

Introducción a POO

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

Clases y objetos

- Una clase es un tipo al cual pertenecen objetos o instancias de la clase.

Clase **Persona**:
Los objetos de esta
clase tienen **nombre**,
capacidad de **leer**,
dormir y **respirar**



Estas son instancias
de la clase **Persona**.

Clases y objetos

- Una clase es el “plano” que permite “construir” objeto de un determinado tipo.
- Conceptualmente las clases, en la POO, especifican:
 - Atributos: datos que almacena el objeto (y que caracterizan su estado).
 - Comportamiento: funcionalidades o tareas que realizan los objetos.

166

Ejemplos de clases

- Clase Cuenta Corriente:

<u>Atributos:</u>	<u>Comportamiento:</u>
Número cuenta	Girar
Nombre titular	Depositar
Saldo	

- Clase Rectángulo:

<u>Atributos:</u>	<u>Comportamiento:</u>
Alto	Calcular área
Ancho	Calcular perímetro

167

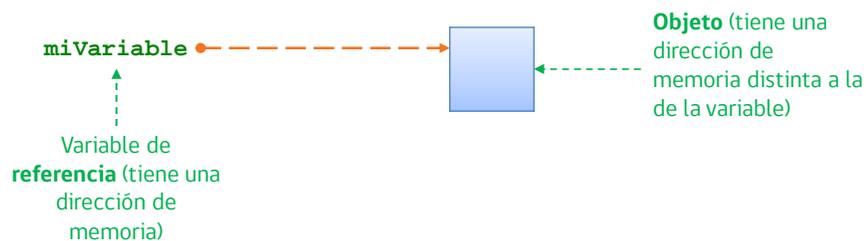
Clases y objetos

- Una clase en el lenguaje Java tiene esencialmente:
 - Variables de instancia: son variables que permiten almacenar los atributos de un objeto.
 - Métodos: procedimientos que implementan el comportamiento de los objetos de la clase.
- Además Java permite definir:
 - Constructores: procedimientos que se ejecutan en el momento de la instanciación del objeto (tienen el mismo nombre de la clase).

168

Clases y objetos

- Los objetos deben instanciarse (mediante el operador new).
- Cada objeto se accede desde una variable de referencia.
- Cada objeto tiene su propia identidad.



169

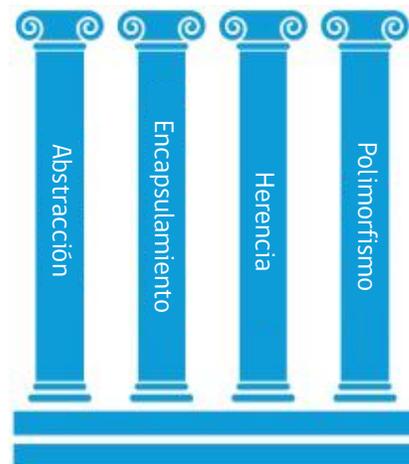
Posibilidades de la POO

- Usar de forma repetida clases previamente implementadas. Ejemplos:
 - `BufferedReader`
 - `String`
- Definir e implementar nuevas clases.

170

Posibilidades de la POO

- La POO (Programación Orientada a Objetos) se basa en cuatro conceptos, que serán estudiados a lo largo de este curso:



171

Abstracción

- Un objeto es capaz de desempeñar una función de forma completamente independiente del contexto en que éste es utilizado.
- En otras palabras, en cualquier ámbito (incluso diferente a aquel en que fue creado), un objeto expone las mismas propiedades y sus operaciones se comportan de la misma forma.



172

Abstracción

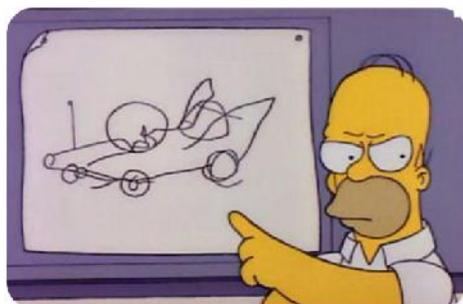
- Por ejemplo, Goofy se comporta como Goofy en la película de Goofy y también en la película La Sirenita.



173

Abstracción

Abstracción



Homero Simpson construyendo el auto de sus sueños



Énfasis en el
¿qué hace? mas
que en el ¿cómo
lo hace?

174

Encapsulamiento

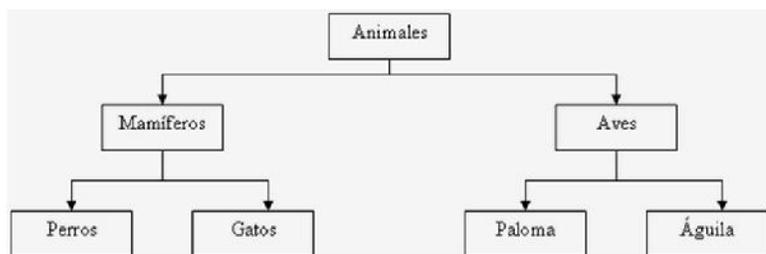
- Un objeto es capaz de responder a peticiones sin exponer la forma en que estas son ejecutadas.
- En otras palabras, la invocación de una operación sobre un objeto gatilla un proceso cuyo efecto es logrado sin dar a conocer sus estructuras internas, ni sus algoritmos.



175

Herencia

- Una clase puede ser generada a partir de otra clase preexistente, heredando las propiedades de esta última.



176

Polimorfismo

- Objetos de un mismo (súper)tipo pueden realizar una misma operación de forma distinta.



177

Análisis y diseño orientado a objetos

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

UML (Unified Modeling Language)

- Lenguaje de propósito general para modelamiento sistemas de software.
- Notaciones principalmente gráficas, basadas en conceptos de Orientación a Objetos (OO)
- Desarrollado por Rational Software en los 90', por Grady Booch, Ivar Jacobson y James Rumbaugh
- UML 1.x (1997) y UML 2.x (2005)



179

UML (Unified Modeling Language)

– ¿En que aspectos aplica el término "unificado"?



