



Tema 1. Introducción a UML

CHRISTOPHER EXPÓSITO IZQUIERDO

AIRAM EXPÓSITO MÁRQUEZ

ISRAEL LÓPEZ PLATA

MARÍA BELÉN MELIÁN BATISTA

JOSÉ MARCOS MORENO VEGA



El Ingeniero de Software



Conocimiento de negocio
Visión económica
Sencillo

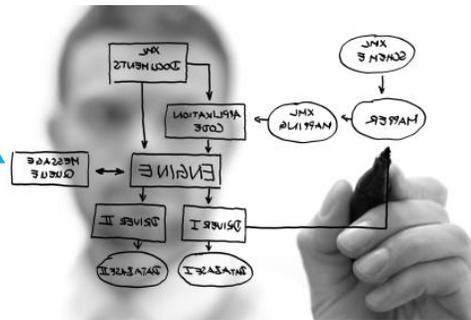


Conocimiento tecnológico
Visión eficiencia
Completo

El Ingeniero de Software



Conocimiento de negocio
Visión económica
Sencillo



Requisitos
Análisis
Diseño



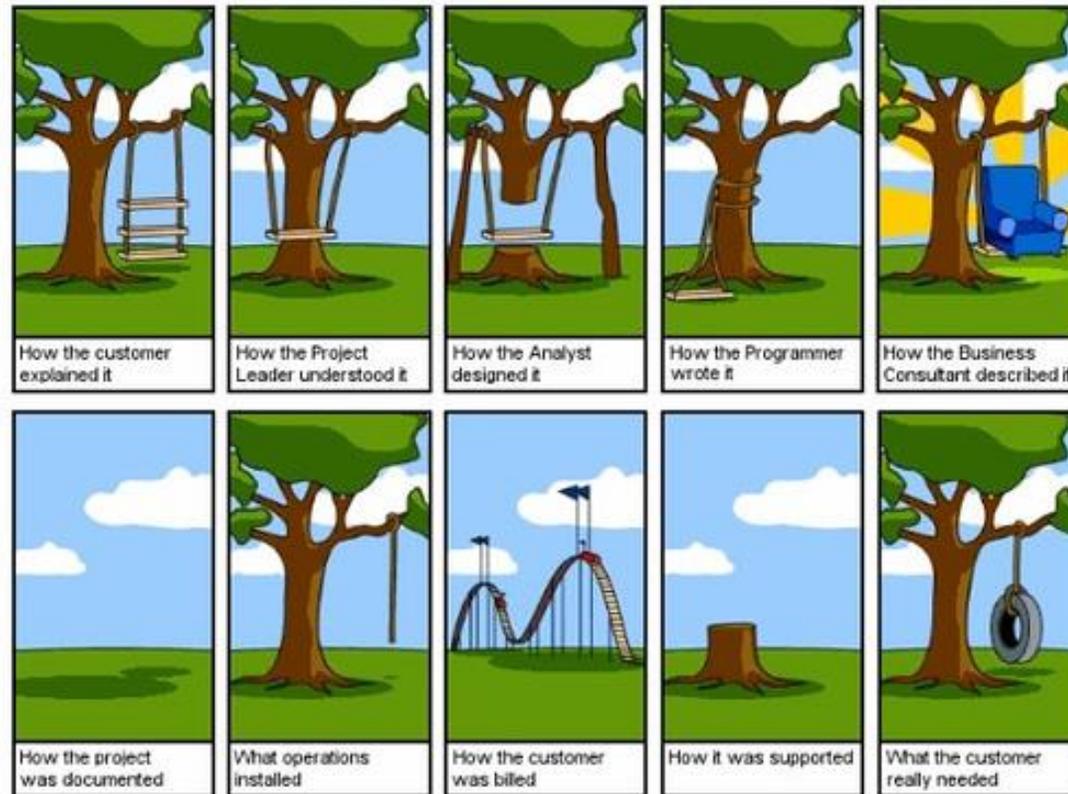
Conocimiento tecnológico
Visión eficiencia
Completo

El Ingeniero de Software

- Desarrollar, mantener y evaluar servicios y sistemas software que:
 - Satisfagan todos los requisitos del usuario
 - Se comporten de forma fiable y eficiente
 - Sean asequibles de desarrollar y mantener
 - Cumplan con las normas de calidad
- Valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades
 - Debe conseguir estos objetivos dentro de las limitaciones de coste, de tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.
- Solucionar problemas de integración en función de las estrategias, estándares y tecnologías disponibles
- Analizar problemas y diseñar, implementar, verificar y documentar sus soluciones
- Identificar, evaluar y gestionar riesgos potenciales

