



Tema 4.3: Modelado conceptual

Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática



Índice

1. Captura de Requisitos
2. Análisis de Requisitos



Objetivos

- Comprender por qué es importante establecer los límites de un sistema y modelar su contexto
- Introducirse en el Lenguaje Unificado de Modelado (UML) y aprender como puede usarse para desarrollar modelos del sistema
- Comprender los conceptos del modelado de requisitos a través de casos de usos



Índice

1. Captura de Requisitos
2. Análisis de Requisitos
 1. Tareas del análisis de requisitos
 2. Técnicas para análisis de requisitos
 3. Modelado conceptual
 1. Introducción a UML
 2. UML y Requisitos
 3. Diagramas de Casos de Uso



3. Modelado conceptual

- Análisis de requisitos, 4 tareas:
 - Clasificación
 - **Modelado Conceptual**
 - Localización
 - Negociación
- El **modelado conceptual** de los requisitos busca la comprensión del problema antes de iniciar el diseño de la solución.
- Principales Técnicas de Modelado de Requisitos:
 - Diagramas de Casos de Uso
 - Flujos de Datos y de Control (DFDs)
 - Modelos de Estados (máquinas de estados)
 - Diagramas de Eventos
 - Diagramas de Interacción de usuarios
 - Modelos de Objetos y de Datos



Índice

1. Captura de Requisitos
2. Análisis de Requisitos
 1. Tareas del análisis de requisitos
 2. Técnicas para análisis de requisitos
 3. Modelado conceptual
 1. Introducción a UML
 2. UML y Requisitos
 3. Diagramas de Casos de Uso



3.1. Introducción a UML

- Énfasis en los requisitos UML.
- Son las siglas de Unified Modeling Language: un lenguaje visual que permite **visualizar, especificar, construir y documentar** los artefactos de sistemas software, incluyendo su estructura y diseño
- Aceptado como un estándar *de facto*
- Puede ser usado para:
 - Diseño software (para lo que surgió en principio)
 - Procesos de negocio
 - **Captura de requisitos**
 - **Documentar** un sistema, proceso u organización existente ...



3.1. Introducción a UML

- La notación puede ser “auto-extendida” con mecanismos propios: estereotipos, valores etiquetados, restricciones.
- De esta forma, UML puede adaptarse a las necesidades de diferentes dominios o tipos de sistemas: de tiempo real, sistemas tolerantes a fallos, sistemas embebidos, etc.
- UML viene acompañado por un lenguaje para expresar restricciones sobre los modelos construidos
 - **OCL** (Object Constraint Language)



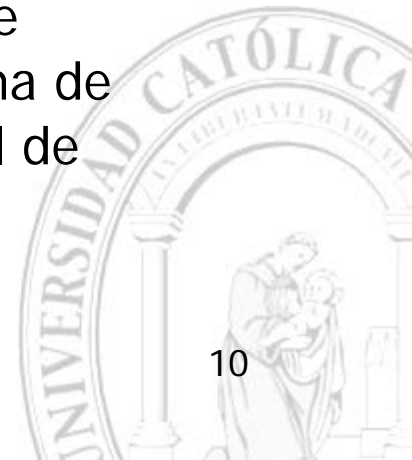
3.1. Introducción a UML

- Está construido sobre conceptos fundamentales de la orientación a objetos (OO), aunque puede ser utilizado para sistemas no-OO
- Elegir bien la colección de modelos a construir
- Cada modelo puede expresarse a diferentes niveles de precisión y usarse en diferentes momentos del ciclo de vida
- Los modelos deben ser realistas
- Usar varios modelos (no sólo uno) para representar los diferentes aspectos de un mismo sistema
- No todas las capacidades son necesariamente útiles en todos los dominios o aplicaciones que utilicen UML



3.1. Introducción a UML

- UML 2.0 define 13 tipos de diagramas (bloque de construcción básico) divididos en dos categorías:
 - **Estructurales:** capturan la organización de los elementos en el sistema (p. ej. como los objetos se relacionan entre si). 6 diagramas: de clases, de objetos, de componentes, de estructura compuesta, de paquetes y de despliegue
 - **Comportamiento:** indican como se comportan los elementos del sistema 7 diagramas: casos de uso, de actividades, de estados y los diagramas de interacción en los que hay cuatro tipos diferentes: diagrama de secuencia, de comunicaciones (anterior diagrama de colaboraciones), de tiempos y diagrama general de interacción



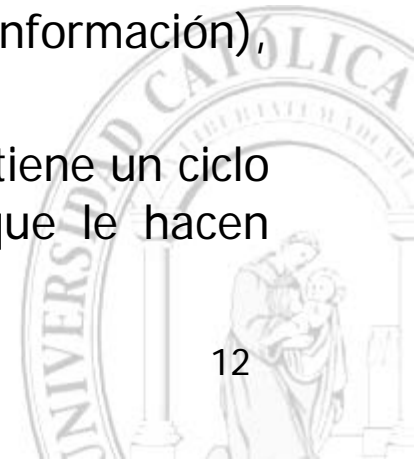
Índice

1. Captura de Requisitos
2. Análisis de Requisitos
 1. Tareas del análisis de requisitos
 2. Técnicas para análisis de requisitos
 3. Modelado conceptual
 1. Introducción a UML
 2. UML y Requisitos
 3. Diagramas de Casos de Uso