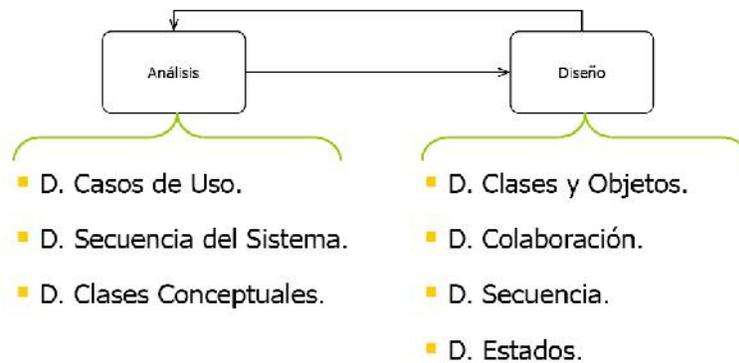
180

UML (Unified Modeling Language)

- El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.
- UML entrega una forma de modelar elementos conceptuales como lo son procesos de negocio y funciones de sistema.
- Además de elementos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables.

181

Diagramas



182

Actividad práctica

- Usando el formato de informes que se encuentra disponible en el aula y en grupos de a lo más 3 personas realice un breve informe donde se señale:
 - Versiones e historia reciente de UML
 - Usos que se le da a UML
 - Diferencias entre modelos de análisis y modelos de diseño
 - Ejemplo de aplicación de modelos de análisis y modelos de diseño.

183

Casos de uso

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

Casos de uso

- Los *Casos de Uso* (Jacobson) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario.
- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación

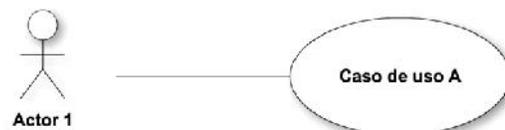
Casos de uso

- Comparación con respecto a los Diagramas de Flujo de Datos del Enfoque Estructurado.
- Los Casos de Uso cubren la carencia existente en métodos previos (OMT, Booch) en cuanto a la determinación de requisitos
- Los Casos de Uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo
- Están basado en el lenguaje natural, es decir, es accesible por los usuarios

186

Casos de uso

- Notación:



187

Actores

- Son los usuarios del sistema.
- En realidad son categorías o tipos de usuario.
- Son entidades externas que interactúan con el sistema para conseguir un objetivo. Son de distintos tipos:
 - **Principales:** personas que usan el sistema
 - **Secundarios:** personas que mantienen o administran el sistema
 - **Material externo:** dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados
 - **Otros sistemas:** sistemas con los que el sistema interactúa.

188

Actores

- La misma persona física puede interpretar varios papeles como actores distintos
- El nombre del actor describe el papel desempeñado
- Los casos de uso son lo que sucede cuando el actor interactúa con el sistema
- El actor usa el sistema para conseguir un objetivo
- Al registrar todas las formas en que el sistema se usa (Casos de Uso) acumulamos todos los objetivos o requerimientos del sistema.

189

Casos de uso

- Un caso de uso es una colección de posibles secuencias de interacción entre el sistema en discusión y sus usuarios relativo a un objetivo determinado.
- La colección de todos los casos de uso define todo el comportamiento del sistema relevante a los actores en cuanto al logro de sus objetivos.
- Cualquier comportamiento irrelevante a los actores no debe ser incluido en los casos de uso.

190

Casos de uso

- Mantienen los requerimientos funcionales en una forma fácil de leer y de trazar.
- Representan los objetivos de la interacción entre el actor y el sistema. Este objetivo representa algo medible y relevante para el actor.
- Registra un conjunto de caminos (escenarios) que atraviesa un actor desde el comienzo del caso hasta la meta (escenarios exitosos).

191

Casos de uso

- Registra un conjunto de caminos (escenarios) que atraviesa un actor desde el comienzo del caso pero que por alguna razón no consigue el objetivo (escenarios de falla).
- Son multinivel porque un caso puede usar o extender la funcionalidad de otro.

192

Restricciones sobre los casos de uso

- No son capaces de especificar la interfaz (intención y no detalles de la acción).
- No son capaces de especificar detalles de implementación, sólo si es de crucial importancia para el logro del objetivo de un actor.
- No son construidos ni representan temporalidad (secuencia).

193

Uso de los casos de uso

- Los casos de uso se usan:
 - Captura de requerimientos de un sistema.
 - Previo y necesario para diseño de un sistema.
 - Para validar el diseño de software contra ellos.
 - Para diseñar las pruebas y aseguramiento de calidad.
 - Diseño inicial de manuales y documentación.

194

Restricciones sobre los casos de uso

- Los Casos de Uso se determinan observando y precisando, actor por actor, las interacciones, los escenarios, **desde el punto de vista del usuario**.
- Un escenario es una instancia de un caso de uso
- Los casos de uso intervienen durante todo el ciclo de vida. El proceso de desarrollo estará dirigido por los casos de uso

195

Actores

– Preguntas que ayudan a encontrar **Actores**:

- ¿Quién está interesado en cierto requisito ?.
- ¿Dónde en la organización se utiliza el sistema ?.
- ¿Quién proveerá, utilizará y eliminará del sistema esta información ?.
- ¿Quién utilizará esta función ?.
- ¿Quién le dará soporte y mantenimiento al sistema ?.
- ¿Usa el sistema un recurso externo ?
- ¿Qué actores necesita este caso de uso ?
- ¿Desempeña un actor varios roles? ¿ Hay actores que desempeñan el mismo rol ?

