



INGENIERÍA DEL SOFTWARE I

Práctica 5, Sesión 1

Modelo de Análisis

Univ. Cantabria – Fac. de Ciencias

María Sierra



Diagramas de Clases

- **Modelado:** Identificación de las “cosas” importantes desde una visión particular.
 - Constituyen el vocabulario del sistema que se modela
 - Cada una de ellas tiene ciertas propiedades y un comportamiento.
- Existen **relaciones** entre esas cosas. En UML estas “cosas” se modelan como clases → Clase: Abstracción
- Generalmente el vocabulario del sistema lo componen las clases → **Sencillo en OO al unir el mundo real con el mundo de la solución tecnológica.** Para modelar el vocabulario de un sistema:
 - Hay que identificar aquellas cosas que utilizan los usuarios/programadores para describir el problema o la solución.
 - Identificar las responsabilidades.
 - Definir atributos y métodos necesarios para cumplir con las responsabilidades.
 - En el **Mundo Real** hay clases: sencillo en OO al unir el mundo real con el mundo de la solución tecnológica.
- **Organización en Paquetes**



Diagramas de Clases: Elementos

- **Nombre:**

- Las distingue de las demás.
- Ha de ser simple.
- Pueden mostrar el "camino"

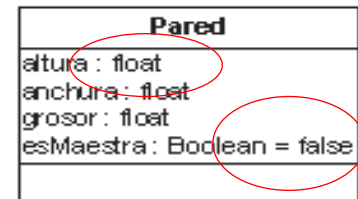


- **Atributo:**

- Propiedad que es compartida por todos los objetos de la clase.

- **Método:**

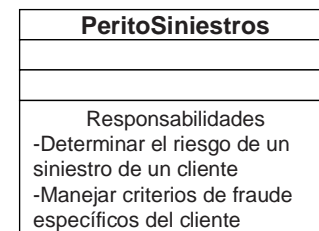
- Es la implementación de un servicio que pueda ser requerido.
- Es una abstracción de algo que puede hacer el objeto.



Diagramas de Clases: Elementos

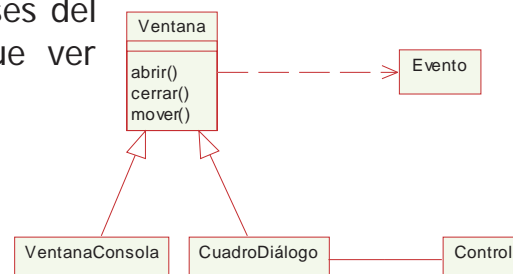
- **Responsabilidades:**

- Contrato u obligación de una clase, debe ser tenido en cuenta al modelar clases.
 - Ejemplo: Una clase pared es responsable de saber su altura, anchura y grosor.
- Al modelar clases se debe tener en cuenta las responsabilidades.



- **Relaciones:**

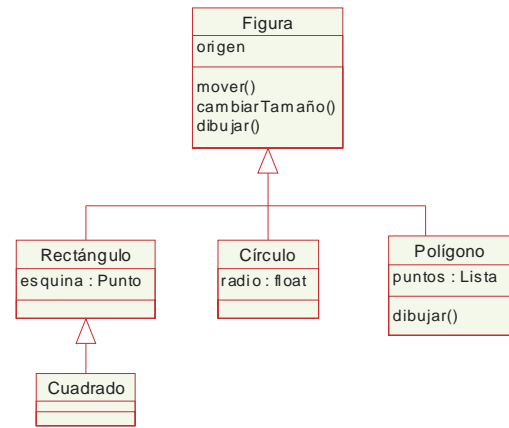
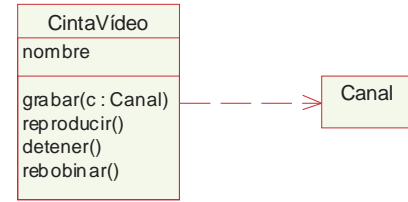
- Forma en que cada una de las clases del vocabulario del sistema tienen que ver unas con otras
 - Dependencias
 - Generalización
 - Asociación





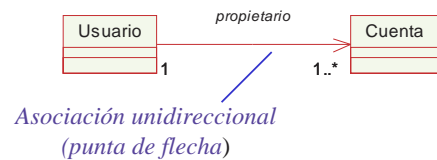
Diagramas de Clases: Relaciones

- **Dependencias:** Relación de uso que indica que un cambio en una clase puede afectar a otra.
 - Especifica que un elemento utiliza a otro
 - También entre notas y paquetes.
- **Generalización:** Relación entre un elemento general (padre/superclase) y un caso más específico (hijo/subclase).
 - Relación "es-un".
 - El hijo puede sustituir al padre.
 - El hijo hereda atributos y métodos del padre y añade algunos más.
 - Un método del hijo con el mismo nombre que el de padre lo redefine (polimorfismo).