3.2. UML y requisitos

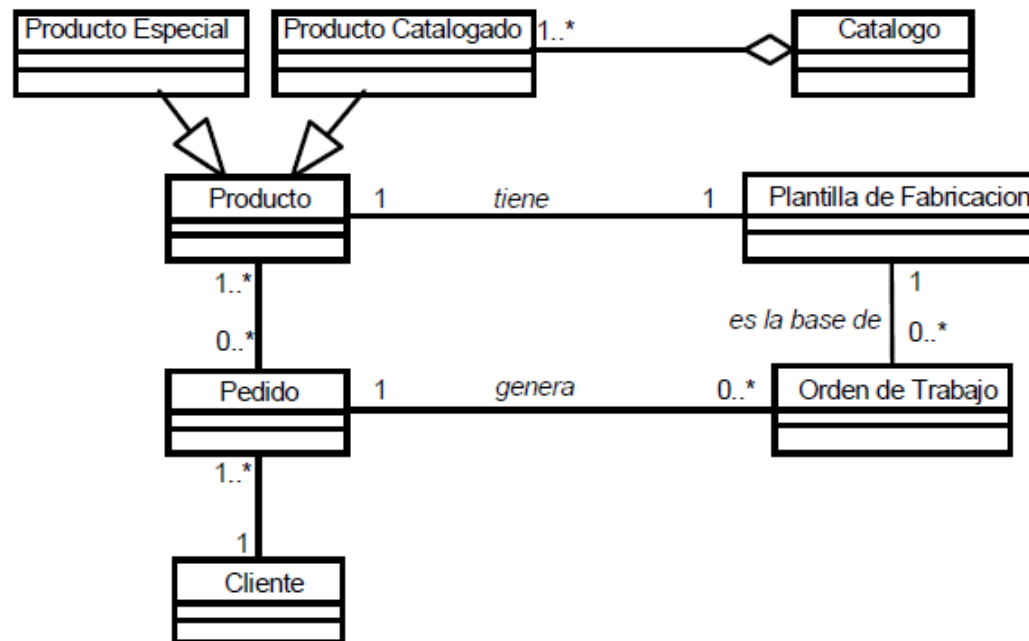
- Durante el análisis de requisitos se intenta descubrir qué quieren los *stakeholders* que haga el sistema
 - **Casos de uso:** describen como las personas interactúan con el sistema
 - **Diagramas de clases:** ofrecen una perspectiva conceptual del sistema, útil para establecer un vocabulario riguroso sobre el dominio
 - **Diagramas de actividades:** muestran como interactúa el software a partir de las actividades realizadas por los usuario, así como el contexto para los casos de uso y como funcionan los casos de uso complicados
 - **Diagramas de interacción:** describen interacción (no información), es decir, la funcionalidad del sistema a construir.
 - **Diagramas de estados:** útiles cuando algún concepto tiene un ciclo de vida complejo, con diferentes estados y eventos que le hacen cambiar de estado



3.2. UML y requisitos

Diagramas de Clases

- Formados principalmente por clases (representa a un grupo de objetos con un comportamiento común) y relaciones entre ellas



3.2. UML y requisitos

Diagramas de Interacción

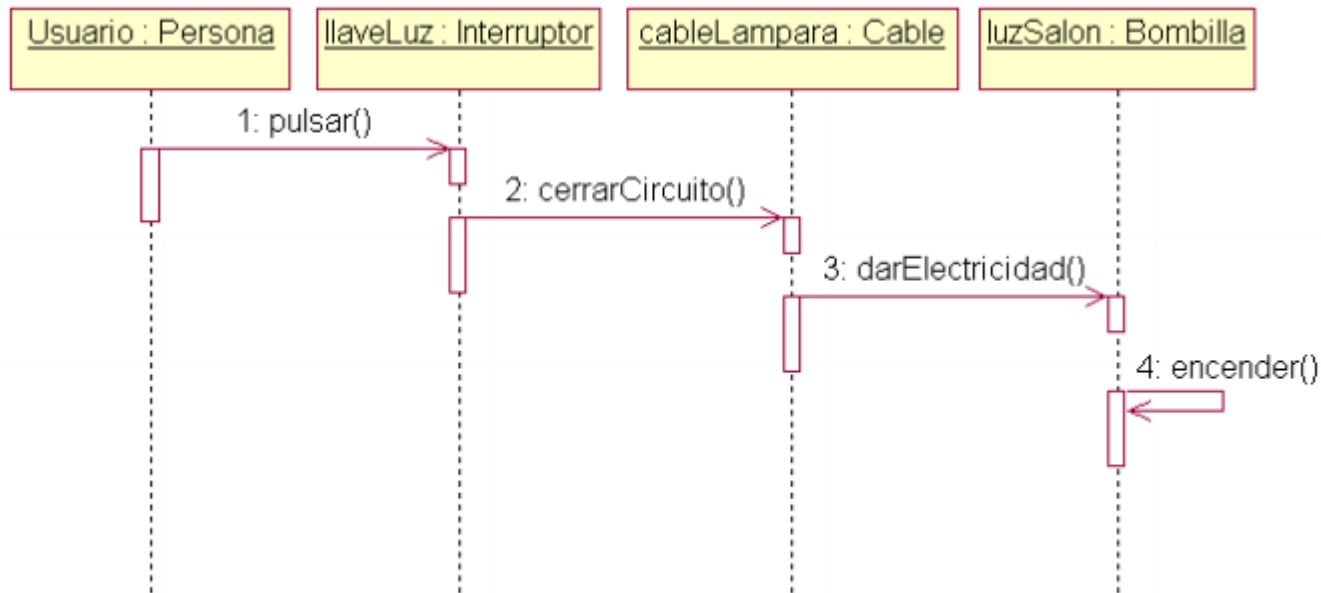
- Los diagramas de interacción describen como grupos de objetos colaboran entre si para conseguir un comportamiento deseado
- Diagramas de Secuencia
 - Énfasis en la ordenación temporal de los mensajes entre objetos
- Diagramas de Colaboración
 - Énfasis en la estructura de los objetos que envían y reciben mensajes (también se explicita la ordenación temporal de los mensajes)
- Son semánticamente equivalentes



