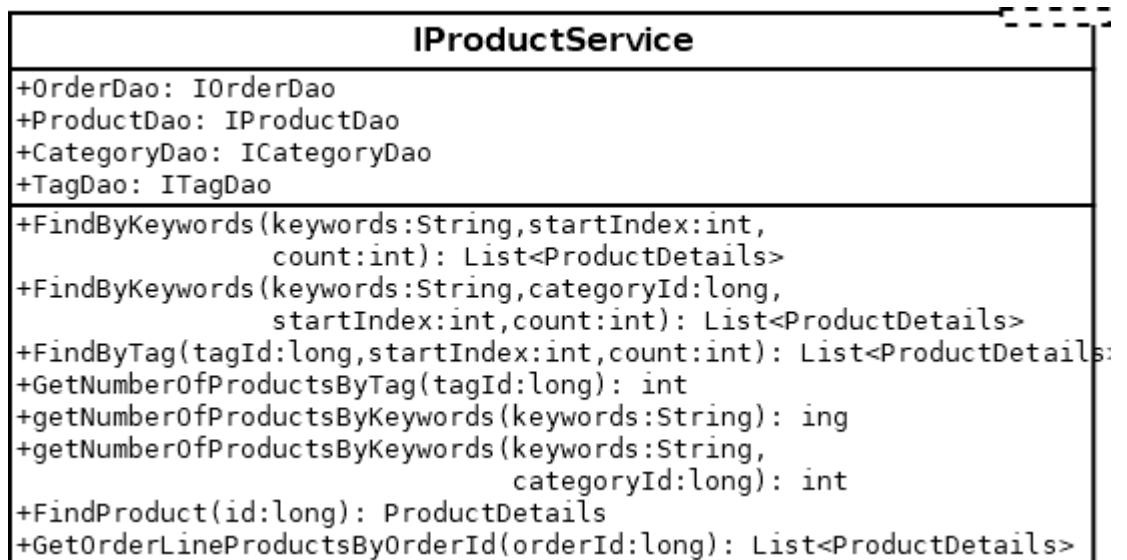
a. Clases persistentes



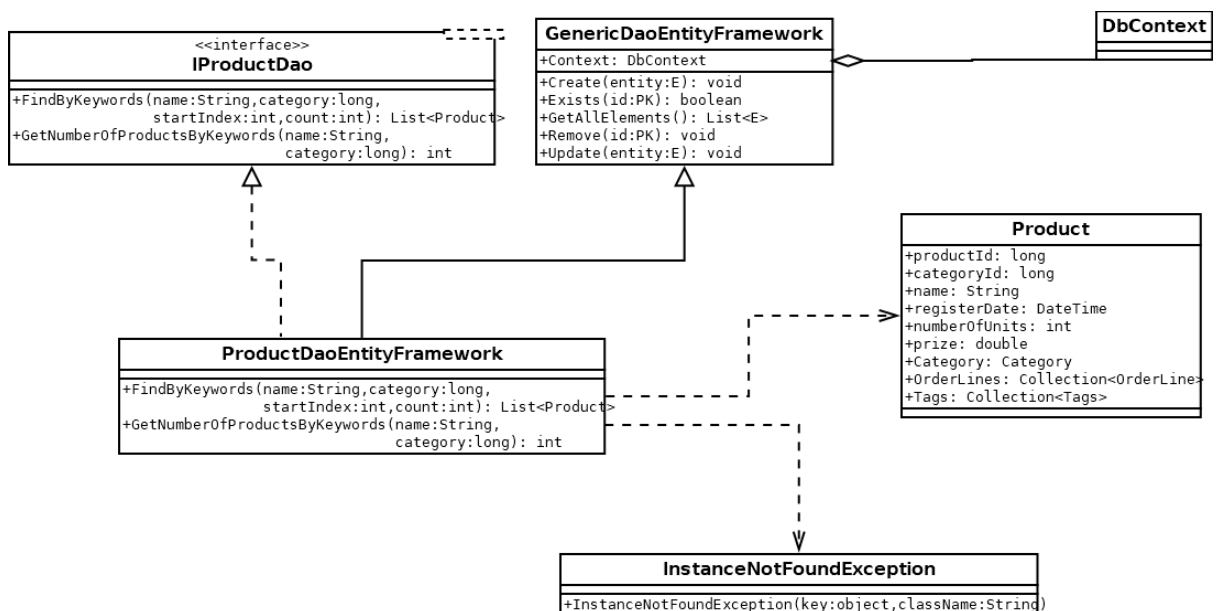
b. Interfaces de los servicios ofrecidos