El Ingeniero de Software

Distintos puntos de vista de un proyecto software (*Fuente. www.projectcartoon.com*)



El análisis y modelado

- El objetivo es describir en detalle todas las facetas de un sistema software
 - Utilización
 - Comportamiento de sus procesos
 - Estructura software y hardware
 - Componentes
- El documento debe poder ser interpretado por el equipo de desarrollo del sistema
- Con el fin de universalizar y facilitar la comprensión, se hace uso del modelado

Modelos y diagramas

- **Modelar** consiste en diseñar aplicaciones software antes de su implementación. Es una parte esencial en proyectos de tamaño medio o grande
- Un **modelo** es una abstracción de un elemento del sistema a desarrollar, desde un punto de vista determinado y, por lo tanto, con un nivel de detalle específico
- Un **diagrama** consiste en una representación gráfica de una colección de modelos para representar una parte específica del sistema. Se suele representar como un grafo

Modelos y diagramas

- Un proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés



Grano de detalle

- Cada modelo es completo desde el punto de vista del usuario al que va dirigido

Modelos y diagramas

- Con el modelado del sistema podemos definir todos sus requisitos de forma totalmente a su implementación
- Esto permite tener una visión global del sistema antes de iniciar su desarrollo, lo que implica:
 - Tener unos requisitos bien definidos
 - Reducir problemas en el proceso de implementación
 - Tener una visión global del sistema en una primera fase de su desarrollo
- Para poder modelar un sistema y conducir el proceso de desarrollo, es necesario el uso de una solución software

UML (Unified Modeling Language)



- *UML es un lenguaje de propósito general que ayuda a especificar, visualizar y documentar modelos de sistemas software, incluido su estructura y diseño, de tal forma que se unifiquen todos sus requerimientos (Definición oficial de UML)*
- El objetivo principal de UML es estandarizar el modelado de sistemas software
- Proporciona vocabulario y reglas para combinar y construir representaciones, modelos conceptuales y físicos del sistema
- Permite representar varios modelos, combinando notaciones específicas de cada uno
- *El 80% de los problemas pueden modelarse usando alrededor del 20% de UML (Grady Booch)*

Historia de UML

- En un inicio, varios autores utilizan lenguajes de modelado (Booch, OMT, etc.)
- En 1995 se unifican en la versión 0.8 de UML
- En 1997 UML se modifica con la experiencia obtenida en su utilización en empresas, lo que permite la creación de la versión **1.0**
- En 1998, UML se aprueba por el OMG (Object Management Group), donde pasa a englobarse desde ese momento

Historia de UML

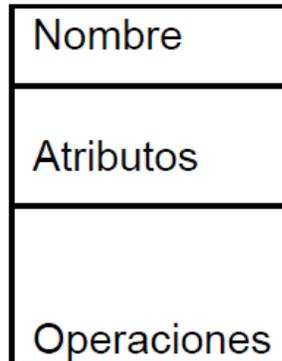
Versión	Fecha	URL
2.5	Junio 2015	http://www.omg.org/spec/UML/2.5
2.4.1	Agosto 2011	http://www.omg.org/spec/UML/2.4.1
2.4	Marzo 2011	http://www.omg.org/spec/UML/2.4
2.3	Mayo 2010	http://www.omg.org/spec/UML/2.3
2.2	Febrero 2009	http://www.omg.org/spec/UML/2.2
2.1.2	Noviembre 2007	http://www.omg.org/spec/UML/2.1.2
2.1.1	Agosto 2007	http://www.omg.org/spec/UML/2.1.1
2.0	Julio 2005	http://www.omg.org/spec/UML/2.0
1.5	Marzo 2003	http://www.omg.org/spec/UML/1.5
1.4	Septiembre 2001	http://www.omg.org/spec/UML/1.4
1.3	Marzo 2000	http://www.omg.org/spec/UML/1.3

Elementos de UML

- Elementos estructurales
 - Partes estáticas del modelo
 - Representan elementos conceptuales, físicos o materiales
 - Clases, objetos, interfaces, casos de uso, actor, etc.
- Elementos de comportamiento
 - Describen el funcionamiento de un sistema
 - Interacciones, máquinas de estado
- Elementos de agrupación
 - Permiten agrupar otros elementos del modelo
 - Componentes, paquetes, nodos
- Elementos de anotación
 - Sirven para realizar anotaciones extra
 - Notas