196

Casos de uso

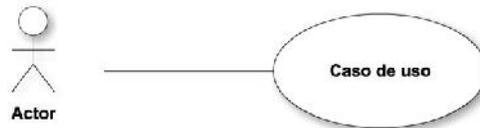
– Preguntas que ayudan a encontrar **Casos de Uso**:

- ¿Cuáles son las tareas de este actor ?.
- ¿El actor, creará, guardará, cambiará, eliminará o leerá información en el sistema ?.
- ¿Qué caso de uso, creará, guardará, cambiará, eliminará o leerá esta información ?.
- ¿Necesitará el actor informar al sistema sobre cambios externos imprevistos ?.
- ¿Es necesario que el actor esté informado sobre ciertas ocurrencias del sistema ?.
- ¿Proporciona el sistema un correcto funcionamiento para el negocio ?.

197

Relaciones

- UML define cuatro tipos de relación en los Diagramas de Casos de Uso:
 - Comunicación:



198

Relaciones

- UML define cuatro tipos de relación en los Diagramas de Casos de Uso:
 - Inclusión : una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino:



- «include» en UML 2.0 reemplazó al denominado «uses» de UML 1.0.

199

Relaciones

- UML define cuatro tipos de relación en los Diagramas de Casos de Uso:
 - Extensión : el caso de uso origen extiende el comportamiento del caso de uso destino.



200

Relaciones

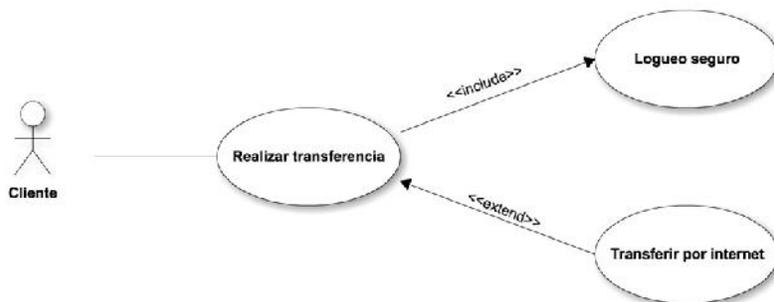
- UML define cuatro tipos de relación en los Diagramas de Casos de Uso:
 - Herencia : el Caso de Uso origen hereda la especificación del Caso de Uso destino y posiblemente la modifica y/o amplía



201

Relaciones

– Ejemplo



202

Casos de uso

– Ejemplos

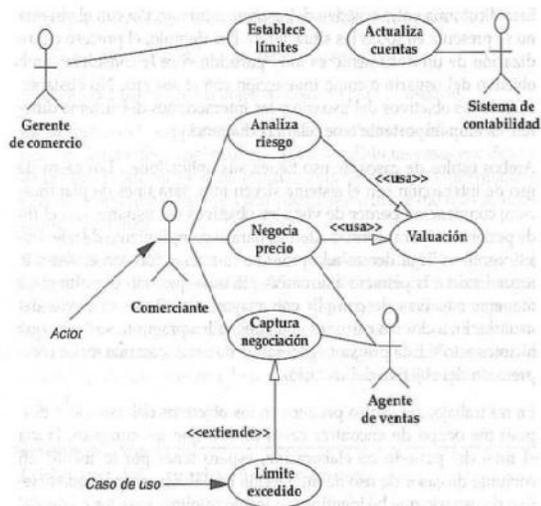


Figura 3-1: Diagrama de casos de uso

203

Casos de uso narrativo



- Los casos de uso están documentados en:
 - Una breve descripción; el propósito del caso de uso en unas pocas líneas.
 - Flujo de eventos detallados; descripción del flujo de eventos primario y alternativas que ocurren cuando el caso de uso es iniciado.
 - La documentación debe leerse como un diálogo entre el actor y el caso de uso.
 - Ambos documentos están escritos en términos que el cliente entiende.

204

Casos de uso narrativo



- Los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos o especificaciones funcionales, sino que ejemplifican e incluyen los requerimientos en las historias que narran.
- El siguiente ejemplo corresponde a un caso de uso de alto nivel, que describe clara y concisamente el proceso de comprar productos en una tienda cuando se emplea una caja registradora en el punto de venta:

205

Casos de uso narrativo

Caso de Uso	Comprar Productos
Actores	Cliente (iniciador), Cajero
Tipo	Principal
Descripción	Un Cliente llega a una caja con productos que desea comprar. El Cajero registra los productos y obtiene el pago. Al terminar la transacción, el Cliente se marcha con los productos.

206

Casos de uso narrativo

- La explicación del formato es:
 - Caso de Uso: nombre del caso de uso.
 - Actores: lista de actores en la que se indica quien inicia el caso de uso.
 - Tipo: que puede ser Primario, Secundario u Opcional, entre otros.
 - Descripción: repetición del caso de uso de alto nivel o alguna síntesis similar.

Caso de Uso	Comprar Productos
Actores	Cliente (iniciador), Cajero
Tipo	Principal
Descripción	Un Cliente llega a una caja con productos que desea comprar. El Cajero registra los productos y obtiene el pago. Al terminar la transacción, el Cliente se marcha con los productos.

207

Casos de uso narrativo expandido

- La estructura del formato expandido agrega a la de alto nivel lo siguiente:
 - Propósito: intención del caso de uso.
 - Referencias Cruzadas: caso de uso y funciones relacionadas con el sistema.
 - Curso Normal de los Eventos: es la parte medular del formato expandido; describe detalles de la conversión interactiva entre los actores y el sistema. Un aspecto esencial de la sección es explicar la secuencia más común de eventos: la historia normal de las actividades y el termino exitoso de un proceso. No incluye situaciones alternativas.
 - Cursos Alternativos: describe importantes opciones o excepciones que pueden presentarse en relación al curso normal. Si éstas son complejas se pueden expandir y convertir en nuevos casos de uso.