El criterio que se ha seguido ha sido crear un servicio para cada clase persistente excepto para pedido y línea de pedido que comparten servicio. Hemos considerado que cada tipo se diferencia lo suficiente del resto como para tener su propio servicio proporcionando así una mayor modularidad.

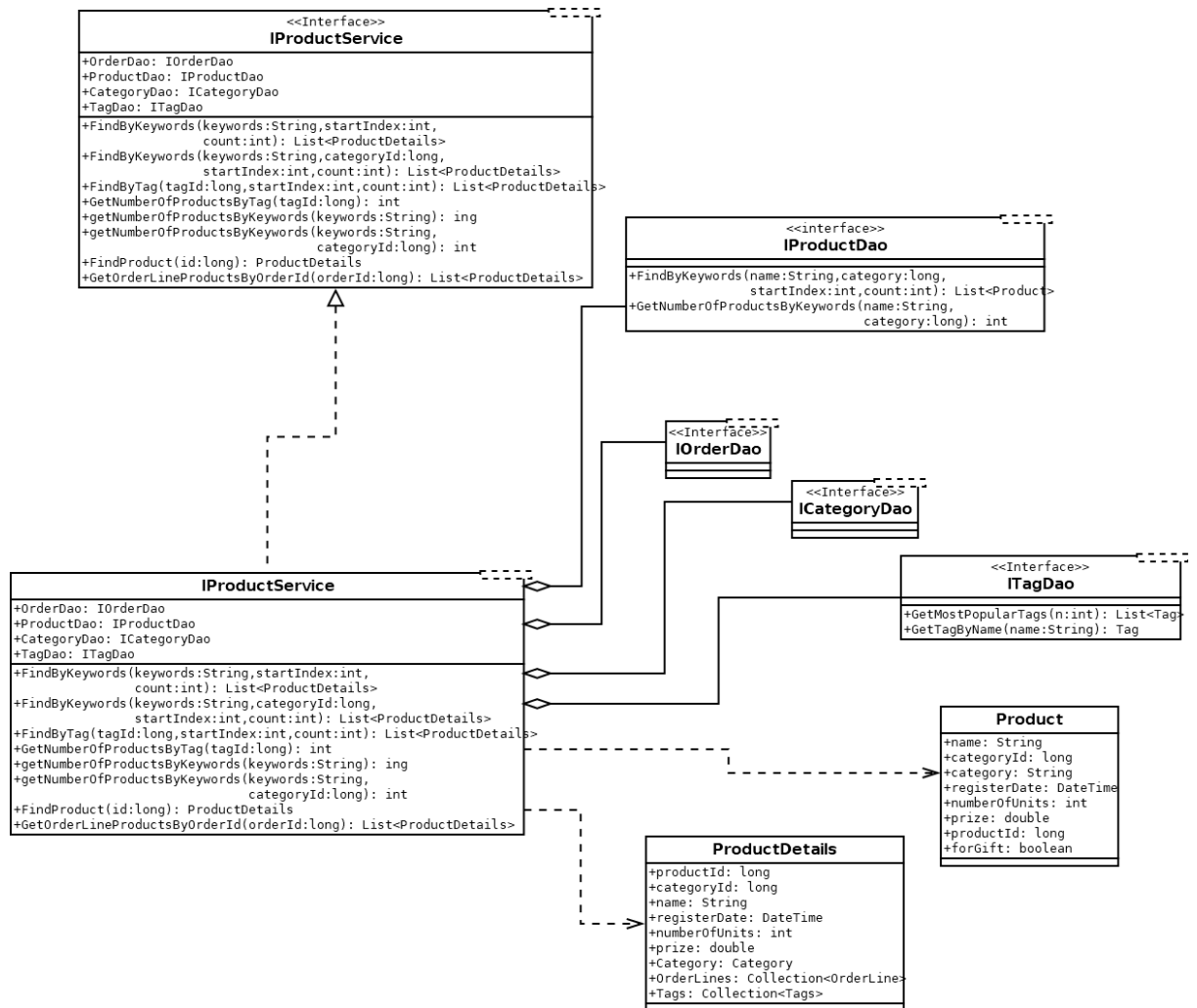


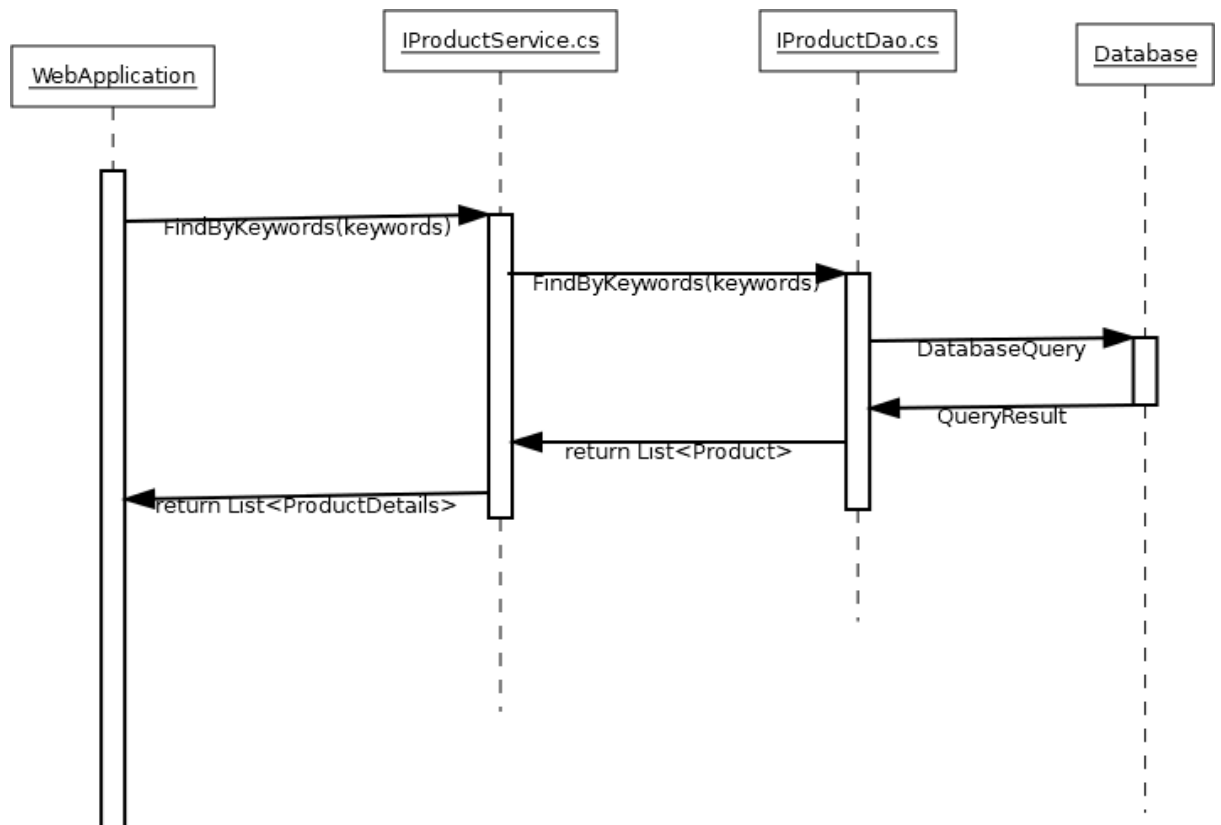


c. Diseño de un DAO