3.2. UML y requisitos

Diagramas de Interacción: Diagramas de Secuencia

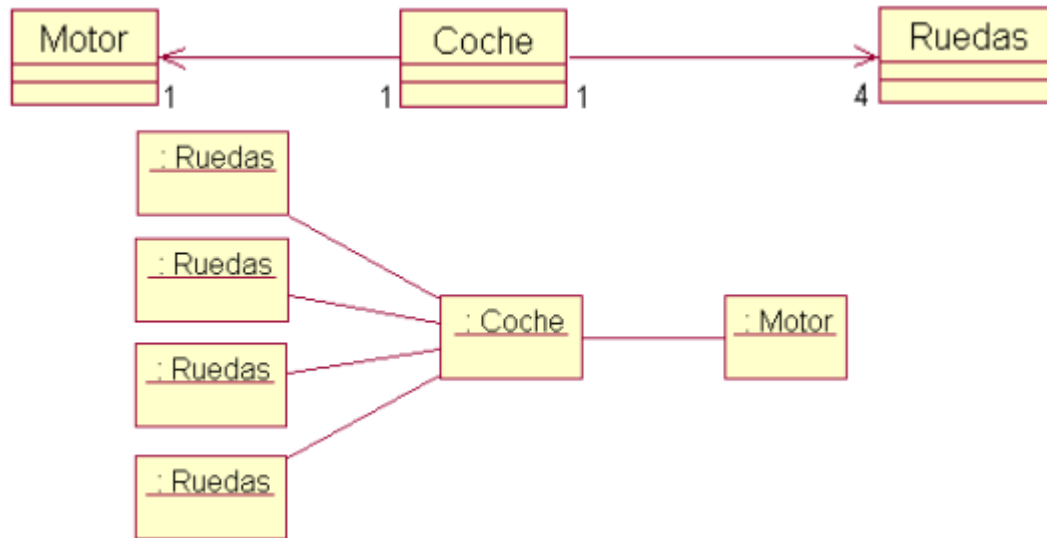
- Énfasis en la ordenación temporal de los mensajes entre objetos



3.2. UML y requisitos

Diagramas de Colaboración

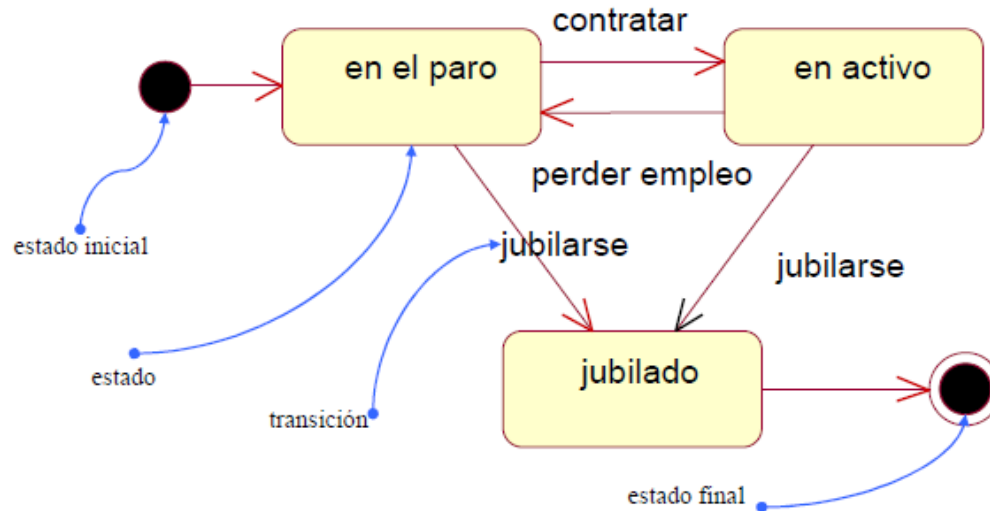
- Énfasis en la estructura de los objetos que envían y reciben mensajes (también se explicita la ordenación temporal de los mensajes)



3.2. UML y requisitos

Diagramas de Estado

- Los diagramas de estado (**Statecharts**) son útiles para representar el comportamiento de un sistema, usando máquinas de estados (autómatas de estados finitos), con estados y transiciones



Índice

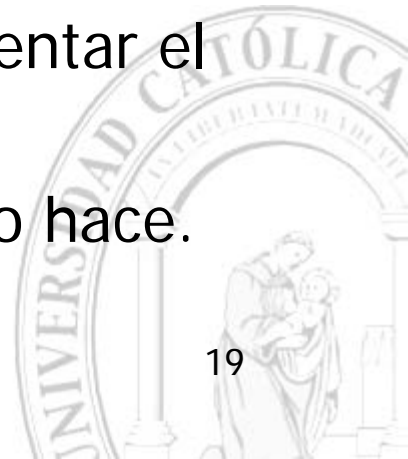
1. Captura de Requisitos
2. Análisis de Requisitos
 1. Tareas del análisis de requisitos
 2. Técnicas para análisis de requisitos
 3. Modelado conceptual
 1. Introducción a UML
 2. UML y Requisitos
 3. Diagramas de Casos de Uso



3.3. Diagrama de Casos de Uso

¿Qué es un caso de uso?