208

Casos de uso narrativo expandido

- El siguiente ejemplo corresponde al mismo caso de uso anterior; describe clara concisamente el proceso de comprar productos en una tienda cuando se emplea una caja registradora en el punto de venta:

Caso de Uso	Comprar Productos
Actores	Cliente (iniciador), Cajero
Propósito	Capturar una venta y su pago
Tipo	Principal y esencial
Descripción	Un Cliente llega a una caja con productos que desea comprar. El Cajero registra los productos y obtiene el pago. Al terminar la transacción, el Cliente se marcha con los productos.
Referencias Cruzadas	Casos de Uso: a. Cajero debe haber terminado el caso de uso llamado Registrar
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
1.- Este caso de uso comienza cuando un Cliente llega a la caja con productos que desea comprar.	
2.- El Cajero registra los productos. Si hay más de un producto, también puede introducir la cantidad.	3.- Determina el precio del producto y agrega la información sobre el a la actual transacción de venta.
4.- Al terminar el registro de los productos, el Cajero indica al sistema que terminó dicho proceso.	5.- Calcula y presenta el total de la venta.
6.- El Cajero le indica el tca. al Cliente.	
7.- El Cliente escoge la forma de pago: a) Si paga en efectivo, ver la sección Pago en Efectivo b) Si paga con tarjeta con crédito, ver la sección Pago con Tarjeta de Crédito c) Si paga con cheque, ver la sección Pago con Cheque	8.- Registra la venta terminada.
	9.- Actualiza los niveles de inventario.
	10.- Genera un recibo.
11.- El Cajero entrega el recibo al cliente.	
12.- El Cliente se marcha con los productos comprados.	

209

Casos de uso narrativo expandido

- El siguiente ejemplo corresponde al mismo caso de uso anterior; describe clara concisamente el proceso de comprar productos en una tienda cuando se emplea una caja registradora en el punto de venta:

(continuación)

Sección: Pago en Efectivo	
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
1.- El Cliente da un pago en efectivo, posiblemente mayor que el total de la venta.	
2.- El Cajero registra el efectivo recibido.	2.- Entrega la diferencia al Cliente.
4.- El cajero guarda el efectivo recibido y saca la diferencia. Luego, le entrega el vuelto al Cliente.	
Cursos Alternativos - Sección Pago en Efectivo	
- Línea 1: el Cliente no tiene suficiente efectivo. Puede cancelar e iniciar otro método de pago.	
- Línea 4: el Cajero no tiene suficiente efectivo para pagar la diferencia. El Cajero pide más efectivo al supervisor o le pide al Cliente otro billete de menor valor u otra forma de pago.	

210

Conclusiones

- Un caso de uso debe ser simple, inteligible, claro y conciso
- Generalmente hay pocos actores asociados a cada Caso de Uso
- Preguntas clave:
 - ¿cuáles son las tareas del actor?
 - ¿qué información crea, guarda, modifica, destruye o lee el actor?
 - ¿debe el actor notificar al sistema los cambios externos?
 - ¿debe el sistema informar al actor de los cambios internos?

211

Conclusiones

- La descripción del Caso de Uso comprende:
 - el inicio: cuándo y qué actor lo produce?
 - el fin: cuándo se produce y qué valor devuelve?
 - la interacción actor-caso de uso: qué mensajes intercambian ambos?
 - objetivo del caso de uso: ¿qué lleva a cabo o intenta?
 - cronología y origen de las interacciones
 - repeticiones de comportamiento: ¿qué operaciones son iteradas?
 - situaciones opcionales: ¿qué ejecuciones alternativas se presentan en el caso de uso?

212

Actividad

- Investigar sobre “Uso de diagramas de caso de uso para representar seguridad en sistemas de información”.
- Grupo de hasta 5 personas.
- Entrega vía aula virtual
- Plazo 1 semana

213

Modelo de clases y objetos

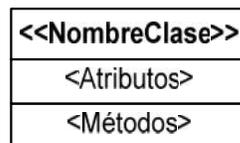
ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

Diagrama de clases y objetos

- El Diagrama de Clases es el diagrama principal para el análisis y diseño del sistema
- Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia
- La definición de clase incluye definiciones para atributos y operaciones
- El modelo de casos de uso debería aportar información para establecer las clases, objetos, atributos y operaciones

Diagrama de clases y objetos

- Elementos de UML de los diagramas de clases
 - Clases, su estructura y su comportamiento
 - Relaciones de asociación, agregación, dependencia, y herencia
 - Indicadores de multiplicidad y navegación
 - Nombres del papel que juega cada clase en una relación



216

Diagrama de clases y objetos

- Atributos: Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:
 - public (+): Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
 - private (-): Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).
 - protected (#): Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven.

217

Diagrama de clases y objetos

– Atributos:

Profesor
+Nombre : char
#Edad : int
-Sueldo : double

218

Diagrama de clases y objetos

– Métodos: Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:

- public (+): Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- private (-): Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).
- protected (#): Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven (ver herencia).

219

Diagrama de clases y objetos

– Métodos:

Profesor
+Nombre : char
#Edad : int
-Sueldo : double
+Evaluar() : decimal
#PrepararClase()
-Eximir() : bool

220

Diagrama de clases y objetos

– Relaciones entre las Clases:

- Asociación
- Agregación (la veremos posteriormente)
- Dependencia

221

Diagrama de clases y objetos

– Relaciones entre las Clases: Asociación

- Una asociación es una conexión bi-direccional entre clases
- Una asociación se representa como una línea que conecta las clases relacionadas

– Relaciones entre las Clases: Agregación

- Una agregación es una relación más fuerte: entre el todo y las partes que lo componen
- Una agregación se representa como una línea que conecta las clases relacionadas con un rombo junto a la clase que representa el todo

222

Diagrama de clases y objetos

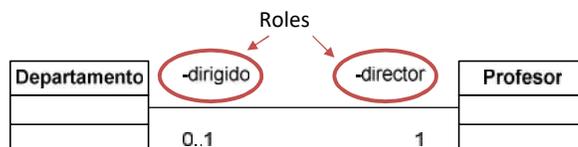
– Relaciones entre las Clases: Dependencia

- Una relación de dependencia es una relación más débil: la que hay entre un cliente y un proveedor cuando el cliente no tiene conocimiento semántico del proveedor
- Una dependencia se representa como una línea punteada que apunta del cliente al proveedor

