

Principios de Diseño OO



Resumen

- UML
- Diagramas
 - Aspectos generales
 - Clasificación: estructurales vs. comportamiento
 - Tipos de diagramas:
 - Casos de uso
 - Clases
 - Interacción: secuencia y colaboración
 - Estados
 - Actividades
- Problemas sencillos de diseño



Objetivos

- Conocer y comprender cómo realizar un modelo de un sistema SW OO
- Conocer y comprender qué es UML y cómo utilizarlo
- Conocer y comprender los distintos tipos de diagramas UML
- Diseñar pequeñas aplicaciones en Java usando UML



UML



UML – Aspectos generales

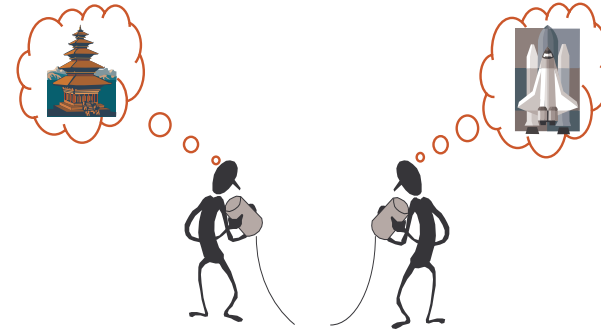
- UML permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos
- Intenta abarcar:
 - Especificación de requisitos (recogida de información)
 - Análisis (qué hay que hacer)
 - Diseño (cómo hay que hacerlo)
 - Despliegue (implementación)
- No es un método, ni un proceso ni una metodología
- No establece qué modelos construir



Programación Orientada a Objetos



UML - ¿Por qué?



- Ofrece una forma estándar de especificar de manera visual y textual sistemas



Programación Orientada a Objetos



Diagramas



Programación Orientada a Objetos



Diagramas – Aspectos Generales

- Representaciones gráficas de un conjunto de elementos con relaciones entre ellos
- Hay varios tipos de diagramas cada uno de los cuales ofrece una vista particular de los elementos que configuran el sistema
- Un mismo elemento de un sistema puede aparecer varias veces en el mismo diagrama o en diagramas diferentes
- ¿Para qué se utilizan?
 - Especificar modelos y a partir de ellos construir un sistema (ingeniería directa)
 - Reconstruir modelos a partir de partes de un sistema ejecutable (ingeniería inversa)



Programación Orientada a Objetos



Diagramas – Tipos - Estructurales

- Estructurales (Aspectos ESTÁTICOS)
 - Modelan los elementos del dominio (clases y objetos) y las relaciones que existen entre los mismos
 - Utilizados durante la fase de diseño de un sistema
- Tipos de diagramas estructurales:
 - Clases: Representa la estructura estática del sistema en términos de clases y relaciones
 - Componentes: Representan los componentes SW de una aplicación
 - Despliegue: Representan la ejecución de los componentes sobre dispositivos concretos



Diagramas – Tipos - Comportamiento

- Comportamiento (Aspectos DINÁMICOS)
 - Casos de uso: Funciones del sistema desde el punto de vista del usuario
 - Secuencia (interacción): Representación temporal de los objetos y sus interacciones
 - Colaboración (interacción): Representación espacial de los objetos, enlaces e interacciones
 - Estados: Comportamiento de una clase en términos de estados
 - Actividad: Comportamiento de una operación en términos de acciones



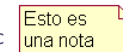
Diagramas – Ciclo de vida

FASE DE CICLO DE VIDA	DIAGRAMAS TÍPICOS
Especificación de requisitos	Casos de uso
Análisis	Diagrama de Clases Diagrama de Secuencia Diagrama de Colaboración
Diseño	Diagrama de Estados Diagrama de Actividades Diagrama de Componentes
Despliegue	Diagrama de Despliegue



Diagramas – Elementos Comunes

- Notas:
 - Puede contener comentarios, restricciones, etc
 - Pueden aparecer solas o unidas a alguno de los elementos del diagrama mediante una línea discontinua



- Paquetes:
 - Mecanismo de propósito general para organizar elementos en grupos
 - Cualquier grupo de elementos, sean estructurales o de comportamiento, puede incluirse en un paquete. Incluso pueden agruparse paquetes dentro de otro paquete.

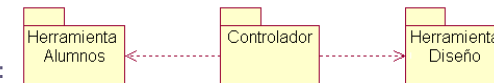


Diagrama de Casos de Uso



Diagrama de Casos de Uso – Aspectos Generales

- Conjunto de casos de uso, actores y relaciones
- Recogida de información para modelar el sistema (**¿Qué?**)
- Representan las acciones a realizar, relaciones entre ellas y relaciones entre actores (no indican la secuencia en la que se realizarán las acciones)
- No son exclusivos de la OO
- Elementos:
 - Actores
 - Casos de uso
 - Relaciones



Diagrama de Casos de Uso – Actores

- Cualquier usuario, rol o aplicación con un comportamiento propio que interactúa con el sistema



Usuario

- ¿Cómo encontrar los actores?
 - Identificar los usuarios y administradores de la aplicación
 - Identificar los roles de los usuarios de la aplicación
 - Identificar los sistemas y/o aplicaciones que interactúan con la aplicación
- El nombre del actor describe el papel desempeñado (la misma persona física puede interpretar varios papeles)



Diagrama de Casos de Uso – Casos de Uso

- Descripción de un conjunto de acciones que ejecuta una aplicación para producir un resultado relevante para un actor (operación o funcionalidad del sistema)
- Los casos de uso representan las funcionalidades del sistema