- Describe un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta el sistema para producir un resultado observable de valor para un actor.
 - Interacción típica entre usuario (actor) y el sistema.
 - Produce algo de valor para algún actor
- Se utilizan durante la captura de requisitos y el análisis para visualizar, especificar, construir y documentar el comportamiento esperado del sistema
- Describe qué hace el sistema, pero no cómo lo hace.



3.3. Diagrama de Casos de Uso

¿Para qué sirven?

- Para capturar el comportamiento deseado del sistema sin tener que especificar como se implementa ese comportamiento
- Cómo medio de comprensión del sistema para desarrolladores, usuarios finales y expertos del dominio.
- Ayudan a validar la arquitectura y a verificar el sistema en el transcurso de desarrollo de este.



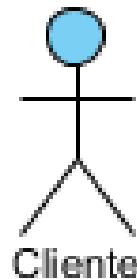
3.3. Diagrama de Casos de Uso

Notación

- Caso de uso:



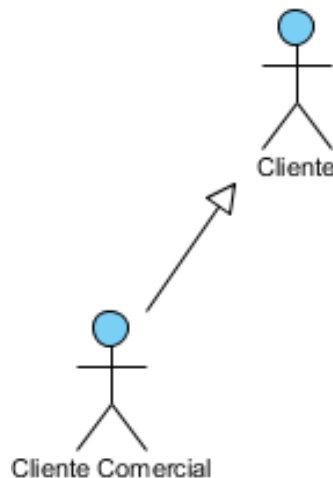
- Actor:



3.3. Diagrama de Casos de Uso

Actores

- *Actor*: representa un conjunto coherentes de roles que los usuarios de los casos de uso representan al interactuar con éstos.
- Representa un rol que es desempeñado por una persona, un dispositivo hardware o incluso otro sistema.
- Se pueden definir categorías generales de actores y especializarlos a través de relaciones de generalización.



3.3. Diagrama de Casos de Uso

Actores

- Actor y caso de uso se pueden comunicar a través de una asociación en donde cada uno de ellos puede enviar y recibir mensajes.
- La misma persona, dispositivo o sistema puede interpretar varios papeles ("**roles**"), como actores distintos
- El nombre del actor describe el papel desempeñado



3.3. Diagrama de Casos de Uso

Flujo de eventos

- El comportamiento de un caso de uso se puede especificar describiendo un flujo de eventos de forma textual (lo suficientemente claro para que alguien ajeno al sistema lo entienda fácilmente)
- Se debe incluir:
 - Cómo y cuándo empieza y acaba
 - Cuando interactúa con los actores y que objetos intercambian
 - El flujo básico y los flujos alternativos



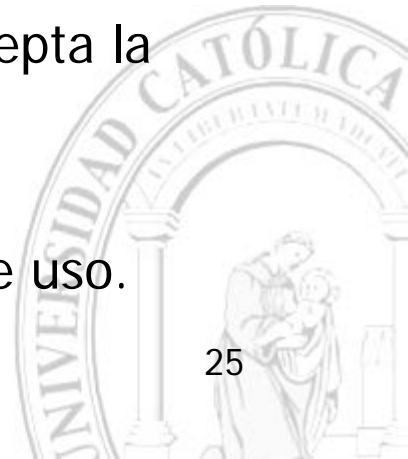
3.3. Diagrama de Casos de Uso

Flujo de eventos

- Ejemplo: CU *ValidarUsuario* en un cajero automático



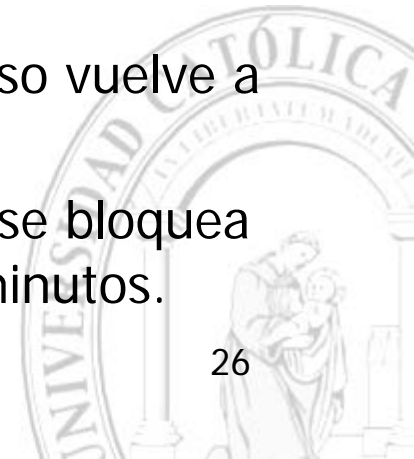
- *Flujo de eventos principal:*
 - El caso de uso comienza cuando el Sistema pide al Cliente un número de identificación personal (PIN).
 - El Cliente introduce el PIN a través del teclado y acepta la entrada pulsando la tecla Enter.
 - El Sistema comprueba si el PIN es válido.
 - El Sistema acepta la entrada y así finaliza el caso de uso.



3.3. Diagrama de Casos de Uso

Flujo de eventos

- *Flujo de Eventos Excepcional 1:*
 - El Cliente puede cancelar el proceso en cualquier momento pulsando el botón Cancelar reiniciando de esta forma el caso de uso.
- *Flujo de Eventos Excepcional 2:*
 - El Cliente puede borrar un PIN en cualquier momento antes de introducirlo pulsando Enter y puede teclear un nuevo PIN.
- *Flujo de Eventos Excepcional 3:*
 - Si el Cliente introduce un PIN inválido, el caso de uso vuelve a empezar.
 - Si esto ocurre tres veces en una sesión, el sistema se bloquea impidiendo que el Cliente use el cajero durante 2 minutos.