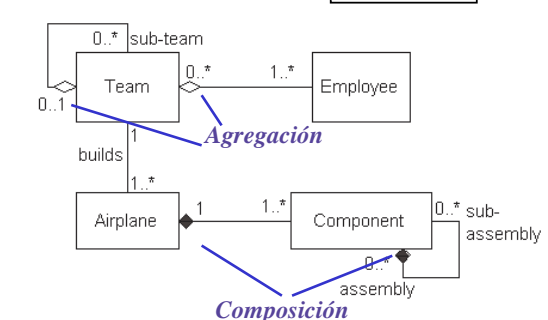
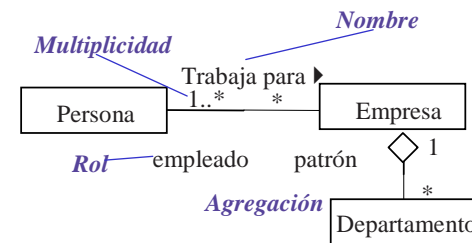


Diagramas de Clases: Relaciones

- **Asociación:** Relación estructural que indica que los objetos de una clase están conectados con los objetos de otra.
 - Una asociación entre dos clases, permite navegar desde un objeto de una clase hasta un objeto de la otra clase y viceversa.
 - Asociaciones binarias, ternarias, n-arias.
 - Asociaciones de una clase consigo misma.
- **Adornos de las Asociaciones:**
 - Nombre (con dirección, opcional)
 - Rol
 - Multiplicidad (varios tipos)
 - Agregación
 - Composición: (más fuerte que la agregación) las clases involucradas tienen el mismo ciclo de vida.



Asociación unidireccional (punta de flecha)

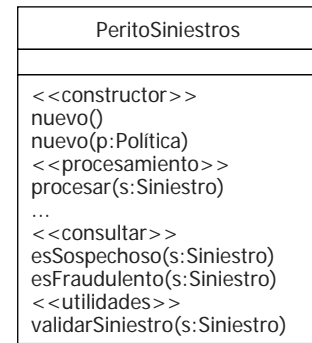




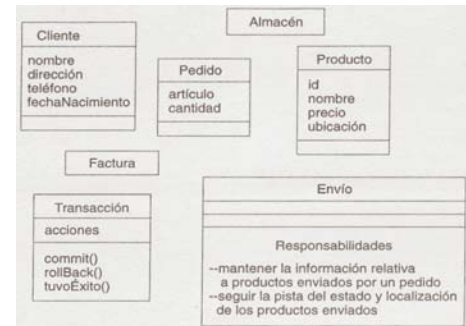
Diagramas de Clases: **Modelado**

• Clases

- Cada clase debe corresponderse con una abstracción tangible o conceptual en el dominio del usuario final o del implementador.
- Una *clase bien estructurada*:
 - No contemplar todo el detalle de las clases (abstracción precisa de modelado).
 - Contiene un conjunto pequeño, bien definido de responsabilidades.
 - Es comprensible y sencilla, a la vez que extensible y adaptable.
- Al *dibujar una clase*:
 - Atributos y métodos importantes
 - (...) → Indica que hay más atributos o métodos.
 - Clases relacionadas en el mismo diagrama
 - Utilizar estereotipos como categoría descriptiva para atributos y métodos.



Ej: Clases extraídas de un sistema de venta

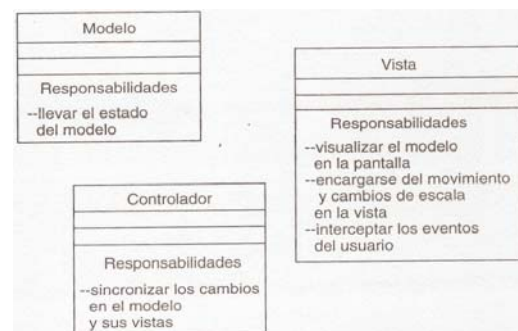


Diagramas de Clases: **Modelado**

• Responsabilidades

- **Equilibrio** para el cumplimiento de responsabilidades.
 - Clases muy grandes difíciles de cambiar y no muy reutilizables
 - Clases muy pequeñas difíciles de manejar y comprender
- Para modelar la *distribución de responsabilidades*:
 - Identificar un conjunto de clases que colaboran para que se realice cierto comportamiento.
 - Identificar las responsabilidades de cada una.
 - Dividir en clases de manera óptima.
 - Redistribuir las responsabilidades de manera adecuada.