223

Diagrama de clases y objetos

– Relaciones entre las Clases: Asociación



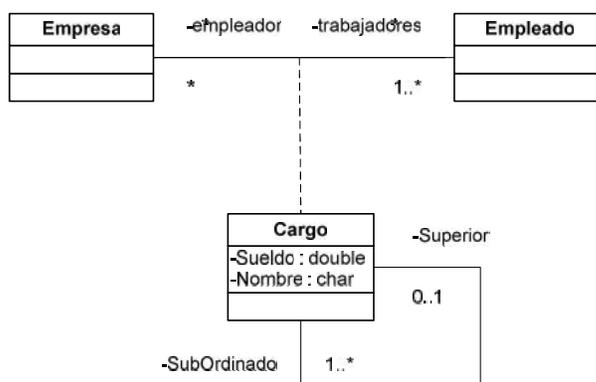
– Se lee como

- Un Profesor es director de 0 ó 1 Departamento
- Un Departamento es dirigido por un Director

224

Diagrama de clases y objetos

– Relaciones entre las Clases: Dependencia



225

Diagrama de clases y objetos

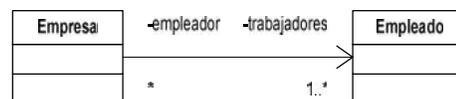
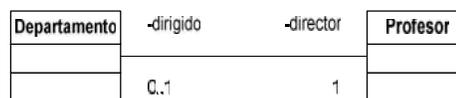
– Multiplicidad y navegación:

- La multiplicidad define cómo varios objetos participan en las relaciones
- La multiplicidad de una relación es el número de ejemplares de una clase relacionados con UNA instancia de otra clase
- Para cada asociación y agregación, hay dos decisiones de multiplicidad que tomar: una para cada extremo de la relación
- Aunque las asociaciones y las agregaciones son bi-direccionales por defecto, a veces es deseable restringir la navegabilidad de la relación a una sola dirección
- Si la navegabilidad está restringida, se añade una punta de flecha para indicar la dirección de navegación

226

Diagrama de clases y objetos

– Multiplicidad y navegación:



227

Diagrama de clases y objetos

– Multiplicidades:

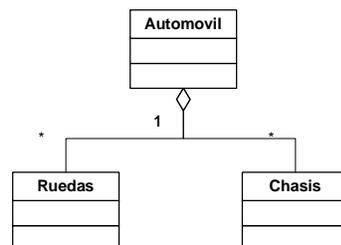
- 0..1
- 1
- 0..*
- *
- 1..*
- 1..6
- 1..3,7..10,15,19..*

228

Diagrama de clases y objetos

– Relaciones entre las Clases: Agregación (Sin composición)

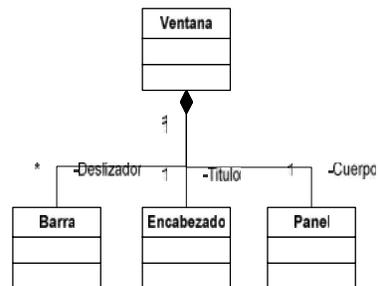
- Connotación semántica de posesión, contención o composición (vista posteriormente)
- Usado en modelos de clases para mostrar como los elementos más complejos se construyen desde una colección de simples elementos (ej. un automóvil de ruedas, neumáticos, motor, etc.)



229

Diagrama de clases y objetos

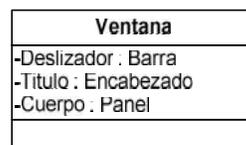
- Relaciones entre las Clases: Agregación → Composición
 - Agregación con connotación de posesión
 - Coincidencia en tiempo de vida entre el todo y las partes



230

Diagrama de clases y objetos

- Relaciones entre las Clases: Agregación → Representación



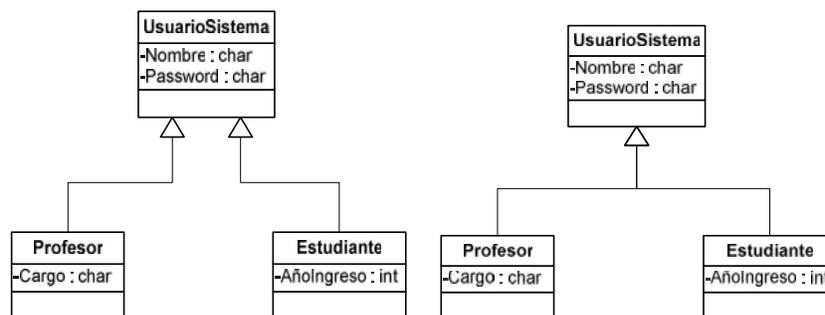
231

Herencia

- Herencia es la relación entre una superclase y sus subclases
- Hay dos maneras de encontrar la herencia:
 - Generalización
 - Especialización
- Los atributos, métodos, y/o relaciones comunes deben representarse al nivel aplicable más alto de la jerarquía

232

Herencia



233

Diagrama de objetos

- Capturan la vista de diseño estática de un sistema desde el punto de vista de los objetos
- En los diagramas de objetos
 - No aparece la herencia
 - No aparecen las dependencias

234

Diagrama de objetos

- Representación de un objeto:

<u>juan : Profesor</u>
Nombre : char = Juan
Edad : int = 45
Sueldo : double = 150.000

235

Diagrama de objetos

- Un diagrama de Objetos está relacionado de cerca con un diagrama de Clases, con la diferencia de que éste describe las instancias de los objetos de clases en un punto en el tiempo.
- Los diagramas de Objetos no presentan arquitecturas que varíen de sus correspondientes diagramas de Clases, pero reflejan la multiplicidad y los roles a los que las clases instanciadas podrían servir.

236

Diagrama de objetos

- Ellos son muy útiles en la comprensión de diagramas de Clases complejos, al crear diferentes casos en los que se aplican las relaciones y las clases
- Un diagrama de Objetos puede ser también un tipo de diagrama de Comunicaciones, el cual modela también las conexiones entre pares de objetos y además las secuencias de eventos a lo largo de cada camino.