Gestión de Ventas

- ¿Cómo encontrar los casos de uso por cada actor?
 - Identificar las tareas principales de los actores
 - Identificar la información a consultar / modificar / eliminar
 - Identificar la información a suministrar
 - Actor al sistema
 - Sistema al actor



Diagrama de Casos de Uso – Ejemplo 1

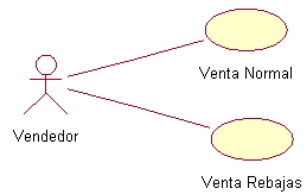
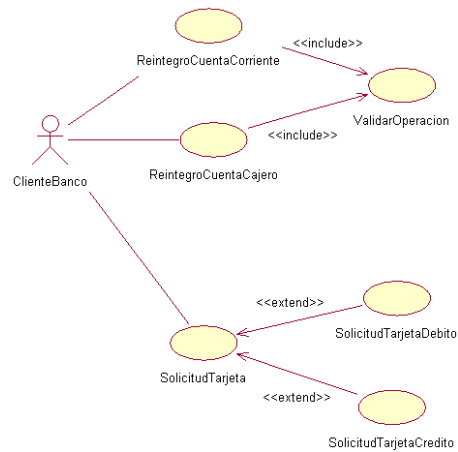


Diagrama de Casos de Uso - Relaciones entre casos de uso

- Asociación: Existe una relación
- Inclusión:
 - Un caso de uso puede incluir otro
 - Etiqueta `<<include >>`
- Extensión:
 - El caso de uso origen extiende el comportamiento del caso de uso destino
 - Etiqueta `<<extend>>`



Diagrama de Casos de Uso – Ejemplo 2



Ejercicio 1 - Diagrama de Casos de Uso

- Una persona desea ir al médico porque se encuentra mal y quiere que le receten un medicamento en EEUU
- Para ir al médico, se debe pedir una cita por teléfono. En cualquier momento, se puede anular una cita previamente solicitada
- Una vez que el paciente está en el médico, el doctor comprueba su historial clínico en la base de datos del sistema antes de expedir una receta
- Cada persona debe pagar sus facturas sanitarias contratando un seguro privado o efectuando el pago de cada una de las prestaciones sanitarias al banco del hospital, permitiéndose el pago aplazado



Diagrama de clases



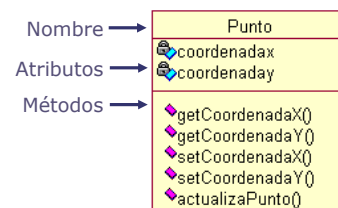
Diagrama de clases – Aspectos Generales

- Sirven para:
 - Definir el conjunto de clases que componen el sistema
 - Establecer las relaciones entre clases, interfaces, etc
 - Modelar un esquema lógico de una BD
- Elementos:
 - Clases
 - Relaciones
 - Paquetes
 - Interfaces



Diagrama de clases - Clases

- Representación:
 - Nombre
 - Atributos: Propiedades de la clase
 - Métodos: Implementación de un servicio a una clase



- Acceso / Visibilidad:
 - Privado (-): Solamente la clase puede utilizar este atributo
 - Protegido (#): Las clases hijas podrán utilizar el atributo
 - Público (+): Cualquier clase podrá utilizar dicho atributo

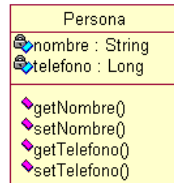


Diagrama de clases – Atributos y métodos

- Atributos:
 - Propiedades de la clase
 - Representación: acceso, nombre (obligatorio), tipo, valor inicial por defecto
- Métodos:
 - Implementación de un servicio a una clase
 - Representación: acceso, nombre (obligatorio), parámetros, valor de retorno



Diagrama de clases – Ejemplo 1



- Conceptual: "Una persona tiene un nombre y un número de teléfono"
- Especificación: "Una persona puede almacenar y consultar su nombre y teléfono"
- Implementación: "Una persona tiene un campo de tipo String que almacena su nombre y dos métodos para consultarlo y almacenarlo"



Diagrama de clases – Clases y Objetos

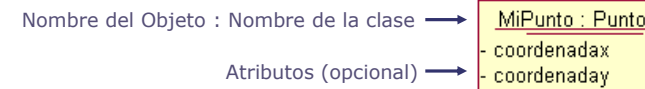
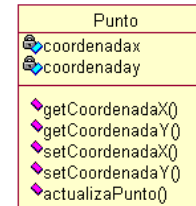
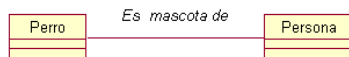


Diagrama de clases – Relaciones

- Asociación: Los objetos de una clase están relacionados con los de otra clase



- Asociación con dirección

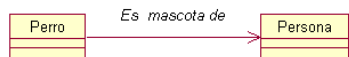
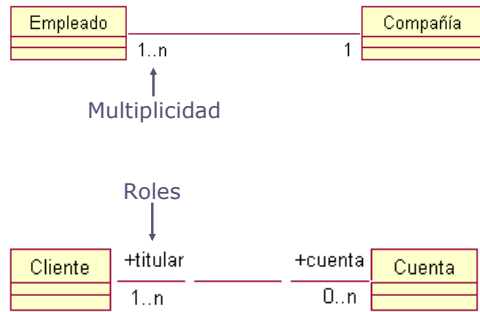


