

**Grado en Ingeniería de Computadores – Curso 2021-22**

**Ingeniería del Software: Laboratorio**

**PECL1 Análisis y Diseño de software**

**Enunciado:**

Se trata de realizar el análisis y diseño de la segunda versión de una aplicación de gestión de venta de libros, cuya primera versión la realizó previamente el profesor. A continuación, se resume la nueva información extraída de las conversaciones con el dueño del producto, que hay que tener en cuenta en la segunda versión, cuyo análisis y diseño hay que realizar.

*El empleado nos pide que la aplicación también permita el procesamiento de los libros enviados por la imprenta. En el momento en que se reciba un paquete de ejemplares de un libro de la imprenta, con su correspondiente factura de impresión, quiere poder utilizar la aplicación para comprobar si existen pedidos pendientes asociados a dicho libro y, en este caso, finalizar el pedido y facturarlo al cliente. Quiere que además de las facturas de los clientes ya tratadas en la primera versión de la aplicación, el sistema también almacene los datos de las facturas de la imprenta. El empleado de la tienda nos aclara que la factura de impresión sólo se puede referir a un libro, y la información que incluye es el código de la orden de impresión original, el código del libro, su título, la cantidad de ejemplares servidos, el precio unitario y el precio total. También nos aclara que tengamos en cuenta que el precio de los libros de la imprenta no es el mismo que el de venta al público que está almacenado en la base de datos, que el precio de la imprenta es el que llega en la factura de la imprenta.*

*El responsable quiere poder realizar un informe de estado de cuentas mensual; nos aclara que este informe no lo debe poder generar el empleado. Quiere poder indicar el mes para el que desea el informe y que se muestre el informe en la pantalla. En dicho informe se reflejarán tanto los gastos como los beneficios durante el mes en términos de libros vendidos y libros comprados a la imprenta. El informe de estado de cuentas incluirá la fecha del informe, la lista de libros vendidos (reflejando el código, título, cantidad de ejemplares cada libro vendido y su precio unitario), el importe total de la venta de esos libros, una lista de los libros comprados a la imprenta (con el código, título, cantidad de ejemplares comprados de cada libro y su importe unitario), el importe total de las compras, y finalmente la diferencia entre el importe de las ventas y el de las compras. También desea que en el informe se presenten dos gráficas sobre la evolución temporal durante los últimos doce meses: una con el número de libros vendidos y libros comprados a la imprenta, y otra con los gastos y beneficios. Este informe se almacenará en otra base de datos diferente de la usada para el resto de funciones de la aplicación. El responsable nos avisa de que la base de datos para los informes de estado de cuentas trabaja con una tecnología diferente a la otra base de datos, en lugar de MySQL, se trata de una base de datos con tecnología Apache Cassandra. Y nos aclara que esta nueva base de datos se instalará en la misma máquina en la que está la otra.*

*El responsable explica que se va a comprar un nuevo ordenador portátil que sólo usará él, en el que también se deberá instalar la aplicación. Se trata de un ordenador portátil y por tanto diferente a los otros existentes en la tienda, y además se instalará otra una impresora para uso exclusivo desde este ordenador sin compartir con el resto de ordenadores de la tienda. Además, nos indica que van a comprar un tercer ordenador para la tienda en el que también debe estar instalada la aplicación.*

*El diseñador ha decidido que la funcionalidad sobre el informe de estado de cuentas se empaquete en un archivo jar diferente a los que incluyen el resto de la aplicación. También ha decidido utilizar la librería* [*Datastax Java driver*](http://cassandra.apache.org/doc/latest/getting_started/drivers.html) *para acceder a la base de datos de informes de estado de cuentas.*

**Se pide:**

1. (15%) Completar el **catálogo de requisitos** de la primera versión, con los nuevos requisitos.
2. (10%) Completar el **diagrama de casos de uso** de la primera versión, teniendo en cuenta los nuevos requisitos.
3. (20%) Describir los nuevos casos de uso, utilizando la misma plantilla que en la primera versión, e incluyendo un **prototipo de las interfaces de usuario** (pantallas) en aquellos casos de uso que corresponda.
4. (30%) Completar los **diagramas de clases** de la primera versión, teniendo en cuenta los nuevos requisitos, y aplicando el patrón de diseño Modelo-Vista-Controlador (**MVC**).
5. (15%) Realizar un **diagrama de secuencia** completo para uno de los nuevos casos de uso, que incluya objetos del modelo, de la vista y del controlador.
6. (5%) Completar el **diagrama de componentes**, teniendo en cuenta los nuevos requisitos.
7. (5%) Completar el **diagrama de despliegue**, teniendo en cuenta los nuevos requisitos.

