

# EJERCICIOS DE UML

# Ejercicio #1

Representa mediante un diagrama de clases la siguiente especificación:

- Una aplicación necesita almacenar información sobre empresas, sus empleados y sus clientes. Ambos se caracterizan por su nombre y edad.
- Los empleados tienen un sueldo bruto, los empleados que son directivos tienen una categoría, así como un conjunto de empleados subordinados.
- De los clientes además se necesita conocer su teléfono de contacto.
- La aplicación necesita mostrar los datos de empleados y clientes

## Ejercicio #2

- Una biblioteca tiene copias de libros. Estos últimos se caracterizan por su nombre, tipo (novela, teatro, poesía, ensayo), editorial, año y autor.
- Los autores se caracterizan por su nombre, nacionalidad y fecha de nacimiento.
- Cada copia tiene un identificador, y puede estar en la biblioteca, prestada, con retraso o en reparación.
- Los lectores pueden tener un máximo de 3 libros en préstamo.
- Cada libro se presta un máximo de 30 días, por cada día de retraso, se impone una “multa” de dos días sin posibilidad de coger un nuevo libro.
- Realiza un diagrama de clases y añade los métodos necesarios para realizar el préstamo y devolución de libros.

## Ejercicio #3

Especificar un diagrama de clases que describa redes de ordenadores.

- Los elementos que se pueden incluir en la red son:
  - Servidor, PC, Impresora.
  - Hub, Cable de red.
- Los PCs pueden conectarse con un único Hub, los servidores con uno o varios.
- Los Servidores y PCs pueden generar mensajes, con una cierta longitud.
- Los Hubs tienen un número de puertos, algunos de los cuales puede usarse para conectar con otros Hubs.
- Tienen cierta probabilidad de “perder” mensajes.
- Las impresoras pueden averiarse, con cierta probabilidad, durante cierto tiempo.

# Ejercicio #4

Realiza el diseño de una aplicación para la gestión de pedidos. La aplicación deberá:

- Manejar clientes (se guarda su nombre, dirección, teléfono y e-mail), que pueden realizar pedidos de productos, de los cuales se anota la cantidad en stock. Un cliente puede tener una o varias cuentas para el pago de los pedidos. Cada cuenta está asociada a una tarjeta de crédito, y tiene una cierta cantidad disponible de dinero, que el cliente debe aumentar periódicamente para poder realizar nuevos pedidos.
- Un cliente puede empezar a realizar un pedido sólo si tiene alguna cuenta con dinero disponible. Al realizar un pedido, un cliente puede agruparlos en pedidos simples o compuestos. Los pedidos simples están asociados a una sola cuenta de pago y (por restricciones en la distribución) contienen un máximo de 20 unidades del mismo o distinto tipo de producto. A su vez, un pedido compuesto contiene dos o más pedidos, que pueden ser simples o compuestos. Como es de esperar, el sistema debe garantizar que todos los pedidos simples que componen un pedido compuesto se paguen con cuentas del mismo cliente. Además, sólo es posible realizar peticiones de productos en stock.
- Existe una clase (de la cual debe haber una única instancia en la aplicación) responsable del cobro, orden de distribución y confirmación de los pedidos. El cobro de los pedidos se hace una vez al día, y el proceso consiste en comprobar todos los pedidos pendientes de cobro, y cobrarlos de la cuenta de pago correspondiente. Si una cuenta no tiene suficiente dinero, el pedido se rechaza (si es parte de un pedido compuesto, se rechaza el pedido entero). Una vez que el pedido está listo para servirse, se ordena su distribución, y una vez entregado, pasa a estar confirmado.

# Ejercicios Diagrama de Caso de Uso

# Ejercicio #1

Una empresa gestiona un conjunto de inmuebles, que administra en calidad de propietaria. Cada inmueble puede ser bien un local (local comercial, oficinas, etc.), un departamento o bien un edificio que a su vez tiene departamentos y locales. Como el número de inmuebles que la empresa gestiona no es un número fijo, la aplicación debe permitir tanto introducir inmuebles nuevos, así como darlos de baja, modificarlos y consultarlos. Asimismo, que una empresa administre un edificio determinado no implica que gestione todos sus departamentos y locales, por lo que la aplicación también deberá permitir introducir nuevos departamentos o locales, darlos de baja, modificarlos y hacer consultas sobre ellos. Cualquier persona que tenga una nómina, un aval bancario, un contrato de trabajo o venga avalado por otra persona puede alquilar el edificio completo o alguno de los pisos o locales que no estén ya alquilados, y posteriormente desalquilarlo. Por ello, deberán poder ser dados de alta, si son nuevos inquilinos, con sus datos correspondientes (nombre, DNI, edad, sexo, ...), poder modificarlos, darlos de baja, consultarlos, etc. La aplicación ofrece acceso web para que un inquilino puede modificar o consultar sus datos, pero no darse de baja o de alta. Para la realización de cualquiera de estas operaciones es necesaria la identificación por parte del inquilino.