Diagrama de clases – Relaciones - Multiplicidad

- Multiplicidad:
 - Restricción sobre una asociación
 - Limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase
 - Tipos:
 - 1
 - 0..1
 - 0..* / *
 - 1..*
 - m..n
 - 2..*
 - 3, 4, 6..*
 - 3

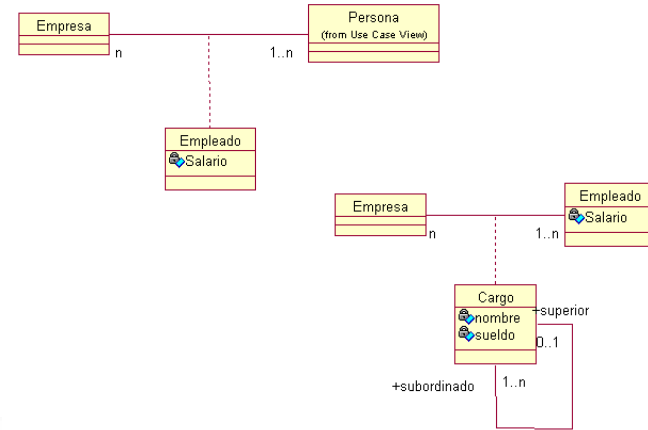


Diagrama de clases – Asociaciones - Multiplicidad



Programación Orientada a Objetos

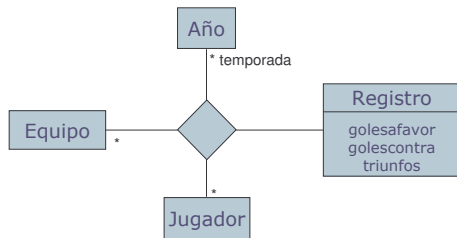
Diagrama de clases – Clases Asociativas



Programación Orientada a Objetos

Diagrama de clases – Relaciones N-Arias

- Determinan relaciones entre tres o más clases
- La multiplicidad se define dado un extremo, relativo a otros extremos
- No están recomendadas en el análisis y diseño
- A veces son demasiado ambiguas para implementarlas



Programación Orientada a Objetos

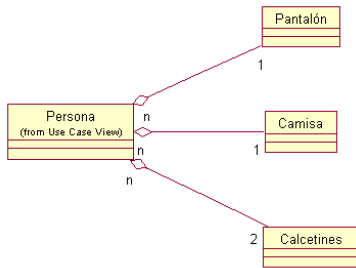
Diagrama de clases – Agregación y Composición

- Una agregación se caracteriza por cumplir alguno de los siguientes criterios:
 - Una clase forma parte de otra
 - Los valores de los atributos de una clase se propagan en los valores de los atributos de otra clase
 - Una acción sobre una clase implica una acción sobre otra
 - Los objetos de una clase se encuentran subordinados a los de otra

Programación Orientada a Objetos

Diagrama de clases – Agregación

- Agregación (débil) - "Tiene":
 - Las partes pueden compartir varios "todos"
 - Si desaparece el "todo" puede permanecer sus "partes"

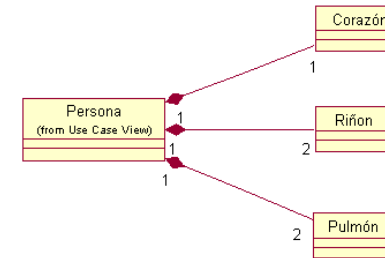


Programación Orientada a Objetos



Diagrama de clases – Composición

- Composición (agregación fuerte) - "Se compone":
 - Si desaparece el "todo" desaparecen sus partes



Programación Orientada a Objetos



Ejercicio 2 - Diagrama de clases – Agregación / Composición

- Especificar si las siguientes relaciones son de agregación o de composición, y cual es la multiplicidad de la relación:
 - Caso 1: Un barco puede pertenecer a cero o una flotas y una flota debe tener al menos un barco
 - Caso 2: Relación entre la clase libro y la clase capítulo de libro
 - Caso 3: Relación entre la clase matrimonio y la clase persona
 - Caso 4: Un almacén de repuestos tiene relaciones con las clases cliente y con la clase cuenta
 - Caso 5: Un sitio web está formado por distintas páginas web

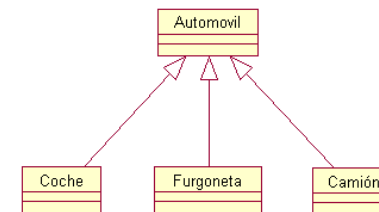


Programación Orientada a Objetos



Diagrama de clases – Generalización o Herencia

- Permite definir una clase a partir de otra
- UML soporta herencia simple como múltiple
- Asociación característica de los modelos OO



Programación Orientada a Objetos



Diagrama de clases – Clases Abstractas

- Define una clase no instanciable pero sí extensible
- Pueden contener:
 - Atributos
 - Métodos e implementación de los mismos (se permite sobreescritura)
 - Prototipos de métodos (obligatoriamente implementados por las subclases)
- Atributos, métodos y prototipos serán heredados por las subclases

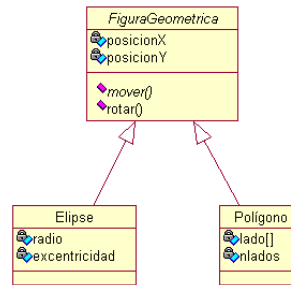


Diagrama de clases – Interfaces

- Conjunto de métodos que:
 - Expresan una determinada funcionalidad (prototipo)
 - Serán implementados por varias clases del modelo
- Relaciones:
 - Entre interfaces y clases: relaciones de realización
 - Es posible definir relaciones de herencia de interfaces