**Forma de entrega:**

El trabajo debe enviarse a través del Campus virtual, desde la sección Trabajos > PECL1.

Debe enviarse un único archivo comprimido (zip o rar) que contenga los siguientes archivos:

* *TiendaLibrosv2(ApellidoNombredelAlumno).eap*. Archivo que debe contener los diagramas realizados por el profesor para la primera versión, con las modificaciones y nuevos diagramas necesarios para la segunda versión realizados por el alumno. **NOTA: Sólo se calificarán las modificaciones y los nuevos diagramas para la segunda versión de la aplicación.**
* *PECL1(ApellidoNombredelAlumno).docx*. Con el informe del trabajo, utilizando la plantilla proporcionada.

**Trabajo individual:**

El trabajo debe realizarse de forma individual.

Pero se da la opción de poder realizarlo en pareja, si además de los diagramas y el informe, se programa el software de la aplicación en Java, usando una base de datos a elección de los alumnos o bien guardando la información de libros, clientes, etc. en archivos.

Si se quiere realizar en pareja debe solicitarse al profesor indicando qué sistema de almacenamiento se utilizará (base de datos o archivos), quien confirmará si se acepta hacerlo en pareja.

Si se realiza en pareja, los dos alumnos deben enviar el mismo trabajo a través del campus virtual.

**Aclaraciones:**

Sobre los requisitos:

* No hay que copiar y pegar el texto literal del enunciado. Cada requisito debe ser auto contenido, no hay que ir a leer en otro texto. Por ejemplo: “En dicho informe se reflejarán...” no es válido. Debe ser “En el informe de estado de cuentas se reflejarán…”.
* Hay que evitar comentarios o información irrelevante. Ejemplo: “También nos aclara que tengamos en cuenta que el precio de los libros de la imprenta…”. Debe ser: “El precio de los libros de la imprenta…”.
* Un requisito debe tener sólo una frase.
* En el texto de un requisito no debe aparecer el nombre de la fuente, para eso está el campo fuente en el catálogo. Ejemplo: “El responsable quiere que el sistema genere...” debe ser “El sistema debe generar…”.

Sobre los casos de uso:

* Los casos de uso deben tener un verbo.
* Si se hace algún cambio que afecte a algún caso de uso de la versión 1, hay que actualizar su ficha, marcando en amarillo lo que se ha modificado.

Sobre los diagramas de clases:

* Las asociaciones en los diagramas de clases deben tener punta de flecha en el extremo que corresponda, y nombre sólo en el extremo de la flecha.
* No debe haber atributos duplicados, como ya se cumple en los diagramas de la versión 1. En la que, por ejemplo, si el título de un libro está en la clase Libro, no está en la clase FacturaVenta. En su lugar, en la estructura de clases debe haber un camino para llegar a dicho valor. En este caso, desde la clase FacturaVenta > Pedido > DetalleLibroPedido > Libro > tituloLibro.
* En la herramienta de modelado, las nuevas clases deben ubicarse en las carpetas adecuadas: modelo, vista o controlador.

Sobre el diagrama de componentes:

* En el diagrama de componentes deben aparecer las clases dentro de cada componente.

Sobre el diagrama de despliegue:

* En el diagrama de despliegue deben aparecer los componentes dentro de los nodos.

Otras aclaraciones:

* En todos los diagramas usar un color de relleno diferente en los nuevos elementos añadidos en la versión 2, para distinguirlos de los que ya había en la versión 1.
* Debe actualizarse el índice del trabajo antes de enviarlo.