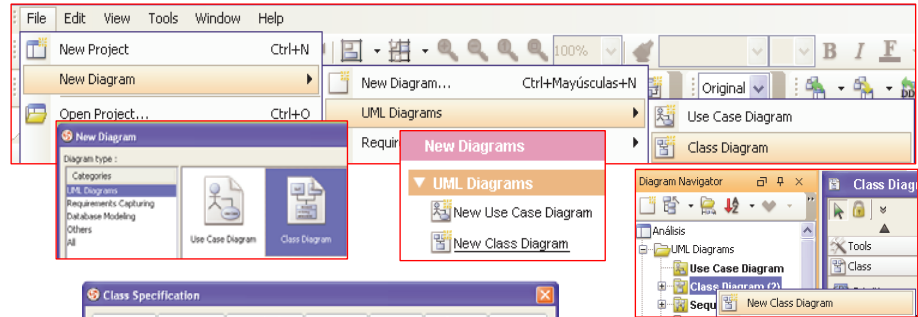
Conjunto de clases con responsabilidades equilibradas



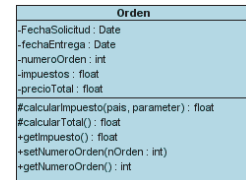
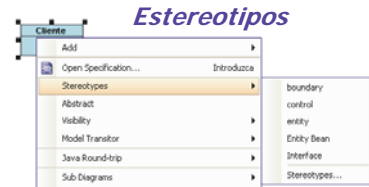
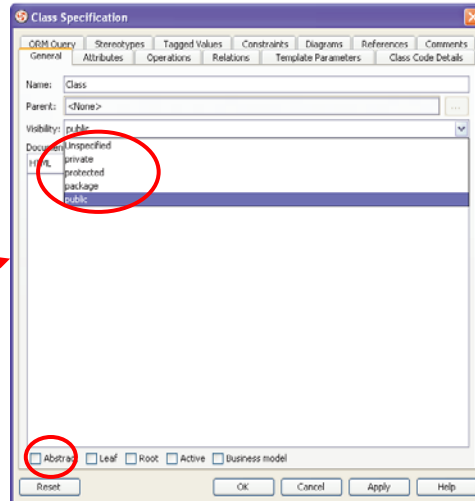
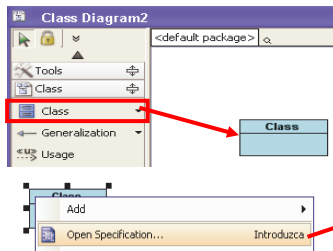


Diagramas de Clases con VP

Crear Diagrama



Crear Clase



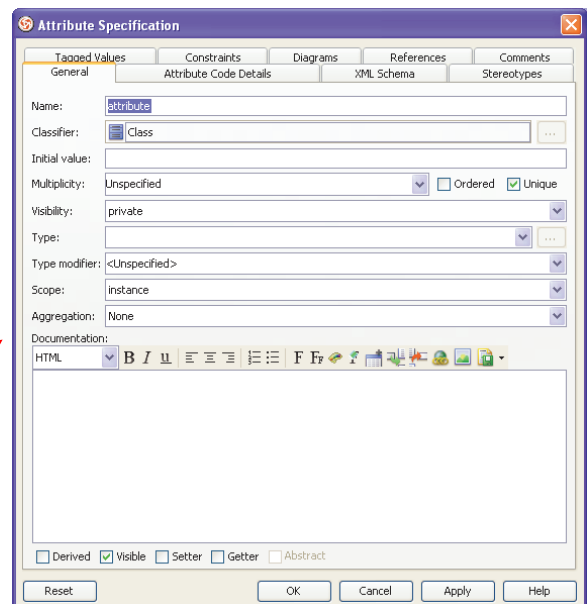
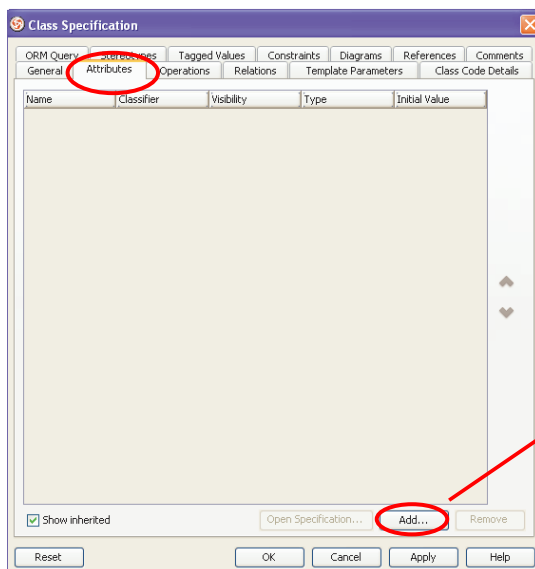
María Sierra - IS1