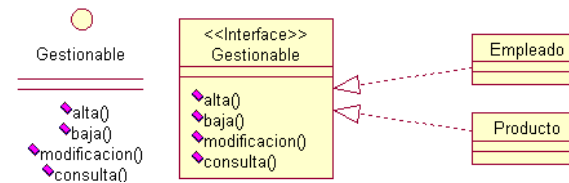
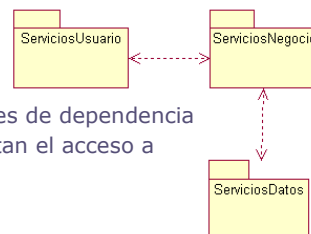


Diagrama de clases – Paquetes

- Agrupación de elementos según un criterio lógico
- Dividen y organizan la información de los modelos
- Pueden contener elementos de cualquier tipo
- Un elemento es exclusivo de un paquete
- Es posible anidar paquetes
- Pueden establecerse relaciones de dependencia entre paquetes que representan el acceso a clases de un paquete a otro



Ejercicio 3 - Diagrama de Clases

- Realizar el diagrama de clases (clases y relaciones) de:
 - Un sitio Web consta de un nombre y un dominio (p.e. google y www.google.com)
 - Los sitios web contienen múltiples documentos los cuales están definidos por una ruta donde se encuentran. Cada documento puede pertenecer a múltiples páginas Web
 - Estos documentos pueden ser bien imágenes o páginas Web. Las imágenes contendrán además de la ruta, sus dimensiones de altura y anchura



Ejercicio 4 - Diagrama de Clases

- Realizar el diagrama de clases de:
 - Un sitio Web tiene tres tipos de usuarios: los administradores del lugar, los usuarios de la empresa, y los clientes que acceden a través de Internet. Cualquier usuario de la aplicación accede a través de la pantalla inicial de acceso que contiene un formulario.
 - El administrador es el encargado de gestionar los distintos usuarios de la aplicación
 - Los usuarios de la empresa podrán consultar el stock de los productos que tienen en la empresa
 - Cualquier usuario después de la pantalla inicial de acceso al sistema podrá realizar pedidos, modificar un pedido realizado o consultar el estado de un pedido



Programación Orientada a Objetos



Ejercicios – Hoja de ejercicios

- Diagramas de Casos de Uso y Clases – Ejercicios 1 a 10



Programación Orientada a Objetos



Diagramas de Interacción

Diagramas de Interacción - Aspectos Generales

- Sirven para modelar los aspectos dinámicos de un sistema - colaboraciones entre objetos
- Elementos:
 - Objetos: instancias de una clase
 - Relaciones: soportan el envío de mensajes
 - Mensajes: enviados entre objetos
- Tipos:
 - Secuencia: Destacan el orden temporal de los mensajes
 - Colaboración: Destacan la organización estructural de los objetos que envían y reciben mensajes
- Son semánticamente equivalentes



Programación Orientada a Objetos



Programación Orientada a Objetos



Diagramas de Interacción - Aspectos Generales

- ¿Cómo construirlos?
 - Escoger la opción por defecto de cada caso de uso
 - Identificar los objetos involucrados en cada caso de uso, a partir de los diagramas de clases
 - Establecer las relaciones y definir la interacción



Diagramas de Secuencia



Diagramas de Interacción – Secuencia - Aspectos Generales

- Resaltan la ordenación temporal de los mensajes
- ¿Para qué sirven?
 - Ayudan a obtener nuevas clases y objetos en un escenario
 - Detectar métodos de las clases al observar las relaciones para llevar a cabo una tarea determinada
 - Especificación de restricciones de interacción en tiempo real
 - Documentación de los casos de uso describiendo las interacciones en términos próximos al usuario
 - Validar el código durante las fases de prueba

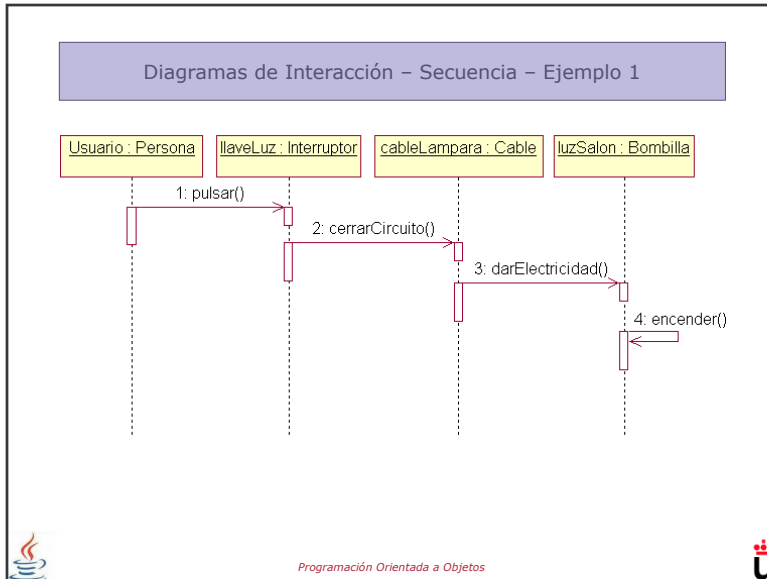


Diagramas de Interacción – Secuencia – Representación Gráfica

- Objetos:
 - Línea de vida de un objeto
 - Foco del sistema
 - Destrucción mediante 'X'
 - Orden de los eventos en la vertical

p1 : Pedido





Diagramas de Interacción – Secuencia – Mensajes

- Mensaje:
 - Origen: Objeto que envía el mensaje
 - Destino: Objeto que ejecuta el mensaje
- Un mensaje es un método del objeto receptor
- Establece una relación cliente/servidor entre el emisor y el receptor
- Un objeto puede enviarse mensajes a sí mismo
- Existe un mensaje de retorno implícito en cada uno de los mensajes que no es necesario especificar
- Pueden incluir:
 - Condiciones: $[x > 10]$: mensaje
 - Iteraciones: $*[i=1..n]$: mensaje