237

Tarea

- Leer y estudiar el Capítulo: Diagrama de Clases, Conceptos Avanzados.
 - Publicado en el aula

238

Actividad

- Investigar sobre:
 - Utilización y forma de representar clases abstractas y clases parametrizadas.
 - Dar un ejemplo real y aplicado.
 - Formato IEEE Xplore

239

Diagrama de estados

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

Diagrama de estados

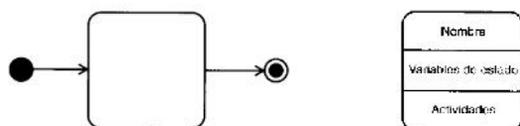
- Los diagrama de estados son la forma que existe en UML para caracterizar un cambio en un sistema.
- Los objetos que componen un sistema modifican su estado como respuesta a sucesos y el tiempo.
- Algunos ejemplos son:
 - Cuando se acciona un interruptor y la luz cambia de apagado a encendido.
 - Cuando se cambia de canal cuando se presiona el control remoto.
 - Luego de un tiempo determinado la lavadora se auto apaga.

241

Diagrama de estados

– Notación:

- Estado: rectángulo de vértices redondeados.
- Transición: línea continua y punta de flecha.
- Punto inicial: círculo relleno.
- Punto final: círculo tipo diana.



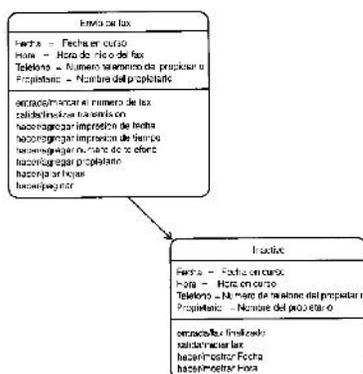
242

Diagrama de estados

– Ejemplo

FIGURA 8.3

La máquina de fax es un buen ejemplo de un estado con variables y actividades.



243

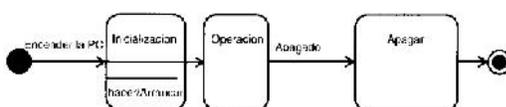
Diagrama de estados

– Sucesos y acciones:

- Se pueden representar transiciones que representan **sucesos**.
- **Acciones** de computo.
- Ambos conllevan a la modificación del estado.

FIGURA 8.4

Los estados y transiciones de una interfaz gráfica del usuario incluyen el desencadenamiento de eventos, acciones y transiciones no desencadenadas.



244

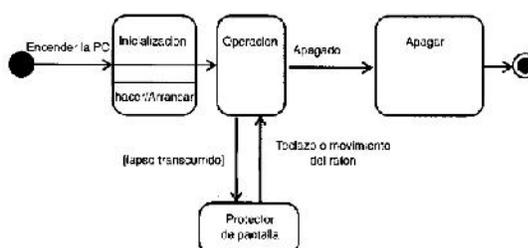
Diagrama de estados

– Condiciones de seguridad:

- Se pueden añadir estados que provean seguridad a la solución diseñada.
- Por ejemplo:

FIGURA 8.5

El diagrama de estados para la GUI con el estado Protector de pantalla y la condición de seguridad.



245

Actividad

- En base a los diagramas de estados investigar y dar un ejemplo sobre:
 - Subestados
 - Estados históricos
 - Mensajes y señales

Diagrama de secuencia

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

Diagrama de secuencia

- *"Un diagrama que representa una interacción poniendo el foco en la secuencia de los mensajes que se intercambian, junto con sus correspondientes ocurrencias de eventos en las líneas de vida."*
- *"A diferencia de un diagrama de comunicación, un diagrama de secuencias incluye secuencias de tiempo pero no incluye relaciones de los objetos. Un diagrama de secuencias puede existir en una forma genérica (describe todos los escenarios posibles) y en una forma de instancia (describe un escenario actual). Los diagramas de secuencias y los diagramas de comunicación expresan información similar, pero la muestran de diferentes formas."*

Diagrama de secuencia

- Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal.
- Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre ellos para ejecutar la funcionalidad descrita por el escenario.
- Los diagramas de secuencia, se utilizan con frecuencia para validar los casos de uso:

249

Diagrama de secuencia

- Los diagramas de secuencia, se utilizan con frecuencia para validar los casos de uso:
 - Documentan el diseño desde el punto de vista de los casos de uso. Observando qué mensajes se envían a los objetos, componentes o casos de uso y viendo a grosso modo cuanto tiempo consume el método invocado,
 - Ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos.
- A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del diagrama de clases.

250

Elementos

- Actor: Es el equivalente al actor que identificamos en el caso de uso que se quiere representar con el diagrama de secuencia
 - Gráficamente igual a los casos de uso



251

Elementos

- Línea de vida: una línea de vida es un participante individual en una interacción
 - No tienen multiplicidad
- Las líneas de vida están disponibles en los diagramas de Secuencias y de Tiempos, y aunque la representación difiere entre los dos, el significado de la línea de vida es el mismo.
- Definición según OMG:

252

Elementos

– Definición según OMG:

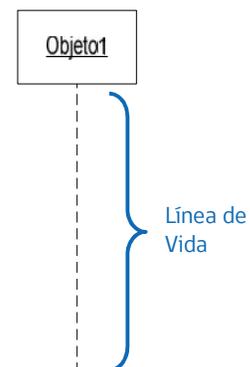
- Una línea de vida representa un participante individual en la Interacción. Mientras las Partes y Características de Estructuras pueden tener multiplicidades mayores a 1, las Líneas de Vida representan solamente una entidad interactuante. La línea de vida es una especialización de un Elemento nombrado.

253

Elementos

– Línea de vida, representación gráfica

- No tiene sentido representarla sino está asociada a la representación de un Objeto.



254

Elementos

- Mensajes: Indican un flujo de información o transición del control entre elementos.
- Se pueden utilizar por todos los diagramas de interacción.
- Corresponden en el modelo de software a operaciones y comportamientos de clases. Son semánticamente similares a los mensajes pasados entre elementos en un diagrama de Comunicación.