P5.1.11



Diagramas de Clases con VP

Indicar Atributos y sus características



- Directamente en la clase

María Sierra - IS1

P5.1.12



Diagramas de Clases con VP

Indicar Métodos y sus características

Class Specification dialog box showing the 'Operations' tab selected. The 'Operation Specification' dialog box is open, showing the 'Parameters' tab selected. A red arrow points from the 'Operations' tab in the Class Specification dialog to the 'Parameter Specification' dialog.

Parameter Specification dialog box showing the 'Name' field set to 'parameter' and the 'Operation' field set to 'operation'.

Context menu for a class showing the 'Add' option selected, with the 'Operation' option highlighted.

- Directamente en la clase



Diagramas de Clases con VP

Relaciones

Asociación

Class Diagram showing an association between 'Teatro' and 'Class'. The 'Association' tool is highlighted in the toolbar, and the 'Composición' and 'Agregación' options are indicated by red arrows.

Dependencia (Usage)

Class Diagram showing a dependency between 'Teatro' and 'Class'. The 'Dependency' tool is highlighted in the toolbar.

Generalización (Usage)

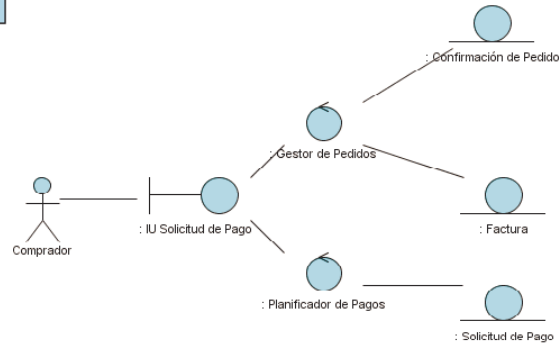
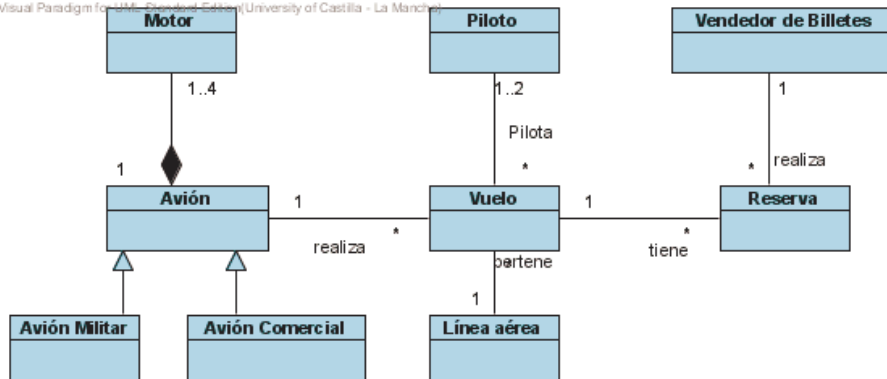
Context menu for a class showing the 'Generalization' option selected, with the 'Multiplicity' sub-menu open, showing 'Unspecified' selected.

Class Diagram showing a generalization between 'Teatro' and 'Class'. The 'Generalization' tool is highlighted in the toolbar.