3.3. Diagrama de Casos de Uso

Escenarios

- Cada flujo de eventos (principal o alternativo) representa un **Escenario**:
 - Secuencia específica de acciones que ilustra un comportamiento específico del sistema.
- Por tanto, **un escenario es una instancia de un caso de uso**.
 - Se pueden representar con diagramas de interacción
- Un caso de uso describe un conjunto de escenarios.
- Cada escenario representa un posible flujo a través de todas las variantes del caso de uso.

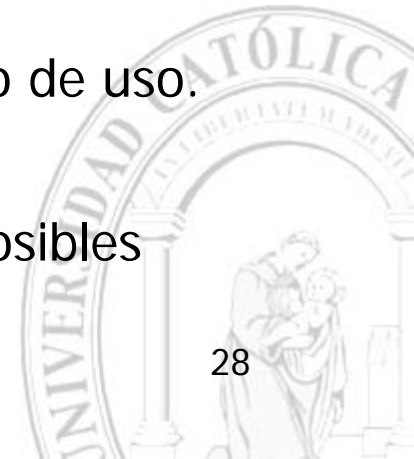


3.3. Diagrama de Casos de Uso

Especificación

Una Especificación de un Caso de Uso suele incluir:

- **Descripción general.**
 - Reflejando, posiblemente, uno o varios requisitos funcionales del sistema o una parte de algún requisito.
- **Pre-condiciones**
 - Condiciones que deben cumplirse para que se realice el caso de uso.
- **Post-condiciones:**
 - Condiciones que se cumplen posteriormente al caso de uso.
- **Escenarios**
 - Con la descripción de todos los flujos de eventos posibles dentro del caso de uso



3.3. Diagrama de Casos de Uso

Especificación

Caso de uso 1	Realizar Depósito
Descripción	- Establecer un depósito económico de cara a futuras interacciones con el sistema.
Actores	- Usuario. - Entidad colaboradora.
Precondiciones	- Disponer de un certificado de usuario expedido por un organismo autorizado al amparo de la normativa tributaria.
Postcondiciones	- El sistema ha establecido el depósito económico.
Escenario principal	<ol style="list-style-type: none">1. El usuario desea construir un depósito en la aplicación.2. El sistema muestra un formulario para solicitar la construcción del depósito.3. El usuario introduce su NIF, la entidad financiera donde tiene abierta su cuenta, los 20 dígitos que la identifican y el importe de la constitución del depósito.4. El sistema valida los datos introducidos por el usuario.5. El sistema muestra al usuario las <i>condiciones de la entidad colaboradora</i>.6. El usuario acepta las condiciones de la <i>entidad colaboradora</i>.7. El sistema envía los datos del usuario a la entidad colaboradora.8. La entidad colaboradora genera automáticamente, a partir de los datos introducidos por el usuario, un código único (NRC) que identifica al depósito.9. El usuario recibe el NRC de la entidad colaboradora correspondiente al depósito establecido.10. El sistema establece el depósito del usuario.
Excepciones	<ol style="list-style-type: none">6a. El usuario rechaza las condiciones de la entidad colaboradora.<ol style="list-style-type: none">1. El sistema cancela el depósito.8a. La entidad colaboradora no autoriza la creación del depósito por parte del usuario.<ol style="list-style-type: none">1. Se notifica del error al usuario y se cancela el depósito.
Comentarios	Ningún comentario

3.3. Diagrama de Casos de Uso

Especificación

- Con Visual Paradigm

<http://www.youtube.com/watch?v=TZhS--jvSqY>

<http://caaeii.cl/wp-content/uploads/2012/07/GuiaVisualParadigm8.0.pdf>



3.3. Diagrama de Casos de Uso

Especificación

- Con Visual Paradigm

The screenshot displays the 'Validar usuario' use case in the Visual Paradigm software. The interface includes a title bar 'Validar usuario Detalles', a menu bar with options like 'Información', 'Use Case Notes', 'Flow of Events', 'Detalles', 'Requisitos', 'Diagramas', 'Test Plan', 'Referencias', and 'Descripción'. Below the menu bar is a toolbar with various icons and a text area containing 'Agency FB' and the number '8'. The main workspace is divided into two panes: 'Principal' on the left and a detailed specification table on the right.

Validar usuario

Información Use Case Notes Flow of Events Detalles Requisitos Diagramas Test Plan Referencias Descripción

Agency FB 8

Super Use Case

Author Mayte

Date 22-oct-2014 12:55:03

Brief Description Establecer un depósito económico de cara a futuras interacciones con el sistema.

Preconditions Disponer de un certificado de usuario expedido por un organismo autorizado al amparo de la normativa tributaria.

Post-conditions El sistema ha establecido el depósito económico

Flow of Events

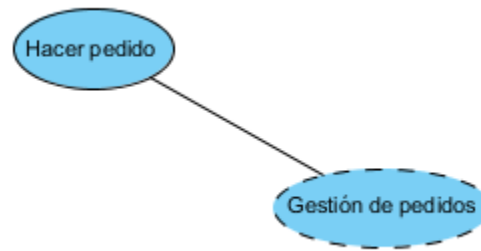
	Actor Input	System Response
1	El usuario desea construir un depósito en la aplicación.	
2		El sistema muestra un formulario para solicitar la construcción del depósito.
3	Usuario introduce su NIF, la entidad financiera donde tiene abierta su cuenta, los 20 dígitos que la identifican y el importe de la constitución del depósito.	
4		El sistema valida los datos introducidos por el usuario.

"Use Case Description" function is obsolete. All "Use Case Description" functions are replaced by "Flow of Events" and "Details" tab. Your Use Case Description can be converted to new function in Professional Edition or above, which supports "Flow of Events" tab.

