

Programación Orientada a Objetos

Descripción general de la práctica del curso

Versión 2.1

Índice

1. Introducción	2
2. Requisitos técnicos	2
3. Objetivos	2
4. Planteamiento del problema	3
5. Casos de uso con funcionalidades del sistema	3

1. Introducción

En este curso se llevará a cabo un proceso de *desarrollo orientado a objetos* siguiendo un *modelo de ciclo de vida evolutivo incremental* guiado por *casos de uso*, es decir, por la funcionalidad que el sistema debe ofrecer a sus futuros usuarios.

El proceso completo de desarrollo abarca desde la planificación y especificación de requisitos, hasta el despliegue o instalación del sistema en el entorno previsto, pasando por su análisis, diseño, implementación y prueba. Es a estas últimas fases a las que dedicaremos la mayor parte de nuestra atención y esfuerzos; en particular, dado el carácter de la asignatura, nos centraremos en los aspectos relativos a la *implementación y prueba del software*.

El sistema se desarrollará en sucesivas etapas. En cada una se implementará un subconjunto de requisitos, correspondientes a un caso de uso, en el lenguaje de programación C++, a partir de un diagrama de clases. Tras ello, se deben realizar una serie de pruebas para asegurar que la aplicación, en su estado actual, funcione correctamente y satisfaga los requisitos especificados. Así, la funcionalidad del sistema irá incrementándose poco a poco y, al finalizar cada etapa, dispondremos de una parte del sistema en funcionamiento.

2. Requisitos técnicos

Para la implementación y prueba del sistema, es necesario disponer, en primer lugar, de un *entorno de ejecución* con una configuración regional que nos permita trabajar en español de España y codificación UTF-8, es decir, cuya *localización cultural* (en inglés, *locale*) sea `es_ES.UTF-8`. Además, se han de instalar previamente los paquetes correspondientes a:

1. Herramientas de uso ordinario, como visores, editores, etc.
2. La infraestructura de compilación (al menos, *clang*, *libclang-dev* y *llvm*).
3. Diversas utilidades de desarrollo (al menos, *make*, *gdb*, *duma* y *valgrind*).

En particular, la programación se realizará siguiendo la norma ISO/IEC 14882:2017 (conocida coloquialmente como C++17) para el lenguaje de programación C++, además de emplearse la versión 14 de la infraestructura de compilación Clang/LLVM.

Resulta muy conveniente emplear contenedores creados a partir de la imagen *Docker* suministrada por los profesores de la asignatura, que ya cumple todos los requisitos necesarios. Por otro lado, es lo aconsejable, pues **los profesores de la asignatura emplearán dichos contenedores para la corrección** de las prácticas.

3. Objetivos

Se trata de implementar un modelo de clases de cierta complejidad, de acuerdo con los principios de la programación orientada a objetos (abstracción, encapsulado, ocultación de datos, generalización y polimorfismo).

4. Planteamiento del problema

Una librería desea ofrecer su catálogo de libros y revistas a la venta por Internet. Para ello, se nos ha solicitado el desarrollo de una librería virtual que más adelante se utilizará para construir una aplicación Web. La librería virtual debe ser capaz de gestionar los pedidos y el carro de la compra, así como el catálogo en general. En lo que sigue, se describen casos de uso típicos, con funcionalidades que deberán implementarse.

5. Casos de uso con funcionalidades del sistema

Los casos de uso típicos que representan las funcionalidades que deben implementarse son los siguientes:

1. Gestión de usuarios y tarjetas.
2. Gestión del carro de la compra.
3. Facturación e histórico de las compras.
4. Gestión del catálogo de artículos.

Caso 1. Gestión de usuarios y tarjetas

De cada usuario se debe guardar su nombre y apellidos, así como su dirección postal. El usuario elegirá un nombre de usuario que debe ser único, y una *contraseña* que debe poseer cierta longitud mínima predefinida y estar compuesta por letras y números. Se deberá guardar información de una o más tarjetas de crédito que el usuario podrá elegir para pagar los diferentes pedidos que realice. De las tarjetas se guardará su tipo, número, titular (según aparece en la tarjeta), y fecha de expiración (mes/año).

Caso 2. Gestión del carro de la compra

Cada cuenta de usuario tiene asociada una cesta o «carro de la compra virtual», en el que el usuario puede ir añadiendo artículos que desea comprar. Este puede adquirir varias unidades de un mismo artículo, especificándolo al añadirlo al carro. En este punto, no se comprobarán las existencias de dichos artículos, pudiéndose añadir cuantas unidades se desee.

Caso 3. Facturación e histórico de compras

Facturación de la compra a partir de los contenidos del carro

Cuando el usuario pasa a facturar la compra (*checkout*), el carro «se vacía» y se genera un pedido. El usuario deberá elegir la tarjeta de crédito con la que pagará el pedido. Se guardará además la fecha del pedido. En este momento se asocia el pedido con la tarjeta de crédito y se actualizan las existencias disponibles de cada artículo. Es decir, si bien se puede ir añadiendo al carro lo que se desee, para finalizar la compra se han de comprobar y ajustar las existencias.

En caso de que no haya suficientes existencias de algún producto, se debe generar un error y vaciar el carro, dejando sin efecto la compra.

Listado del histórico de compras

Se permitirá listar todos los pedidos realizados hasta la fecha, ordenados y agrupados por nombre de artículo o fecha de compra, a elección del usuario, incluyendo la suma total de los importes de venta, bien sea por artículo o por fecha, según se elija. Nótese que debe contemplarse el hecho de que los precios de venta de los artículos pueden ser diferentes de sus precios actuales.

Caso 4. Gestión del catálogo de artículos

Se pueden añadir al catálogo tres tipos de artículos: libros, revistas y libros digitales. Todos ellos tienen un código de referencia interno, un precio actual, un título y una fecha de publicación, pero solo de los libros y revistas hay que almacenar el número de artículos disponibles (sus existencias o *stock*), ya que los libros digitales se distribuyen mediante su descarga a través de Internet. De todos los artículos también se guardan los autores, que puede ser uno o varios, y que se emplearán como criterio de búsqueda. Los libros digitales tienen además una fecha de expiración en la que dejan de venderse, ya que se consideran obsoletos. De los libros se almacena el número de páginas, y de las revistas, su número y periodicidad.