Diagramas de Clases con VP

Visual Paradigm for UML - Desktop Edition (University of Castilla - La Mancha)



María Sierra - IS1

P5.1.15



Introducción al Análisis

- **Analizamos los requisitos** que se describieron en la captura de requisitos, **refinándolos** y **estructurándolos**.
- **Objetivo:** Conseguir una **comprensión** más precisa de los requisitos y una **descripción** de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero
- Lenguaje: **Modelo de Análisis** (basado en un modelo de objetos conceptual)
 - Jerarquía de paquetes del análisis que contiene clases del análisis y realizaciones de casos de uso
- El **análisis no es diseño ni implementación**
 - Diseño modelamos el sistema y encontrar su forma (incluyendo arquitectura); forma que de vía a todos los requisitos incorporados al sistema, que incluya componentes de código y que sea mantenible a largo plazo. En él se toman decisiones relativas a cómo debería ser el sistema (rendimiento, distribución,...)
- Consigue una separación de intereses que prepara y simplifica las actividades de diseño e implementación, delimitando los temas a tratar y decisiones a tomar en esas actividades

María Sierra - IS1

P5.1.16



Modelo de Casos de Uso vs Modelo de Análisis

<i>Modelo de Casos de Uso</i>	<i>Modelo de Análisis</i>
Descrito con el lenguaje del cliente	Descrito con el lenguaje del desarrollador
Vista externa del sistema	Vista interna del sistema
Estructurado por los casos de uso; proporciona la estructura a la vista externa	Estructurado por clases y paquetes estereotipados; proporciona la estructura a la vista interna
Utilizado fundamentalmente como contrato entre el cliente y los desarrolladores sobre qué debería y qué no debería hacer el sistema	Utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado
Puede contener redundancias, inconsistencias, etc...	No debería contener redundancias, inconsistencias, etc...
Captura la funcionalidad del sistema, incluida la funcionalidad significativa para la arquitectura	Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura; sirve como una primera aproximación al diseño
Define casos de uso que se analizarán con más profundidad en el modelo del análisis	Define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso



Clases de Análisis

- Abstracción de una o varias clases y/o subsistemas del diseño del sistema.
- **Características:**
 - **Req. Funcionales**, pospone los no funcionales (Req. especiales)
 - Contexto del dominio del problema (más conceptual)
 - **Comportamiento** definido mediante **responsabilidades** de nivel más alto y menos formal. (Raramente define u ofrece interface en términos de operaciones y sus firmas)
 - **Responsabilidad**: descripción textual de un conjunto cohesivo del comportamiento de una clase
 - **Atributos a un nivel alto** (tipos conceptuales y reconocibles en el dominio del problema)
 - En diseño son tipos de L. de programación
 - Con frecuencia los atributos de análisis son clases en diseño e implementación
 - **Relaciones a nivel conceptual.**
 - Navegabilidad de las asociaciones no importante → en diseño es fundamental. (Ej: pueden utilizarse generalizaciones en análisis pero no es posible utilizarlas en diseño si no las soporta el L. de programación)
- **Tipos:** de interfaz, de control o de entidad.



Clases de Análisis vs Clases de Diseño

Análisis

Orden
FechaSolicitud
FechaEntrega
NumeroOrden
Calcular el precio total()
Calcular los impuestos de gestión de la orden()
Registrar el número de orden()

Diseño

Orden
-FechaSolicitud : Date
-fechaEntrega : Date
-numeroOrden : int
-impuestos : float
-precioTotal : float
#calcularImpuesto(pais, parameter) : float
#calcularTotal() : float
+getImpuesto() : float
+setNumeroOrden(nOrden : int)
+getNumeroOrden() : int

- Req. Funcionales
- Atributos Importantes
 - Nivel alto
- Comportamiento mediante Responsabilidades
- Relaciones a nivel conceptual

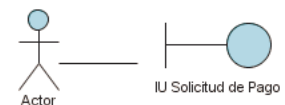
- Visibilidad:
 - - (private)
 - # (protected),
 - + (public)
- Atributos y Tipos
- Métodos:
 - Parámetros y tipos
 - Valores retorno
- Más cercano a L. Programación



Tipos de Clases de Análisis

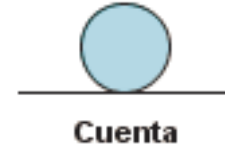
• Clases de Interfaz

- Modelan la interacción entre el sistema y sus actores, implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y los sistemas externos
 - Modelan las partes del sistema que dependen de sus actores, por lo que clarifican y reúnen los requisitos en los límites del sistema
 - Representan a menudo abstracciones de ventana, formularios, paneles, interfaces de impresoras, sensores, terminales, API, ...
 - Deben mantenerse a un nivel de abstracción alto y conceptual (suficiente que describan lo que se obtiene con la interacción y no como se ejecuta esta)
 - Cada clase interface debe asociarse al menos con un actor
- Ej: Interface del Cliente, Interface del Cajero,