3.3. Diagrama de Casos de Uso

Colaboraciones

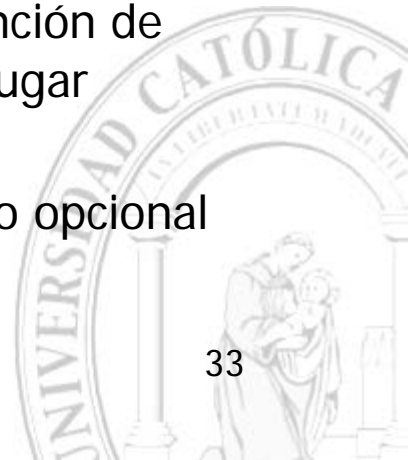
- Un caso de uso captura el comportamiento deseado de un sistema (el qué) sin especificar cómo se implementa.
 - El caso de uso se debe implementar en las actividades posteriores del proceso de desarrollo.
- La realización de un caso de uso expresa explícitamente la colaboración que implementa el caso de uso.



3.3. Diagrama de Casos de Uso

Relaciones

- Generalización:
 - El caso hijo hereda el comportamiento y el significado del padre
 - El hijo puede añadir o redefinir el comportamiento del padre.
- Inclusión:
 - Un caso base incorpora el comportamiento de otro caso en el lugar especificado en el caso base.
- Extensión:
 - Un caso base puede incorporar de forma opcional (en función de alguna condición) el comportamiento de otro caso en el lugar especificado en el caso base.
 - La funcionalidad del caso base se extiende con la del caso opcional



3.3. Diagrama de Casos de Uso

Relaciones: Generalización

- Relaciona un caso de uso especializado con uno más general.
- El caso de uso hijo hereda el comportamiento y el significado del caso de uso padre.
- El caso hijo puede:
 - Ser colocado en cualquier lugar donde aparezca el padre.
 - Añadir o redefinir el comportamiento del padre.



3.3. Diagrama de Casos de Uso

Relaciones: Generalización

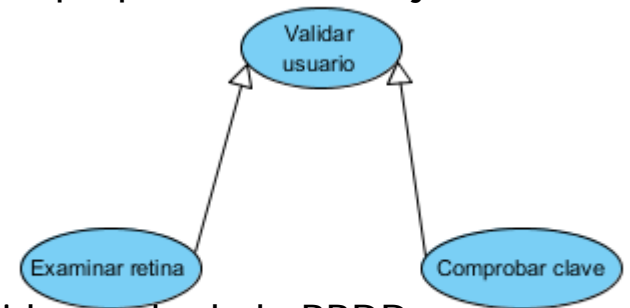
- Ejemplo

Validar Usuario:

- El CU es abstracto por lo que su comportamiento lo proporcionan los hijos

Comprobar Clave:

- Obtener contraseñas de la BBDD
- Pedir al usuario la contraseña
- El usuario introduce la contraseña
- Comprobar si la contraseña introducida coincide con la de la BBDD



Examinar Retina:

- Obtener lecturas de retinas almacenadas en la BBDD
- Escanear la retina del usuario y obtener lectura de retina
- Comprobar si la lectura de retina del usuario coincide con la de la BB

3.3. Diagrama de Casos de Uso

Relaciones: Inclusión

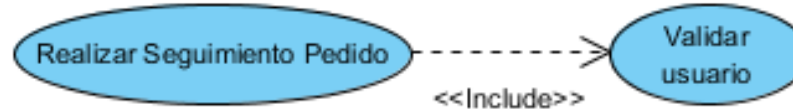
- Se usa para evitar describir el mismo flujo de eventos repetidas veces.
- El comportamiento común se pone en un caso de uso aparte.
- Si los casos de uso A y B presentan una parte común, ésta se puede sacar a un tercer caso de uso C. Entonces, habrá una relación “include” del caso de uso A al C y otra del B al C.
- Para especificarla en el flujo de eventos se debe escribir include seguido del nombre del caso de uso que se quiere incluir.



3.3. Diagrama de Casos de Uso

Relaciones: Inclusión

- Ejemplo:



Realizar Seguimiento del Pedido.

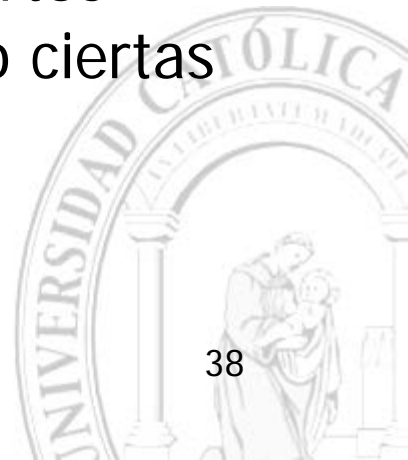
- *Flujo de Eventos Principal:*
 - Obtener y Verificar el Número de Pedido
 - Include (Validar Usuario)
 - Examinar el estado de cada parte del pedido
 - Preparar un informe para el usuario



3.3. Diagrama de Casos de Uso

Relaciones: Extensión

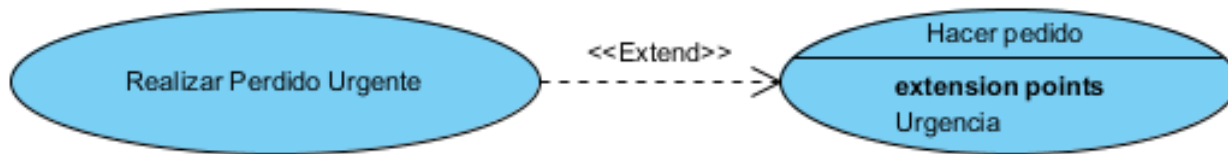
- Un caso extiende el comportamiento de otro caso (base).
- Sólo es posible en ciertos puntos (puntos de extensión)
 - Un caso de uso puede tener varios puntos de extensión.
- Sirve para separar el comportamiento “obligatorio” del comportamiento “opcional” o para modelar ciertos subflujos de eventos que se ejecutan sólo bajo ciertas condiciones.