d. Diseño de un servicio del modelo

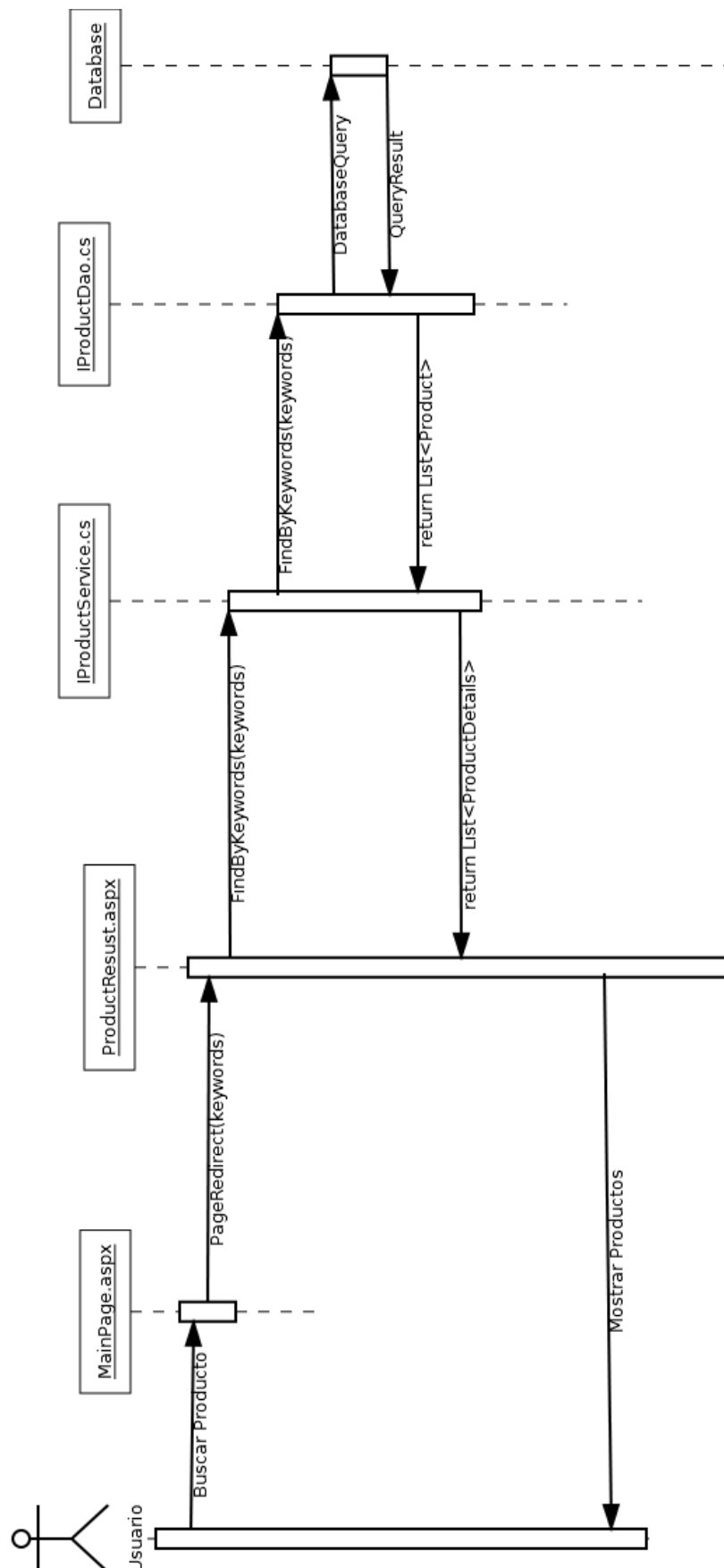




e. Otros aspectos

3. Interfaz gráfica

Diagrama de secuencia para el caso de uso de buscar productos:



Search products

Category

Books

Search

Copyright ©

Name	Category	Prize	
El nombre del viento	Books	23,5	Add to Cart

Copyright ©

4. Apartados opcionales

a. Cacheado de búsquedas

Para realizar esta funcionalidad se ha empleado el espacio de nombres `System.Runtime.Caching` y se ha desarrollado la clase `CacheApplication`, la cual posee una serie de métodos (almacenar, devolver un objeto de la caché...) para controlar el almacenamiento y gestión de los objetos cacheados durante las búsquedas. El cacheado tiene lugar siempre que una búsqueda devuelva uno o más resultados, en caso contrario no se realiza el cacheado. En el caso de obtener resultados tras la consulta, se emplean los caracteres de la misma y el identificador de categoría como clave para almacenar los resultados en caché. Éstos se almacenan la primera vez que se realiza la consulta, y serán utilizados siempre que se realice una petición cuya clave está registrada en la caché. A su vez hemos considerado necesario eliminar de la caché aquellos objetos cuya información pueda ser modificada debido a una compra de un usuario, lo cual se produce cuando este realiza la compra de un producto que actualmente está almacenado en caché. Si no se realizara el paso anterior, la información de la aplicación sería errónea y podría permitir, por ejemplo, que un usuario añadiese un producto al carrito cuando no hay stock o se ha eliminado. Por petición del enunciado, tampoco se realiza una revisión de si las características del producto han cambiado tras su almacenamiento en caché.

b. Etiquetado de productos

Para el etiquetado de productos de la aplicación, hemos colocado en la página de detalle de un producto una serie de desplegables que permitirán a los usuarios autenticados en la aplicación, etiquetar y desetiquetar el producto. También se ha incluido una entrada de texto junto con un botón para crear una nueva etiqueta para la aplicación y etiquetar el producto. Cada una de las etiquetas creadas de la forma anterior estarán disponibles para cualquier producto. No se le permite al usuario crear una etiqueta ya existente, pero sí puede seleccionar una etiqueta del desplegable ya asignada al producto. Esto hará que aumente el número de veces que se ha utilizado la etiqueta seleccionada, lo cual puede proporcionar información de la relevancia de la etiqueta en la aplicación. La página que muestre la información de un producto contendrá además las etiquetas asociadas a él, las cuales tendrán un tamaño relacionado con su empleo en la aplicación (tamaño mayor cuanto más se utilice una etiqueta). En el caso de clicar sobre una de las etiquetas, la aplicación redireccionará al usuario a otra página, en la cual se mostrarán los productos asignados a la misma. También se incluye un botón que permite a los usuarios autenticados eliminar la etiqueta de la web. Por último, todas las etiquetas creadas conformarán una nube de etiquetas que será visible en todas las páginas de la web, y en la que se mostrarán todas las etiquetas de la aplicación.

Amazon Market

Hola yeray - Mi cuenta - Cerrar sesión

Carrito(0)

Nombre: The Dark Knight

Categoría: Movies

Título: The Dark Knight

Director: Christopher Nolan

Resumen: Batman se enfrenta al Joker.

Tema: Superheroes, Accion

Duración: 150

Desetiquetar etiqueta

Select a tag

Seleccionar una Etiqueta

Select a tag

Crea tu propia etiqueta

Crear etiqueta

Cantidad: 1

batman

Agregar al carrito

Nube de etiquetas

80s Queen PopMichael m rock batman

Copyright ©

5. Compilación e instalación de la aplicación

a. Compilación

Para la compilación bastaría con realizar compilación de la solución (Solución 'PracticaMaD') en *Visual Studio*.

b. Instalación

Crear la base de datos con *SQL Server Management Studio* utilizando los scripts contenidos en la carpeta Sql del proyecto Model.

En el caso de que se deseen ejecutar las pruebas de la aplicación, sería necesario crear la base de datos de test de la misma forma que mencionamos anteriormente y seleccionando los scripts sql de la carpeta ScriptsSql del proyecto Test.

Dentro del proyecto WebApplication, acceder a la carpeta Pages y abrir la MainPage.aspx en el navegador.

6. Problemas conocidos

Hemos detectado los siguientes problemas:

1. Una vez el usuario se autentica y selecciona “recordar contraseña” mediante el empleo de cookies, observamos que en determinados momentos, la aplicación web recompila el proyecto y, aunque en el navegador el usuario aparece como autenticado, acciones como las de tramitar pedido lo obligan a volver a autenticarse.
2. En la página de tramitar pedido, a veces la aplicación devuelve error al intentar cargar la tarjeta por defecto del usuario, aunque en el código se comprueba que este no sea null.