Tipos de Clases de Análisis



• *Clases de Entidad*

- Modelan la información y el comportamiento asociado de algún fenómeno o concepto (persona, objeto y suceso del mundo real). Persisten durante la aplicación.
 - Derivan de una clase de entidad de negocio (o de una clase de dominio), tomada del modelo de objetos de negocio (o del modelo de dominio)
 - **Diferencia:** clases entidad objetos manejados por el sistema, clases de entidad del negocio objetos presentes en el negocio en general. Luego reflejan información que beneficia a los desarrolladores al diseñar e implementar el sistema
 - Un objeto entidad no ha de ser necesariamente pasivo y puede tener un comportamiento complejo relativo a la información que representa. Aíslan los cambios en la información que representan
 - Suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema
- Ej: Cliente, Contrato, Dirección, ...



Tipos de Clases de Análisis



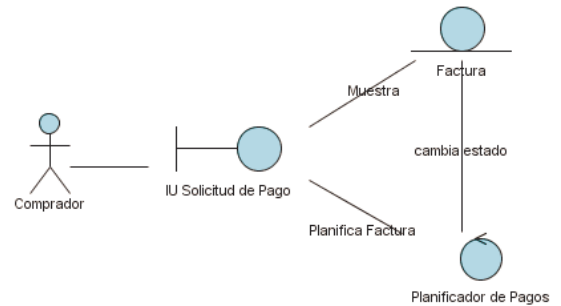
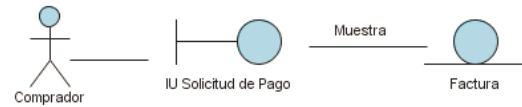
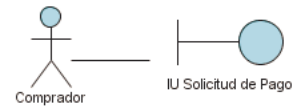
• *Clases de Control*

- Representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso concreto (un proceso laboral, de control o de cálculo)
 - También derivaciones y cálculos complejos, como la lógica de negocio, que no pueden asociarse con ninguna información concreta, de larga duración, almacenada por el sistema (es decir una clase de entidad concreta)
 - Aspectos dinámicos del sistema se modelan con clases de control, pues manejan y coordinan las acciones y los flujos de control principales, delegando trabajo a otros objetos (de interfaz, de entidad)
 - Encapsulan y aíslan los cambios del control, la coordinación, la secuencia, las transacciones y a veces la lógica del negocio compleja que implica a varios objetos
- Ej: Retirada de Efectivo, Calcular precio obra, Planificador de pagos, ...



Tipos de Clases de Análisis

- Clase de Interfaz **IU Solicitud Pago**
 - Interacción entre actor **Comprador** y Caso de Uso **Pagar Factura**. Permite al comprador: consultar facturas a pagar, facturas concretas con más detalle, solicitar pago factura. Permite descartar factura que el comprador no quiere pagar
- Clase de Entidad **Factura**
 - Representa facturas. Se asocia con el interfaz **IU Solicitud de Pago** por el cual el usuario consulta y gestiona facturas
- Clase de Control **Planificador de Pagos**
 - Responsable de la coordinación entre **IU Solicitud de Pago** y **Factura**. Acepta una solicitud de pago, como una solicitud de pagar una factura, y una fecha en la cual debe pagarse. En la fecha indicada **Planificados de Pagos** lleva a cabo el pago solicitado mediante transferencia de dinero entre las cuentas apropiadas



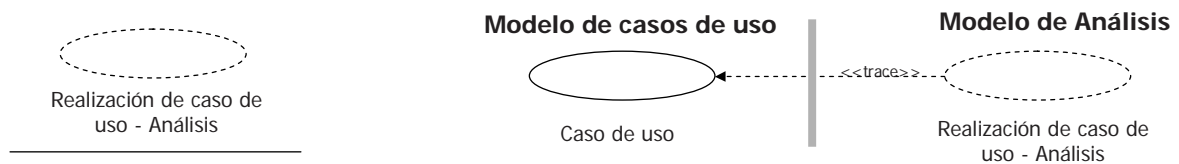
María Sierra - IS1

P5.1.23

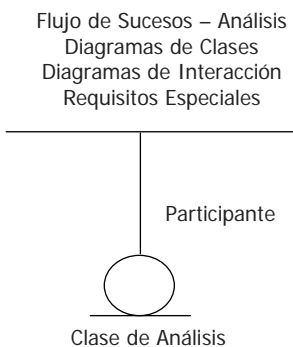


Realización de Casos de Uso de Análisis

- Colaboración dentro del modelo de análisis que describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases de análisis y de sus objetos del análisis en interacción.



