UNIVERSIDAD DE BURGOS

Diseño y mantenimiento de software

Memoria del proyecto.

Marcos Orive Izarra

Índice

Tabla de contenido

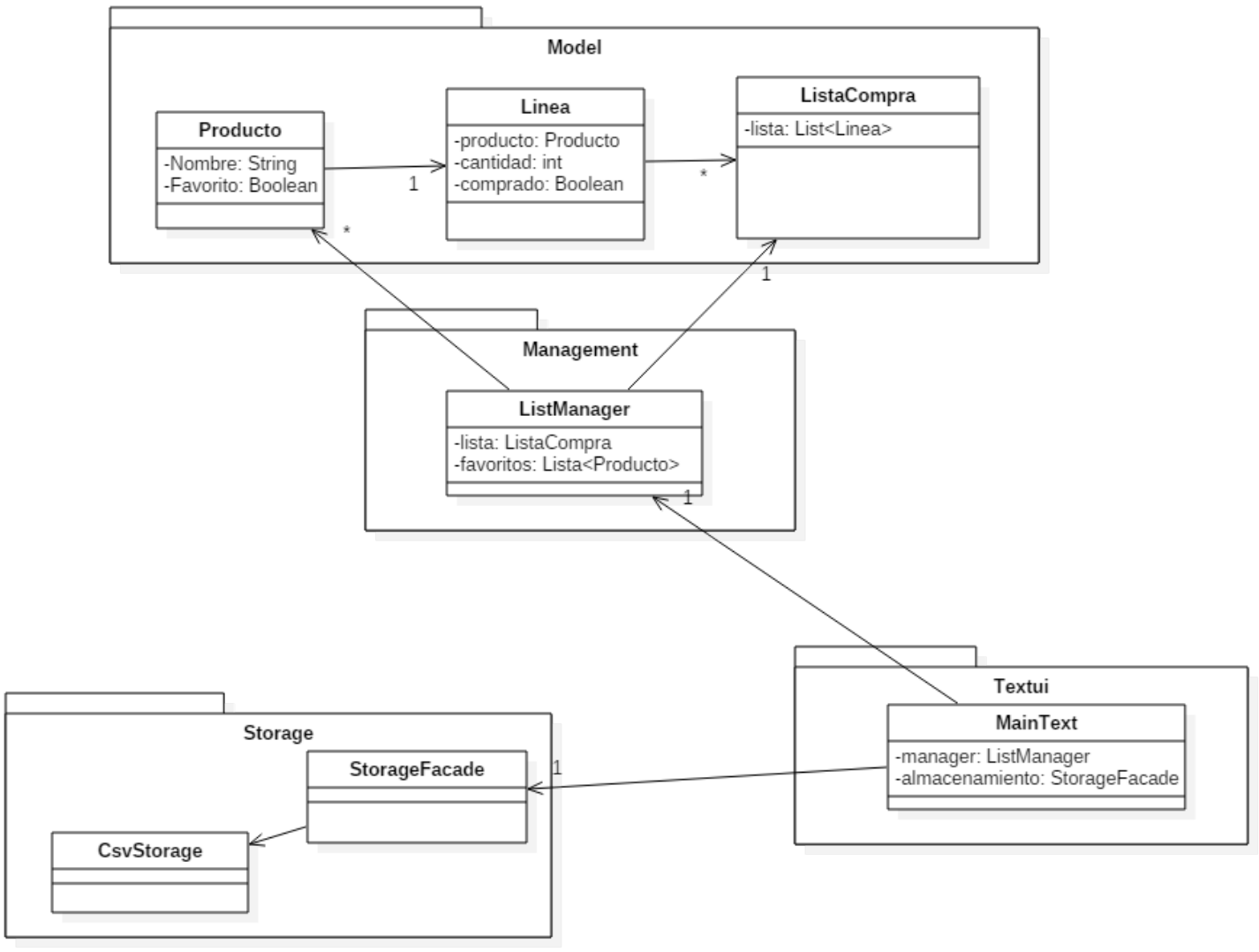
1. Introducción

Como parte de la asignatura de Diseño y mantenimiento de software hemos tenido que diseñar una aplicación para gestionar la lista de la compra.

El objetivo de esta memoria es recopilar el uso de buenas prácticas y patrones de diseño a la hora de desarrollar la aplicación pedida, de forma que se facilite la corrección al profesor.

2. Diagrama de clases

Lo primero que quiero mostrar es el diagrama de clases, así se puede ver de forma sencilla y gráfica como he estructurado la aplicación.



3. Patrones Utilizados

Fachada: Probablemente uno de los patrones con aplicación más sencilla en nuestro caso. Al tener un sistema de almacenamiento (que además debe poder admitir distintos sistemas de almacenamiento en el futuro) estaba claro que íbamos a necesitar una fachada. En esta clase la clase "StorageFacade" hace de fachada para el sistema de almacenamiento, dentro del paquete "Storage".
Por otro lado, la clase "ListManager" hace también de fachada, pero esta vez para el paquete "Model". De forma que si cambiase el modelo de datos (de forma no muy muy significativa, claro) no haría falta cambiar la implementación del cliente que use "ListManager".

Singleton: El patrón fachado y singleton van casi de la mano, de forma que una fachada es bueno que solo tenga una instancia. De esta forma, las clases "ListaManager" y "StorageFacade" son Singleton. (Aunque he olvidado anotarlo en el diagrama de clases)

4. Buenas prácticas

Documentación: Es una de la práctica que cuando empezamos la carrera probablemente más tediosas nos parecen y que desde luego más importantes son. Si bien comencé el proyecto con una muy buena intención de documentar todo, la presión y las prisas han hecho que fuera dejándolo para el final y no pudiera completarlo.

Métodos cortos: Odio los métodos largos. Cuando veo algo de código por ahí con muchas líneas siempre pienso en refactorizarlo. En este proyecto he seguido esta máxima y a excepción del método main de la clase "TextMain", he hecho los métodos de menos de 20 líneas.

5. De cara a la segunda entrega

Principalmente hay dos cosas que quiero mejorar para la siguiente entrega:

Completar la documentación: Tanto para el código que me faltaba de esta primera entrega como para el código nuevo de la segunda.

Implementar más patrones: Me parece que no he sido capaz de encontrar suficientes situaciones en las que implementar patrones dados en clase, me gustaría encontrar alguno más. De todas formas, es mejor no forzar. Si no encuentro una forma decente, aplicar un patrón de forma forzada podría tener efectos contraproducentes.