Programación Orientada a Objetos

Diagramas de Interacción – Secuencia – Tipos de Mensajes

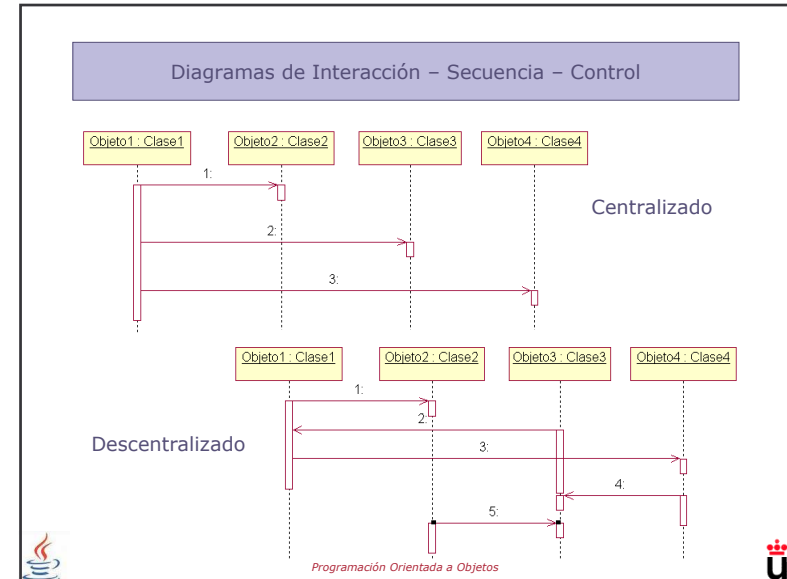
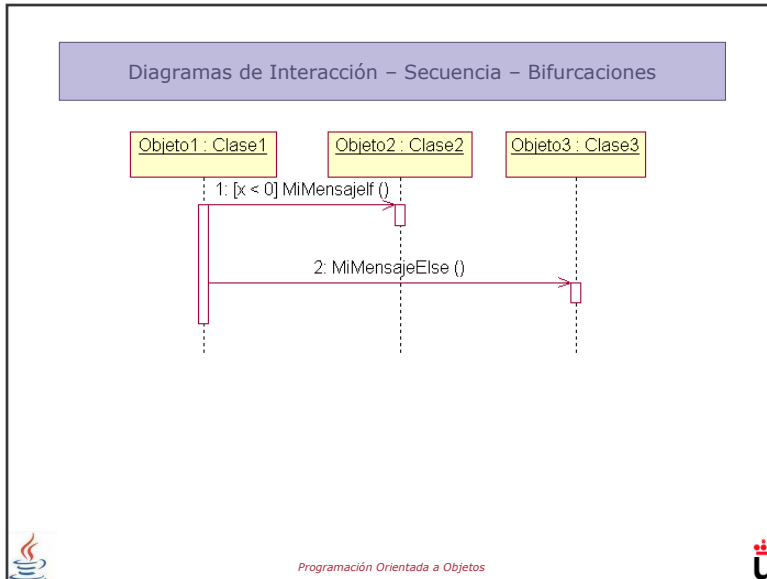
- Simple: Un objeto envía un mensaje a otro
- Síncrono: El cliente envía un mensaje y espera a que el servidor lo acepte
- Abandono: El cliente envía un mensaje al servidor sólo si el servidor está preparado para aceptarlo. El cliente abandona el envío si detecta que el servidor no está preparado

Programación Orientada a Objetos

Diagramas de Interacción – Secuencia – Tipos de Mensajes

- Time-out: El cliente abandona el envío cuando el servidor no puede dar una respuesta en un tiempo previamente establecido
- Asíncrono: El cliente envía un mensaje al servidor y continúa ejecutando su código mientras espera una respuesta del servidor

Programación Orientada a Objetos



Ejercicio 5 – Diagramas de Interacción - Secuencia

- Tenemos un sistema de préstamos de un videoclub que registra todos los préstamos de los socios y el estado de las películas del videoclub. Un socio solicita al encargado del videoclub una determinada película. Si la película se encuentra en el videoclub y el socio no tiene ninguna sanción, se presta la película al socio. Como justificante del préstamo el sistema imprime un recibo para que el encargado se lo entregue al socio. Realizar el diagrama de secuencia del sistema de préstamos del videoclub

Programación Orientada a Objetos

Ejercicio 6 – Diagramas de Interacción - Secuencia

- Realizar el diagrama de secuencia de un sistema de llamadas telefónicas. Cuando una persona quiere llamar a otra, descuelga el teléfono produciéndose en la línea el tono de llamada. A continuación, se teclea cada uno de los dígitos del número de teléfono destino. Una vez introducido el número de teléfono completo, la central telefónica enruta la llamada y crea un objeto de tipo conversación que es el encargado de llamar al receptor de la llamada. Cuando el receptor descuelga su teléfono se produce una conexión desde el objeto conversación a los dos interlocutores