Traza directa hacia un caso de uso concreto del modelo de casos de uso



- Posee una descripción textual del **flujo de sucesos**, **diagramas de clases** (clases de análisis) y **diagramas de interacción** (realización de un flujo o escenario particular del caso de uso en término de interacciones de objetos del análisis).
- Se centra en los **requisitos funcionales**. Posponen los no funcionales (Req especiales en la realización)

María Sierra - IS1

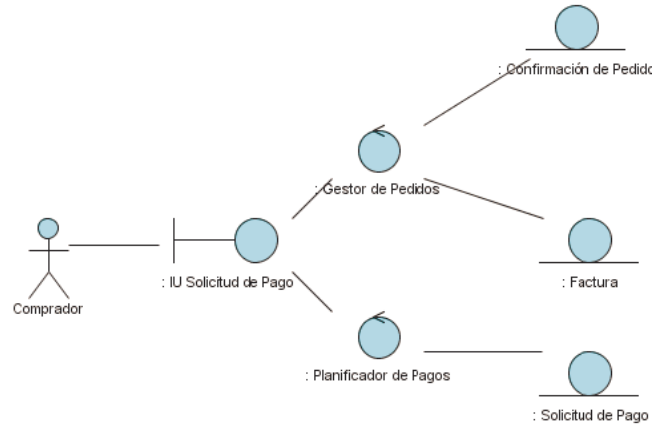
P5.1.24



Realización de Casos de Uso de Análisis

Diagramas de Clases

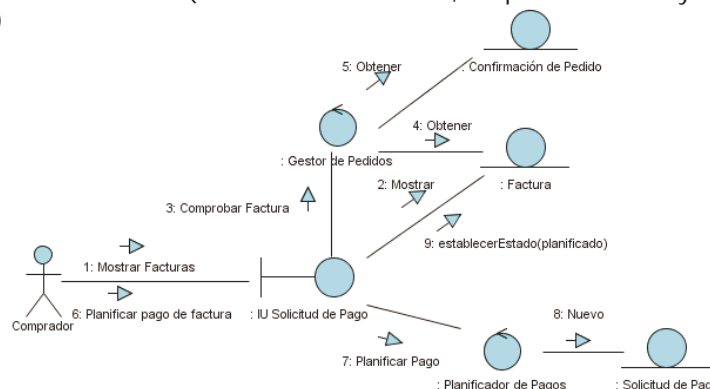
- Una clase de análisis y sus objetos normalmente participan en varias realizaciones de casos de uso, y algunas de las responsabilidades, atributos, y asociaciones de una clase concreta suelen ser sólo relevantes para una única realización de caso de uso. → Es importante durante el análisis coordinar todos los requisitos sobre una clase y sus objetos que pueden tener diferentes casos de uso.



Realización de Casos de Uso de Análisis

Diagramas de Interacción

- La secuencia de acciones en un caso de uso comienza cuando un actor envía un mensaje al sistema.
 - Interior del sistema: un objeto de interfaz recibirá este mensaje del actor y enviará a su vez el mensaje a algún otro objeto, y de este modo los objetos implicados interactuarán para llevar a cabo el caso de uso.
- En análisis nos interesa identificar requisitos y responsabilidades sobre objetos
- Flujo de sucesos:** una realización de un casos de uso pueden ser difícil de leer, es útil un texto adicional que los explique.
 - (En términos de objetos, particularmente de objetos de control que interactúan para llevar a cabo el caso de uso. (no menciona atributos, responsabilidades y asociaciones pues cambian)