3.3. Diagrama de Casos de Uso

Relaciones: Extensión

- Ejemplo:



Hacer Pedido.

- *Flujo de Eventos Principal:*
 - Obtener los productos pedidos por el Cliente
 - **Extension Point: Urgencia (Realizar Pedido Urgente)**
 - Enviar el pedido



3.3. Diagrama de Casos de Uso

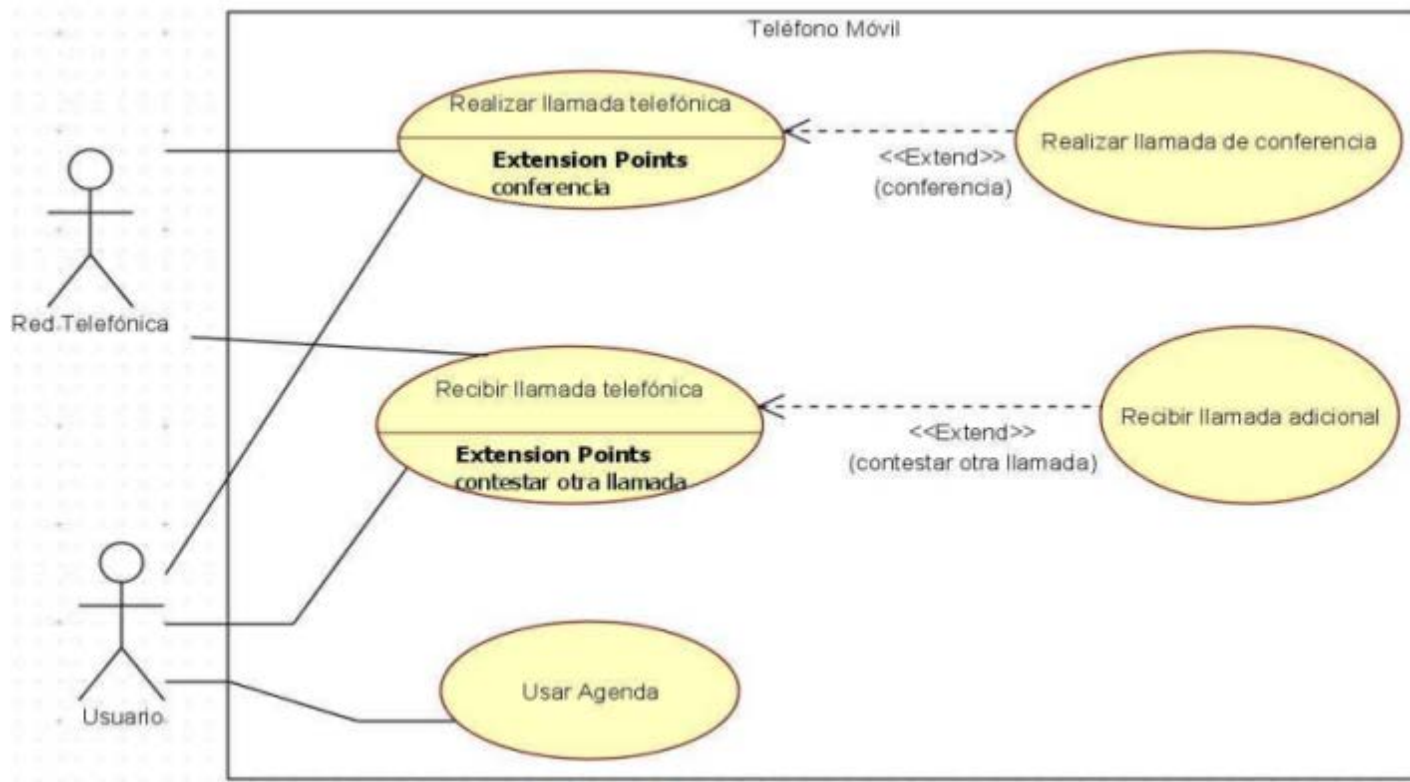
Diagramas de Casos de Uso

- Se utilizan para el modelado de aspectos estáticos de la Vista de Casos de Uso de un sistema.
- Un diagrama de casos de uso contiene:
 - Elementos: Casos de Uso, Actores y Sujetos
 - Relaciones: Dependencia (<<Extend>> e <<Include>>), Generalización y Asociación
 - También pueden contener:
 - Paquetes (para agrupar)
 - Notas y restricciones (para anotar)



3.3. Diagrama de Casos de Uso

Diagramas de Casos de Uso



3.3. Diagrama de Casos de Uso

Diagramas de Casos de Uso

Los Diagramas de Casos de Uso sirven para modelar:

- El Contexto y los Requisitos de un Sistema
- El Comportamiento de un Elemento
 - Sistema, Subsistema, Componente, Clase
- Además, los casos de uso sirven para validar la arquitectura y para verificar que el sistema evolucione durante el desarrollo de manera congruente.
 - Son fuentes excelentes de pruebas de integración y de sistema.