255

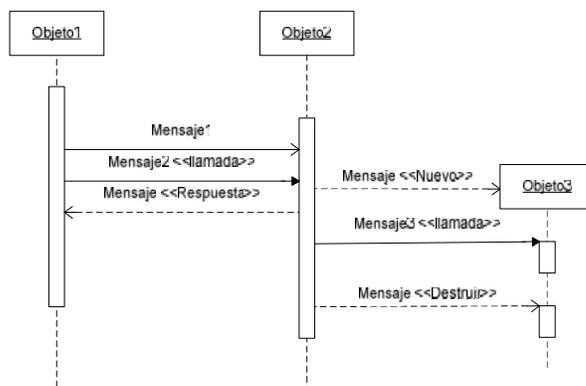
Elementos

- Condiciones de los mensajes:
 - Indican qué debe ser verdadero para que el mensaje se envíe.
- Valor de retorno
- Sincronización
- Frecuencia
- Creación (Ciclo de Vida)
 - Se utiliza nuevo o crear nuevo para la creación de elementos
 - Eliminar o destruir para la destrucción de elementos

256

Elementos

– Ejemplo de mensajes:

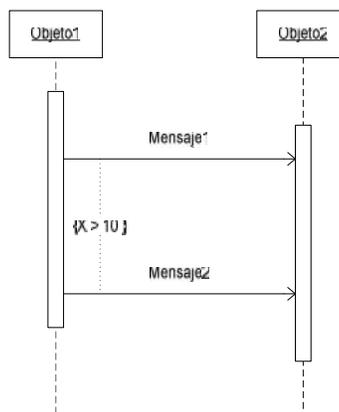


257

Elementos

– Flujos alternativos:

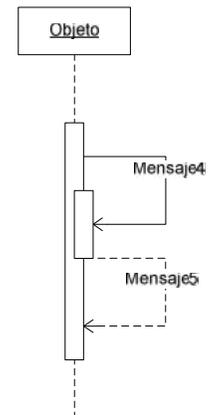
- Los diagramas de secuencia permiten mostrar flujos alternativos en base a una condición lógica.



258

Elementos

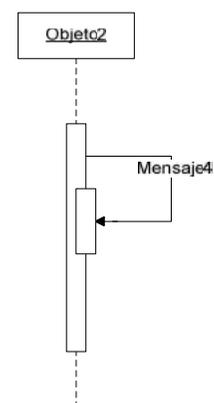
- Automensajes: Un auto mensaje refleja un nuevo proceso o método que se invoca dentro de la operación de la misma línea de vida. Es una especificación de un mensaje.
 - Auto Mensaje como retorno
 - Es posible describir un retorno desde la llamada de un auto mensaje.



259

Elementos

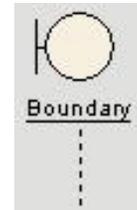
- Mensajes recurrentes: indican recursividad. No es lo mismo que un automensaje



260

Elementos

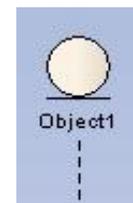
- Límites: Un límite es una clase estereotipada que modela algunos límites del sistema
 - Normalmente una pantalla de interfaz de usuario.
- Se usa en la fase conceptual para capturar usuarios interactuando con el sistema a un nivel de pantalla (o algún otro tipo de interfaz de límite).



261

Elementos

- Entidad: Una entidad es un almacén o mecanismo de persistencia que captura la información o el conocimiento en un sistema.

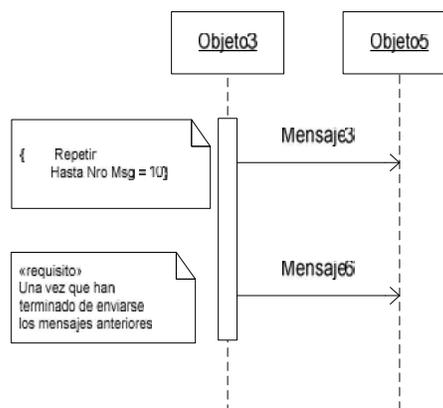


262

Elementos

– Notas y Restricciones

- Sirven para hacer una especificación en caso de tener estructuras repetitivas o bien para aclarar ciertas interacciones
- Restricciones especiales a clases o mensajes también pueden ser incluidas



263

Modelado de flujos por orden de tiempo

- Establecer el contexto para la interacción (sistema, subsistema, operación, clase o escenario de un caso de uso o colaboración).
- Establecer el escenario para la interacción identificando cuáles objetos juegan un rol en la interacción.
 - Poniéndolos en el diagrama de secuencia de izquierda a derecha, colocando los objetos más importantes y a sus objetos vecinos a la derecha.

264

Actividad

- Realizar un diagrama de secuencia para modelar “La compra de pasajes de bus”.
- Grupo de hasta 3 personas.

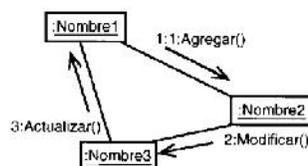
Diagrama de colaboración

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

Que es un diagrama de colaboración

- Es la forma de mostrar los objetos como tales y sus relaciones entre si.
- Es una extensión de un diagrama de objetos.
- Muestra los mensajes que se envían entre objetos.

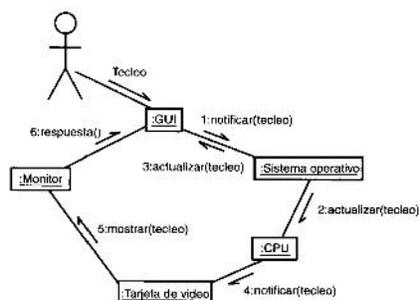
FIGURA 10.1
*Simbología del
diagrama de
colaboraciones.*



Actores

- En este modelo también se puede incluir el uso y participación de actores.
- Es importante señalar que el actor no es parte formal de la simbología de este diagrama.