Programación Orientada a Objetos

Diagramas de Colaboración



Programación Orientada a Objetos



Diagramas de Interacción - Colaboración - Aspectos Generales

- Resaltan la organización estructural de los objetos que envían y reciben mensajes
- Una colaboración es un mecanismo compuesto de elementos estructurales y de comportamiento
- Son un mecanismo de organización:
 - Un mismo elemento puede encontrarse en diferentes colaboraciones
- ¿Qué diferencias hay con los diagramas de secuencia?
 - Indican qué objetos actúan como atributos de otros
 - Indican qué objetos son temporales y cuáles no
 - Indican qué relaciones estructurales actúan en una colaboración



Programación Orientada a Objetos



Diagramas de Interacción - Colaboración - Representación Gráfica

- Objetos:
 - Igual que en los diagramas de secuencia

p1 : Producto

- Objeto utiliza múltiples instancias para participar en una colaboración concreta

p2 :
Producto

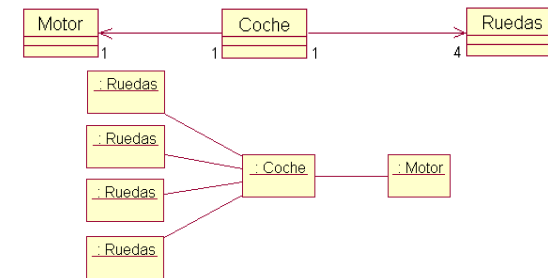


Programación Orientada a Objetos



Diagramas de Interacción - Colaboración - Representación Gráfica

- Enlaces:
 - Relaciones entre objetos que permiten el envío de mensajes entre los mismos
 - Instancias de las relaciones previamente establecidas entre las clases a las que pertenecen dichos objetos



Programación Orientada a Objetos



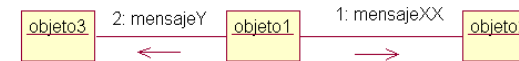
Diagramas de Interacción – Colaboración – Relaciones

- Visibilidad entre objetos:
 - A nivel de campo: El objeto servidor es uno de los atributos del objeto cliente (forma parte de la estructura)
 - A nivel de parámetro: El objeto servidor es un parámetro de alguno de los métodos del objeto cliente
 - Local: El objeto servidor es una variable local a alguno de los métodos del objeto cliente
 - Global: El objeto servidor es global a alguno de los métodos del objeto cliente
- Posibilidad de establecer relaciones reflexivas



Diagramas de Interacción – Colaboración – Mensajes

- Representación de las interacciones entre objetos.
- Secuencia de los mensajes está dada por los números de secuencia
- Posibilidad de incluir condiciones e iteraciones como en los diagramas de secuencia



Ejercicio 7 – Diagramas de Interacción – Colaboración

- Realizar el diagrama de colaboración:
 - María quiere contratar un vuelo desde Madrid a Barcelona operado por Iberia. Para ello, acude a una agencia de viajes donde José le atiende
 - Para la reserva de un vuelo es necesario comprobar si existen todavía plazas, lanzándole una petición a la compañía aérea
 - Si es posible realizar una reserva, se comunica los detalles del vuelo antes de solicitar al sistema un asiento. Cuando se confirma la reserva del asiento, se envía una confirmación. A continuación se procede a efectuar el pago del billete y a emitir el billete del vuelo



Ejercicio 8 – Diagramas de Interacción - Colaboración

- Tenemos un sistema de gestión de pedidos de una empresa. Cada pedido puede tener una o varias líneas de pedido. Por cada producto del pedido, hay que comprobar si hay stock suficiente de un determinado elemento. Si existe stock suficiente se agrega a la línea de pedidos. Si no existe stock suficiente, se crea una petición para pedir más productos al proveedor. Realizar el diagrama de colaboración de este sistema



Ejercicios – Hoja de ejercicios

- Diagramas de Interacción – Ejercicios 1 a 6



Diagrama de Estados



Diagrama de estados – Aspectos Generales

- Modelan clases cuyo comportamiento es dinámico y cuyos objetos atraviesan una serie de estados en respuesta a estímulos recibidos, junto con sus respuestas y acciones
- Elementos principales:
 - Estados: Condición/es de los objetos en un momento determinado
 - Eventos: Estímulo que provoca el cambio de estado de un objeto
 - Acciones: Ocurren cuando un objeto llega a un estado
- Muestran:
 - Los estados que pueden tener los objetos de una clase
 - Qué eventos causan un cambio de estado (transición)



Diagrama de estados – Aspectos Generales

- Se utilizan para:
 - Identificar atributos de objetos
 - Hacer explícito el/los atributo/s que modifican el comportamiento del objeto
 - Muy útiles para modelar sistemas reactivos
- Pueden dibujarse para todas las clases del sistema, aunque no es recomendable

Evento (parámetros) [condiciones] / acciones



Diagrama de estados – Estados

- Periodo de tiempo donde el objeto está esperando algún mensaje u operación
- El estado inicial SIEMPRE debe existir
- El/los estado/s finales pueden NO existir (p.e. sistema que se arranca y no se para nunca)

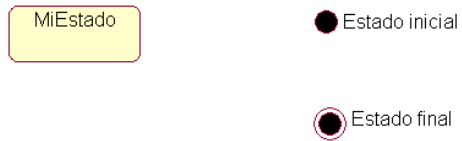
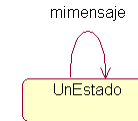