# Ejercicio #2

Se desea diseñar el software necesario para una red bancaria provista de cajeros automáticos (ATMs), que serán compartidos por un consorcio de bancos. Cada banco dispone de una serie de servidores, provistos de software propio, que llevan la información sobre sus cuentas y procesa las transacciones que actúan sobre dichas cuentas. A estos servidores están conectados las estaciones de cajero, que son propiedad del banco y en las que operan cajeros humanos, que pueden crear cuentas e introducir transacciones sobre ellas.

Los cajeros automáticos aceptan tarjetas de crédito, interaccionan con el usuario, se comunican con un ordenador central para llevar a cabo las transacciones, entregan dinero en efectivo al usuario e imprimen recibos.

El sistema llevará el registro de las transacciones efectuadas, cumplirá características aceptables de seguridad y manejará accesos concurrentes a la misma cuenta.

El coste de desarrollo de la parte compartida del sistema se dividirá entre los bancos que forman parte del consorcio en función del número de los bancos que forman parte del consorcio en función del número de clientes provistos de tarjetas de crédito

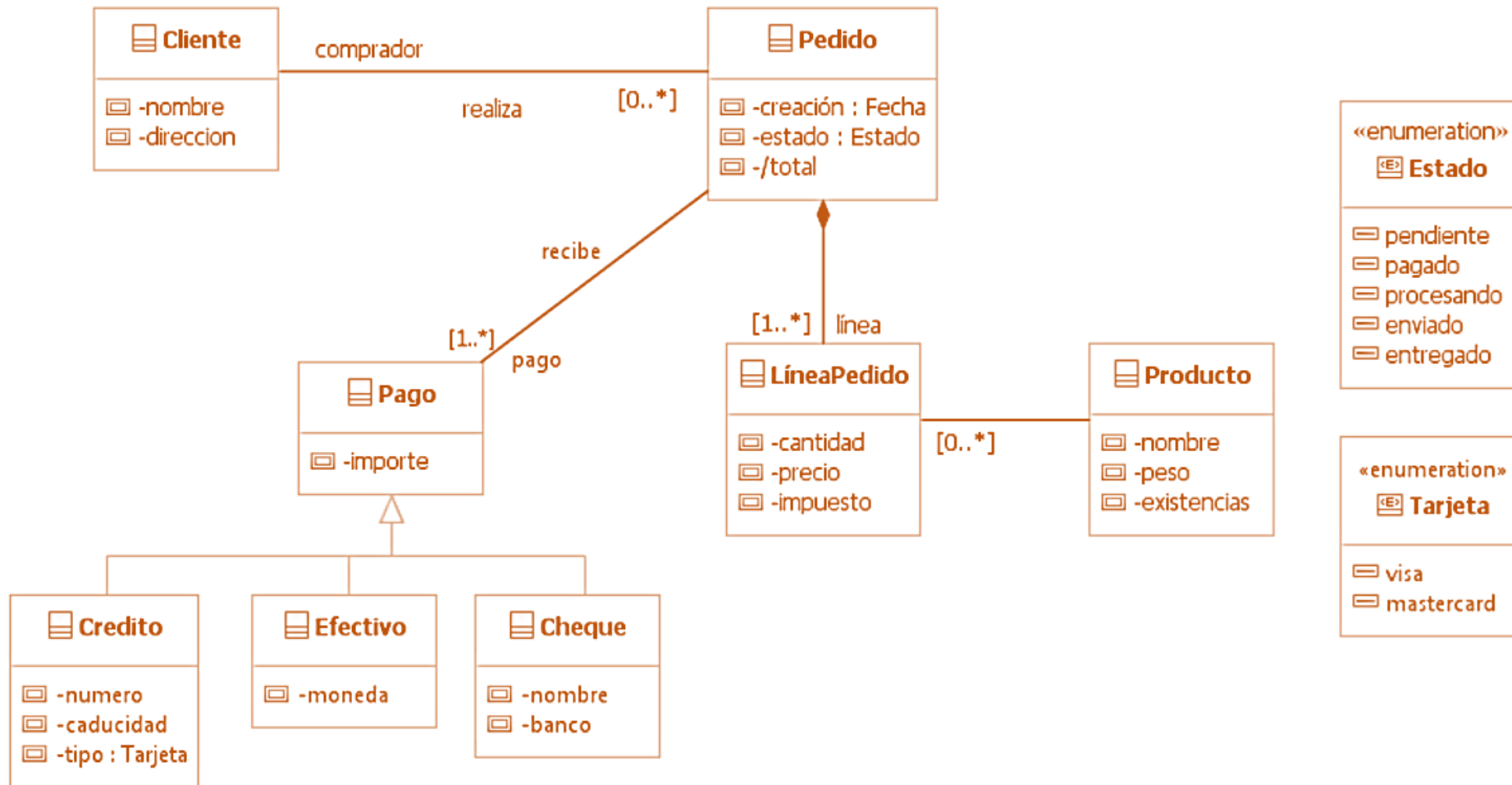


# Ejercicios Diagrama de Secuencia

# CASO

- Una aplicación debe guardar la información de los pedidos que realizan sus clientes y la fecha en que se formalizan.
- Los pedidos constan habitualmente de varios productos.
- El coste total del pedido se calcula a partir de los precios individuales de cada producto, así como de la cantidad e impuestos asociados a cada uno.
- Es importante mantener información sobre las existencias de cada producto con el fin de poder informar al cliente si habrá retrasos en la entrega del pedido.
- El pedido podrá pagarse de una vez o en varios pagos.
- Las formas de pago posible son
  - Tarjeta de crédito (fecha de caducidad, número, VISA o MASTERCARD)
  - Efectivo (moneda)
  - Cheque (nombre, entidad bancaria)
- El pedido podrá estar en uno de los siguientes estados: pendiente, pagado, procesando, enviado y entregado

# Diagrama de Clases



# Ejercicio 1 : Diagrama de Secuencia UNIVERSIDAD DE LIMA

**Identificador:** Realiza\_pedido

**Actor :** Cliente

**Precondiciones:** Un cliente registrado en el sistema ha accedido correctamente al sistema

**Secuencia de eventos de flujo:**

1. El cliente solicita la realización de un pedido
