

Unidad 2: Modelado I

Tema 3: Modelado de Estructura del Sistema

1. Gestión de Pedidos

Modelar el esquema conceptual de datos mediante un diagrama de clases para el sistema gestión de pedidos que se describe a continuación.

Se desea crear un sistema de gestión de pedidos, sabiendo que:

- *Un cliente puede realizar varios pedidos en un periodo de tiempo (un pedido es realizado por un solo cliente).*
- *Cada pedido está formado por varias líneas de pedido, cada una de las cuales se refiere a un solo producto.*
- *Se diferencian dos tipos de clientes, el cliente personal y el cliente corporativo. La diferencia entre los dos tipos de clientes es que el cliente personal pagará mediante una tarjeta de crédito, mientras el cliente corporativo tiene un contrato con la empresa y un límite de crédito.*
- *Además, los vendedores de la empresa se encargan de atender las peticiones de los clientes corporativos, de forma que cada vendedor se hace cargo de una cartera de clientes corporativos, y a cada cliente corporativo sólo le atiende un vendedor.*

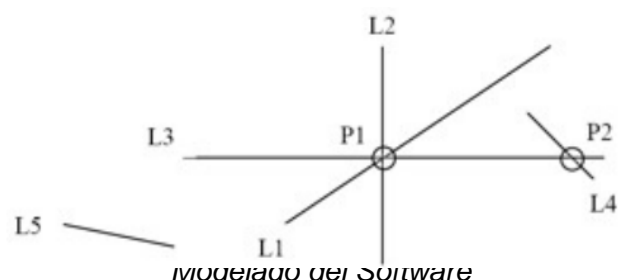
2. Líneas de dos Dimensiones

Modelar un sistema que contiene una colección de líneas en dos dimensiones y que permite la siguiente funcionalidad:

- Dada una línea, buscar todas la líneas que la cortan.
- Dado un punto, buscar todas las líneas que pasan por él.

Se pide:

- Realizar un diagrama de clases para mostrar las clases y asociaciones



- Realizar un diagrama de objetos con instancias y enlaces para el caso de ejemplo de la figura.

3. Venta de Coches de Segunda Mano

A continuación se describe un sistema de gestión de una empresa de venta de coches de segunda mano.

- *Los coches los suministran distintos proveedores, interesando conocer la marca, modelo, matricula, precio de compra y precio de venta.*
- *Los coches pueden ser turismos, industriales y todoterrenos. Además pueden necesitar ser reparados, por lo que se debe tener un control de las reparaciones hechas, que pueden ser mecánicas, eléctricas o de chapa.*
- *En la empresa hay dos tipos de vendedores: asalariados y por comisión. De los asalariados interesa saber también el salario y de los que van con comisión los coches que han vendido.*
- *Además se tendrá un control de los clientes, tanto de los que han comprado un coche, como de los interesados en algún tipo de coche del que podrán hacer reservas.*
- *Los coches pueden estar en distintas exposiciones, y se necesita saber en todo momento donde se encuentra cada coche.*
- *Se necesitan operaciones para realizar una venta de un coche, para reparar los coches que lo necesiten, para comprar nuevos coches a los proveedores, etc.*
- *También interesa tener operaciones que devuelvan que cliente compra un cierto coche, y que se realicen listados de los coches que se encuentran en stock en un momento dado.*

Se pide:

- 1) Modelar la estructura del sistema mediante un diagrama de clases.
- 2) Realizar un diagrama de objetos (con instancias tomadas del mundo real) para modelar una estructura de objetos que represente el estado en un momento concreto del sistema.

Tema 4: Modelado de Interacciones del Sistema

1. Procesamiento de pedidos

Por medio de un diagrama de secuencia, realizar el modelado de comportamiento del procesamiento de pedidos en un sistema de compras por Internet, cuya descripción se proporciona a continuación:

- *El cliente introduce en primer lugar su nombre y dirección.*
- *Si el cliente introduce el ZIP, el sistema detecta la ciudad y región, que son introducidas automáticamente en el formulario.*
- *El cliente introduce los códigos de los productos que sea incluir en el pedido.*
- *Por cada código de producto introducido, el sistema aporta la descripción y precio del producto, así como realizar la suma al total del pedido.*
- *Una vez finalizada la introducción de productos, el sistema almacena temporalmente la lista de productos incluidos en el pedido.*
- *Para proceder con la confirmación del pedido, el cliente debe introducir la información de la tarjeta de pago.*
- *El cliente pulsa el botón que confirma el pedido, lo que provoca al sistema realizar las verificaciones oportunas para comprobar que todo está correcto en el pedido. Entre estas comprobaciones, se encuentra recibir la confirmación del banco.*
- *Si el sistema detecta algún error en el pedido, se solicita al cliente su corrección.*
- *Cuando el pago es confirmado, se acepta el pedido, se almacena y se le asigna un ID que se retorna al cliente.*

2. Máquina expendedora

Dibujar el diagrama de secuencia o de colaboración del siguiente escenario:

El usuario de una máquina expendedora de bebidas introduce las monedas necesarias de acuerdo al precio del producto deseado. Selecciona Coca-Cola, pero la máquina no tiene este tipo de bebida y lo manifiesta a través de un mensaje en la pantalla de la máquina. A continuación el usuario elige Fanta que sí está disponible y además es más barata. La máquina suministra la lata de Fanta y devuelve el cambio.

Nota: Podéis tratar de identificar los diferentes elementos que forman la máquina, como los botones, la pantalla, el cajetín de las monedas, etc.

3. Gestión de Pedidos

A continuación se describe detalladamente un conjunto de clases, interfaces y métodos junto con su funcionamiento:

- **ICliente** es una interfaz que especifica un método llamado **service()**.
- **Clase** es una clase que incluye una referencia a un objeto **another** de tipo **OtraClase** y un método llamado **useful()** que invoca a **helper()** sobre el objeto **another**.
- **OtraClase** contiene un atributo **text** que se inicializa con el String “¿Necesitas ayuda?”, y un método **helper()** que simplemente imprime el contenido de **text** en la salida estándar o **System.out**.
- **MiClase** es una clase que implementa **ICliente** y que hereda de **Clase**. La implementación de **service()** simplemente invoca a **useful()** sobre sí misma.
- El método **main()** debe crear primero un objeto de **MiClase** y luego una instancia de **OtraClase** que se asigna al campo **another** del primero (del objeto de **MiClase**). Finalmente el método **main()** llama a **service()** sobre la instancia de **MiClase**.

Dibujar el diagrama de secuencia (objetos y mensajes) que se desencadena a partir de la ejecución de **main()**. Se debe desarrollar previamente un diagrama de clases que refleje la estructura que se explica puede ser de ayuda.