Diagrama de estados – Transiciones y eventos

- Una transición es una conexión unidireccional que une el estado de partida con el estado destino
- Puede haber transiciones al mismo estado de origen



- Evento → Causa del paso de un estado a otro:
 - Condición que toma el valor de verdadero o falso
 - Recepción de una señal de otro objeto del modelo
 - Recepción de un mensaje
 - Paso de un cierto periodo de tiempo



Diagrama de Estados - Ejemplo 1

- Estado de una bombilla de una lámpara

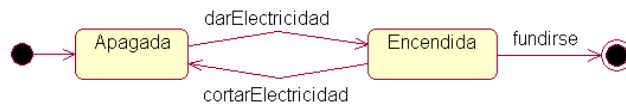


Diagrama de Estados - Ejemplo 2

- Estado de una persona en relación a su vida laboral

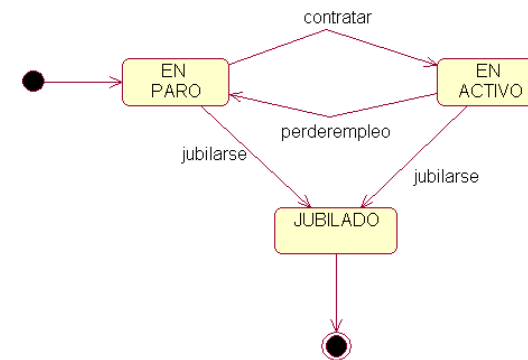


Diagrama de estados – Condiciones

- Se pueden representar condiciones booleanas que indican si se desencadena un evento o no

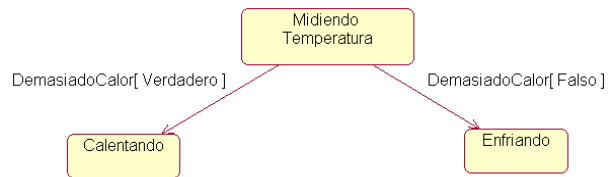
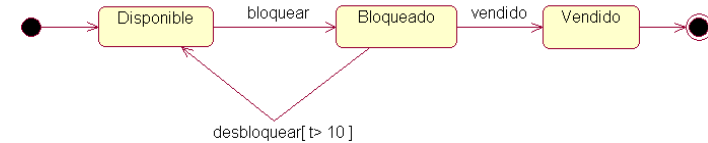


Diagrama de Estados – Ejemplo 3

- Sistema de gestión de vuelos donde:
 - El usuario selecciona un asiento disponible
 - El sistema bloquea el asiento mientras el usuario realiza el pago del billete de avión
 - Si el usuario no realiza el pago en 10 minutos, el asiento volverá a estar disponible para nuevos clientes
 - Una vez pagado el vuelo, se emite un billete electrónico considerando en ese momento el asiento como vendido



Ejercicio 9 – Diagrama de Estados

- El tablero de juego está compuesto de varias celdas o células. Cada célula percibe a sus 8 vecinos. Una célula tiene dos posibles estados: viva o muerta
- Reglas de evolución:
 - Si una célula muerta está rodeada en la actual generación por exactamente tres vecinos vivos, esta célula "nacerá" en la siguiente generación
 - Si una célula viva está rodeada por 2 o 3 células vivas, ésta permanecerá viva en la siguiente generación
 - En el resto de los casos, las células mueren o permanecen muertas

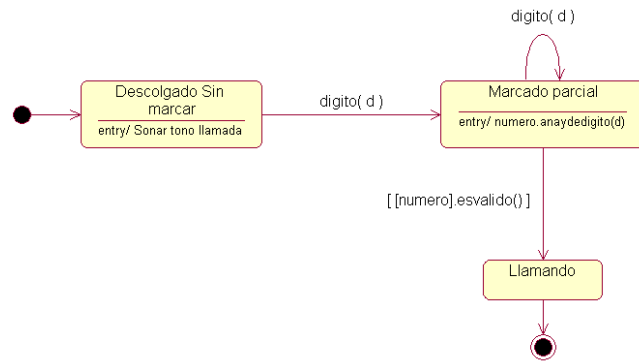


Diagramas de Estados - Acciones

- Las acciones pueden acompañar a los eventos cuando las transiciones son desencadenadas por estos
- Tipos de acciones:
 - Al entrar en un estado se produce una acción (on entry)
 - Al salir de un estado se produce una acción (on exit)
 - Dentro de un estado hay que realizar una acción (do acc)



Diagrama de Estados – Ejemplo 4



Ejercicio 10 – Diagrama de Estados

- Realizar el diagrama de estados de un libro de una biblioteca. Los posibles estados de un libro son "no_reservado" y "reservado"



Ejercicio 11 – Diagrama de Estados

- Cuando una persona quiere comprarse una casa, solicita una hipoteca al banco. Una vez que el banco estudia su caso y acepta la concesión de la hipoteca, el usuario debe pagar a fines de mes una letra
- Mientras que el usuario no abone el total del importe que le queda por pagar, se encontrará en un estado donde su hipoteca se encuentre parcialmente pagada. Cada mes que el cliente pague una letra, se actualizará el importe total que le queda por pagar
- Si el usuario impaga una cuota, su hipoteca pasará a estar impagada, embargándole el banco su casa
- Al cabo de los años, si el usuario consigue pagar totalmente su casa, la hipoteca será cancelada y el piso será de su propiedad