Realización de Casos de Uso de Análisis

• Ejemplo de Flujo de Suceso

- El comprador consulta a través de IU Solicitud de Pago las facturas gestionadas por el sistema para encontrar las recibidas (1,2). El IU Solicitud de Pago utiliza el Gestor de Pedidos para comprobar las facturas con sus correspondientes confirmaciones de pedidos (3,4,5), antes de mostrar la lista de facturas del comprador (comprobación que depende de las reglas de negocio del comprador: comparación de precio, fecha entrega, contenido factura con confirmación pedido,...). El objeto Gestor de Pedidos utiliza las reglas del negocio para decidir las preguntas a hacer (4,5) a los objetos Confirmación Pedido y Factura y cómo analizar las respuestas. Cualquier factura sospechosa quedará marcada de alguna forma por el IU Solicitud de Pago, quizás mediante un color diferente que lo resalte
- El comprador selecciona una factura mediante el IU Solicitud de Pago y planifica su pago (6). El IU Solicitud de Pago solicita al Planificador de Pagos que planifique el pago de la factura (7). Después el Planificador de Pagos crea una solicitud de pago (8). El IU Solicitud de Pago cambia a continuación el estado de la factura a "planificada" (9)
- El Planificador de Pagos iniciará el pago en el día debido (no se muestra en diagrama)

• Requisitos Especiales

- Descripciones textuales que recogen todos los requisitos no funcionales sobre la realización de un caso de uso
- Ej Pagar Factura:
 - Cuando el comprador solicite ver las facturas recibidas, no debería tardar más de medio segundo en mostrarlas por pantalla
 - Las facturas deben pagarse empleando el estándar SET



Realización de Casos de Uso de Análisis

Análisis de una Clase

- **Identificar Responsabilidades:** recopilando todos los roles que cumple en diferentes realizaciones de casos de uso
- Identificar:
 - Atributos
 - Asociaciones y Agregaciones
 - Generalizaciones
- **Captura Req. Especiales** (ej rango, volumen, frecuencia actualización)

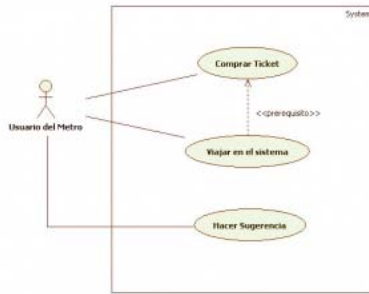
Análisis de un Caso de Uso

- **Identificar Clases de Análisis** cuyos objetos son necesarios para llevar a cabo el flujo de sucesos de casos de uso. (Entidad, Interface, Control)
- **Descripción de interacciones:** si el caso de uso tiene flujos o subflujos diferenciados y distintos puede ser útil crear un diagrama de colaboración para cada flujo.
- **Captura de Req. Especiales** sobre las clases que intervienen.



Realización de Casos de Uso de Análisis

Modelo de Contexto



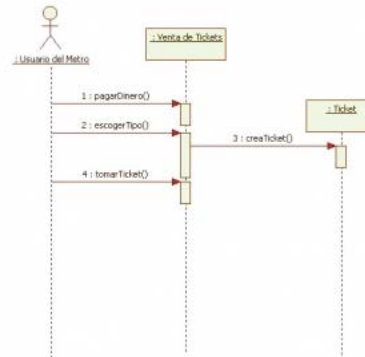
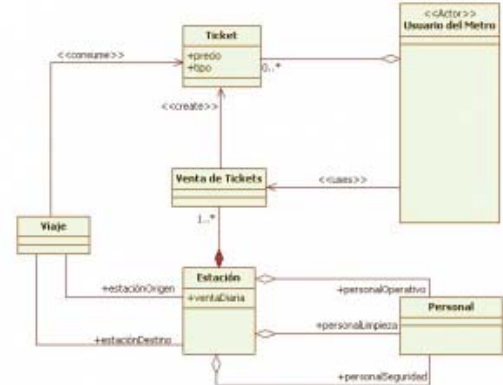
Caso de Uso: *Compra de Ticket*

Actor: Usuario del Metro.
Descripción breve: El usuario del metro compra un boleto del sistema luego del pago de la cantidad apropiada.

1. ¿Qué clases del modelo de dominio intervienen en el caso de uso?
 - No existe una respuesta necesariamente correcta, queda a juicio del analista el tomar una u otra clase según su visión y particular enfoque del problema. Clases: *Usuario del Metro*, *Venta de Ticket*, y *Ticket*.
2. Indicar cuales son los mensajes que estas instancias deben intercambiar para completar la funcionalidad. Este intercambio lo presentamos en un diagrama de UML, típicamente un diagrama de interacción

María Sierra - IS1

Modelo de Dominio



P5.1.29