2. El sistema pedidos muestra al cliente un listado con los productos
3. El cliente selecciona el producto que desea incluir en el pedido
4. El sistema muestra la descripción y el precio del producto al cliente
5. El cliente confirma la selección
6. El sistema devuelve el id del pedido al usuario

**Postcondiciones:** Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente

*\* Describir el diagrama de secuencia para CUS especificado*

# Ejercicio 2 : Diagrama de Secuencia

**Identificador:** Pagar\_pedido

**Actor que lo inician:** Cliente

**Precondiciones:** Un cliente registrado en el sistema ha accedido correctamente al Sistema y va a pagar un pedido realizado previamente

## **Secuencia de eventos de flujo:**

- El cliente solicita pagar un pedido
- El sistema pedidos muestra al cliente un listado con los pedidos pendientes de pago
- El cliente selecciona el pedido a pagar
- El sistema muestra el resumen del pedido
- El cliente confirma el pedido
- El sistema solicita que se seleccione una forma de pago
- El cliente selecciona una forma de pago e introduce sus datos
- El sistema de pago contacta con el banco para comprobar los datos
- El pago es confirmado se acepta el pedido y se le comunica al cliente

**Postcondiciones:** Si el pedido no ha sido cancelado, es registrado como pagado en el sistema

***\* Describir el diagrama de secuencia para CUS especificado***

# Ejercicio 3 : Diagrama de Secuencia

**Identificador:** Realiza\_pedidos2

**Actor que lo inician:** Cliente

**Precondiciones:** Un cliente registrado en el sistema ha accedido correctamente al sistema.

**Secuencia de eventos de flujo:**

- El cliente solicita la realización de un pedido
- El sistema pedidos muestra al cliente un listado con los producto
- El cliente selecciona los productos que desea incluir en el pedid
- Por cada producto
  - El sistema crea una línea de pedido sobre la que se aporta descripción, precio y cantidad
  - Dicha línea de pedido se asocia al pedido
  - El sistema suma el precio del producto en el total del pedido
- El cliente confirma la selección
- El sistema devuelve el id del pedido al usuario

**Postcondiciones:** Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente

**\* Describir el diagrama de secuencia para CUS especificado**