Ejercicio 12 – Diagrama de Estados

- Diagrama de estados de un sistema de validación de tarjetas de un cajero automático.
- El sistema permitirá al usuario tres intentos para introducir su PIN de la tarjeta
- El cajero permitirá tres opciones:
 - Ingreso de dinero
 - Reintegro de dinero
 - Transferencia de dinero



Ejercicios Diagramas de Estado – Hoja de ejercicios

- Diagramas de Estados – Ejercicios 1 a 9



Diagrama de Actividades



Diagrama de Actividades – Aspectos Generales

- Modelan el flujo de las actividades de un sistema
- Elementos principales:
 - Actividades
 - Transiciones
 - Bifurcaciones
- Muestran:
 - Orden en el que se realizan las tareas dentro de un sistema
- Se utilizan para especificar:
 - Un método
 - Un caso de uso
 - Un proceso de negocio



Diagrama de Actividades – Actividades

Preparar
Pedido

ProcesarFactura(fact)
entry/ SacarPrimeraFactura(fact)



Diagrama de Actividades - Transición

- Una transición refleja el paso de un estado a otro
- Cuando una actividad finaliza desencadena una transición y comienza una nueva actividad

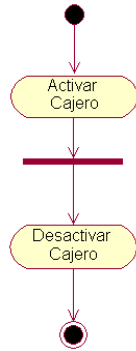
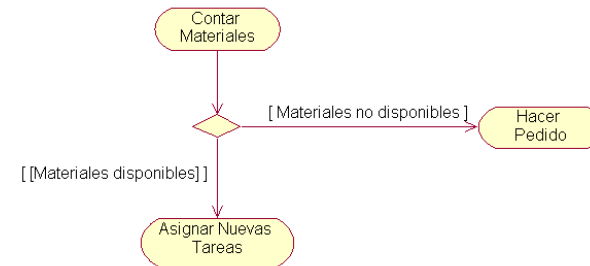


Diagrama de Actividades - Bifurcación

- Expresan distintos caminos dependiendo de una o varias condiciones

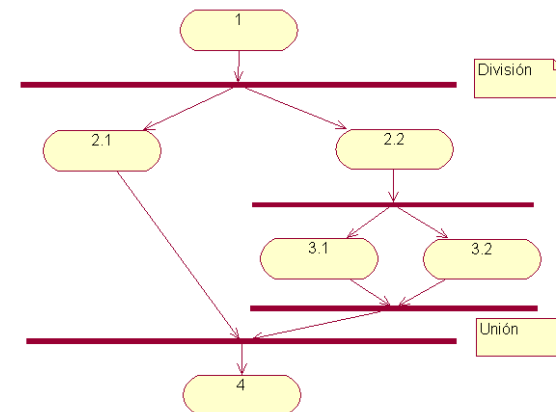


Ejercicio 13 – Diagrama de Actividades

- Realizar el diagrama de actividades de la siguiente situación:
 - Un alumno puede decidir entre estudiar un examen si el examen es difícil, o ver la televisión si el examen es muy fácil y ha hecho las prácticas de la asignatura. Independientemente de lo que elija, al día siguiente se presentará al examen



Diagramas de Actividades – División y Unión

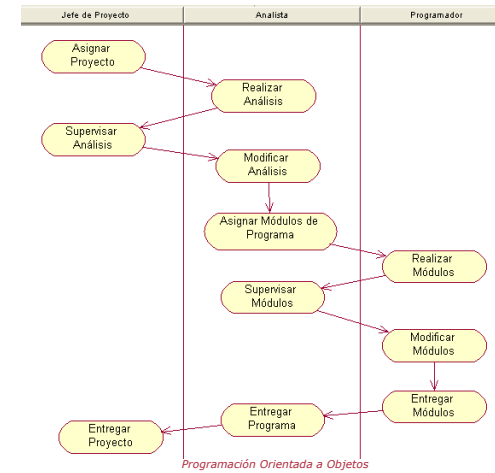


Diagramas de Actividades – División en "calles"

- Se puede representar la división de responsabilidades dentro de un sistema dividiendo por "calles" / columnas.
- Cada calle representa a uno o más objetos y las responsabilidades de ese objeto (actividades de cada objeto)



Diagramas de Actividades – División en "calles" - Ejemplo



Ejercicios – Hoja de ejercicios

- Diagramas de Actividades – Ejercicios 1 a 3



Ejercicio 14 – Diseño OO – Parte I

- Se pretende diseñar una máquina de lavar vehículos. Las acciones a tener en cuenta son las siguientes:
 - El usuario inserta dinero en la máquina para poder lavar su coche
 - El usuario puede pedir opcionalmente abrillantador para el coche
 - La máquina le permite seleccionar un programa de lavado
 - La máquina le permite cancelar un programa de lavado
 - La máquina les proporciona el cambio correspondiente al usuario o le devuelve el importe completo en caso de haya cancelado la operación o que no tenga cambio



Ejercicio 14 – Diseño OO – Parte II

- Representar los siguientes diagramas:
 - Casos de uso del funcionamiento de la máquina de lavado
 - Clases
 - Secuencia que muestre la operación normal de selección de un programa entre el usuario y la máquina
 - Secuencia que muestre la cancelación de un programa
 - Colaboración que sea la traducción de los diagramas de secuencia anteriores
 - Estados que muestre algunos de los estados por los que pasa el sistema cuando un usuario selecciona la opción de "consultar programas de lavado"