3.3. Diagrama de Casos de Uso

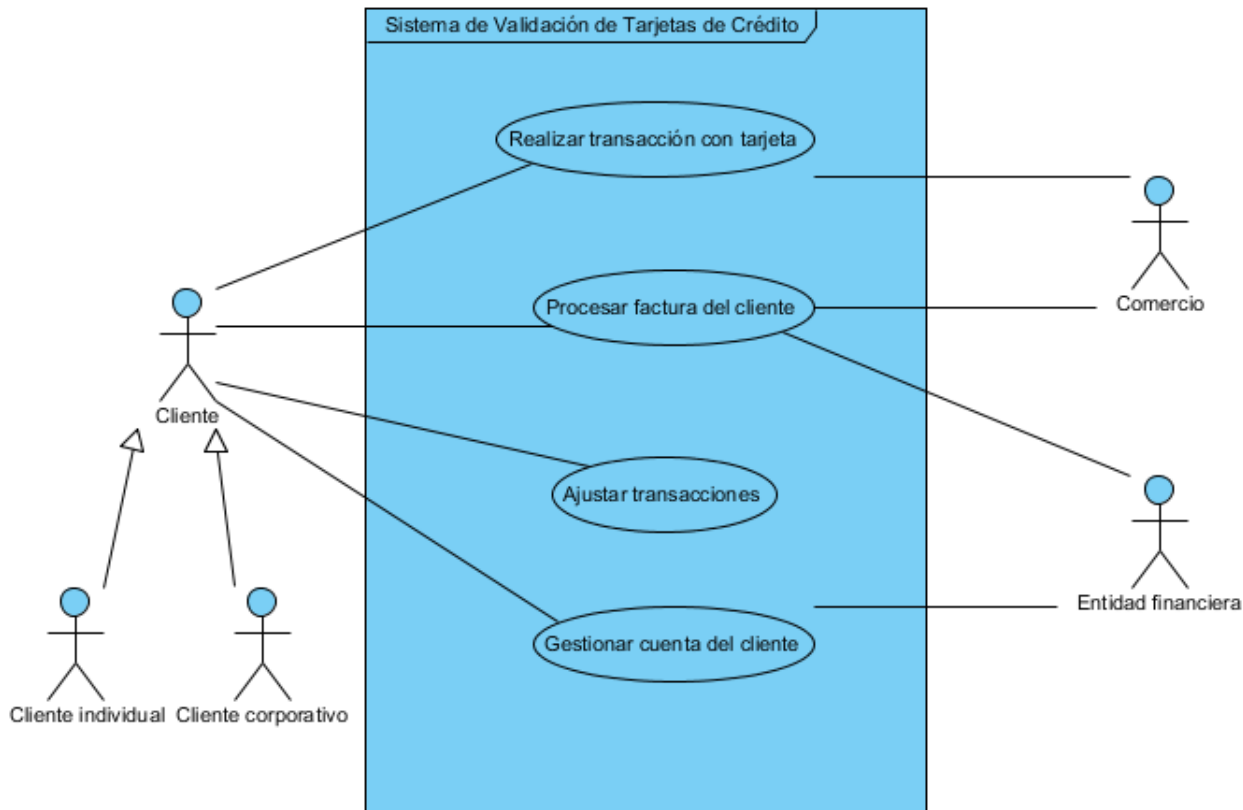
Modelo de contexto

- El modelo de contexto delimita el sistema, identificando sus fronteras.
- Se obtiene:
 - Identificando los actores externos al sistema pero que interactúan con el
 - Organizando los actores en jerarquías de generalización.
 - Modelando esos actores y jerarquías en un diagrama de casos de uso.



3.3. Diagrama de Casos de Uso

Modelo de contexto



3.3. Diagrama de Casos de Uso

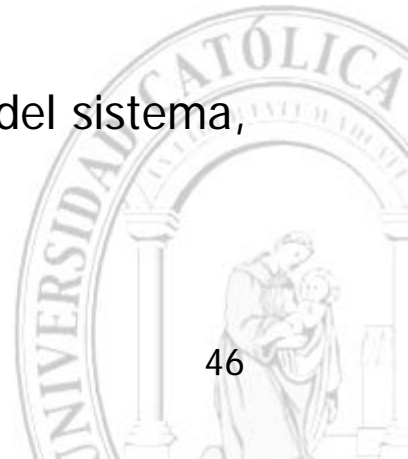
Ventajas

- “Ofrecen un medio sistemático e intuitivo para capturar los requisitos funcionales, centrándose en el valor añadido para el usuario”
- Dirigen todo el proceso de desarrollo puesto que la mayoría de actividades (planificación, análisis, diseño, validación, test,..) se realizan a partir de los casos de uso.
- Mecanismo importante para soportar “trazabilidad” entre modelos.



Puntos clave

- Un modelo es una vista abstracta de un sistema que prescinde de algunos detalles del mismo. Pueden desarrollarse modelos del sistema complementarios para presentar otra información sobre dicho sistema.
- Los modelos de objetos describen las entidades lógicas del sistema y su clasificación y agregación.
- Los casos de uso describen como las personas interactúan con el sistema.
- Los diagramas de clases ofrecen una perspectiva conceptual del sistema, que nos aporta conocimiento sobre el dominio.



Puntos clave

- Los modelos de secuencia muestran las interacciones entre actores y objetos en un sistema se utilizan para modelar el comportamiento dinámico.
- Los modelos de colaboración resaltan la organización estructural de los objetos que envía y reciben mensajes. Muestran explícitamente las relaciones entre roles
- Los diagramas de estado 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