FIGURA 10.2
Un diagrama de colaboraciones para el ejemplo de la GUI.

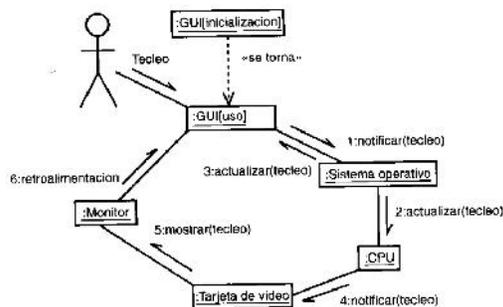


270

Estados en los diagramas de colaboración

- Además, se pueden representar estados como un objeto más.

FIGURA 10.3
Un diagrama de colaboraciones puede incorporar cambios de estado.



271

Ejemplo

FIGURA 10.4

El diagrama de colaboraciones para el mejor caso de "Comprar gaseosa".

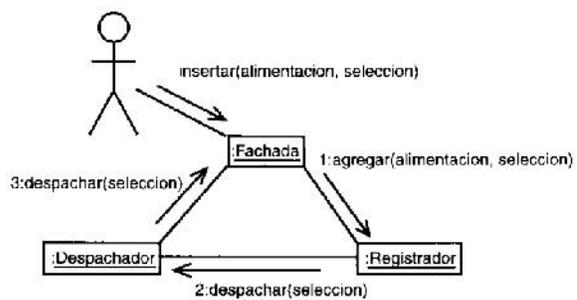


Diagrama de actividad

ICI3242- Modelamiento de sistemas de software
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

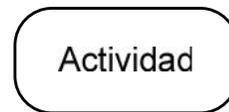
Diagrama de actividad

- El diagrama de actividad se usa para mostrar la secuencia de actividades.
- Muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad.
 - Estos también pueden usarse para detallar situaciones donde pueden ocurrir procesos en paralelo.
- Los Diagramas de Actividades son útiles para el Modelado de Negocios donde se usan para detallar el proceso involucrado en las actividades de negocio.

274

Diagrama de actividad

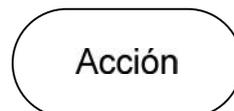
- Una actividad es la especificación de una secuencia parametrizada de comportamiento.
- Incluye todas las acciones, flujos de control y otros elementos que constituyen la actividad
 - Enterprise Architect le llama actividad.
 - Microsoft Visio le llama Estado



275

Diagrama de actividad

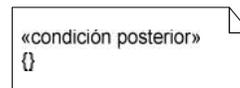
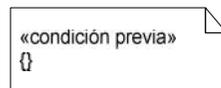
- Acción: Una acción representa un solo paso dentro de una actividad.
 - Enterprise Architect le llama Acción
 - Microsoft Visio le llama Estado de Acción



276

Diagrama de actividad

- Restricción: Las restricciones se pueden adjuntar a una acción.
- Estereotipos:
 - Precondiciones
 - Postcondiciones
- Puede Estar Escrito en Código, Pseudocódigo, OCL o Texto



277

Diagrama de actividad

- Flujo de Control:
 - Un flujo de control muestra el flujo de control de una acción a otra
 - Gráficamente:

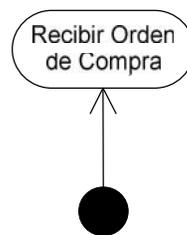


278

Diagrama de actividad

– Nodo Inicial:

- Marca el Inicio de la Actividad.
- No se rotula ni estereotipa, ya que no implica una acción de ningún tipo
- Gráficamente



279

Diagrama de actividad

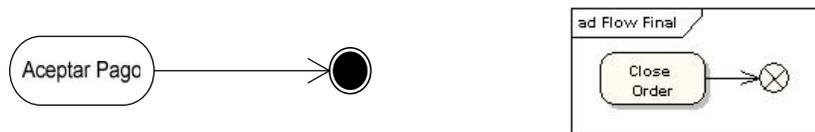
– Nodo final: Hay dos tipos de nodos finales:

- nodos finales de actividad y
- nodos finales de flujo.
- El nodo final de actividad indica el fin de TODA la actividad. Una vez allí NO se realizan más acciones.
- El nodo final de Flujo indica el fin de un Flujo determinado, sin embargo flujos alternativos pueden seguir ejecutándose.
 - Microsoft Visio NO implementa este tipo de flujos

280

Diagrama de actividad

- Nodo final:

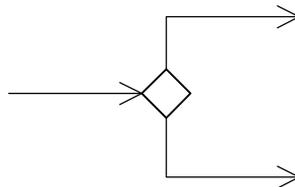


281

Diagrama de actividad

- Nodos de Decisión:

- Los flujos de control que provienen de un nodo de decisión tendrán condiciones que permitirán el control para fluir si la condición se cumple.
- Permite establecer caminos alternativos NO simultáneos



282

Diagrama de actividad

– Nodos de Bifurcación y Unión:

- Estos indican el comienzo y final de hilos actuales de control.
- Permite establecer caminos alternativos simultáneos

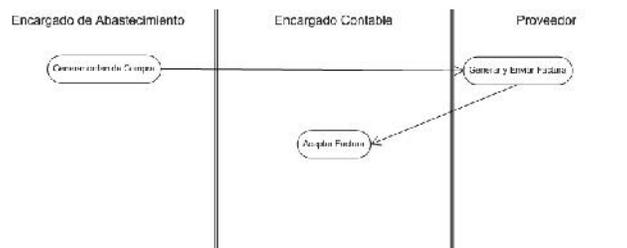


283

Diagrama de actividad

– Partición:

- Una partición de una actividad se muestra como calles horizontales o verticales.
- Las particiones se usan para separar acciones dentro de una actividad, especificando quienes las realizan.



284

Actividad

- Investigar sobre los siguientes modelos, entregando un resumen de su uso y un ejemplo elaborado por usted mismo:
 - Diagrama de objetos
 - Diagrama de componentes