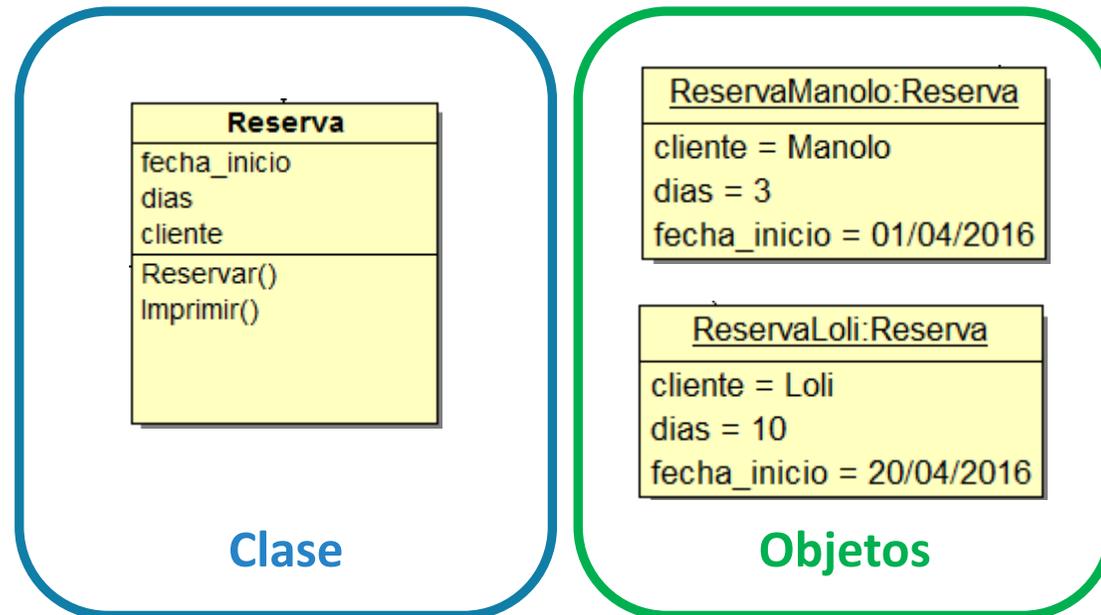
Elementos estructurales. Clases

- Una clase es una definición de un modelo abstracto de datos, que incluye todos los atributos y métodos del elemento a modelar
- Engloba el estado y el comportamiento de un elemento del sistema a modelar
- Punto de vista de la implementación



Elementos estructurales. Objetos

- Instancia de una clase
- En UML se representa por un rectángulo con el nombre del objeto subrayado. El estado del objeto se representa por los valores del atributo

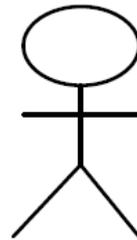


Elementos estructurales. Objetos

- Cada objeto posee un identificador único llamado **Oid (Object Identifier)**. Posee las siguientes características:
 - Identificador único y global para cada objeto dentro del sistema
 - Es determinado en el momento de creación del objeto
 - Es independiente de la localización física del objeto
 - Es independiente de las propiedades del objeto
 - No cambia durante la vida del objeto
 - No se reutiliza nunca, aunque el objeto deje de existir

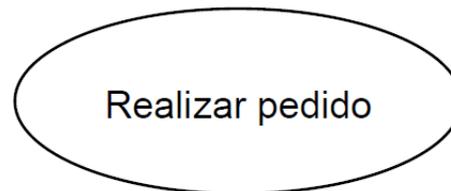
Elementos estructurales

- **Actor.** Conjunto de roles que desempeñan los usuarios del sistema. Un actor no tiene por qué ser un ser humano, ya que puede representar a otro sistema en un proceso de interacción



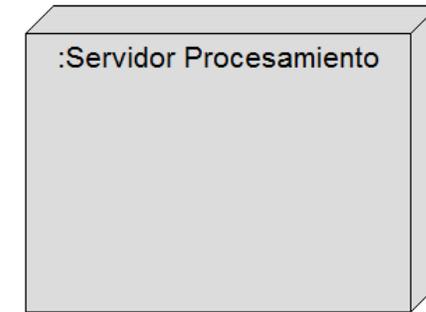
Estudiante

- **Caso de uso.** Es una descripción de un conjunto de secuencias de acciones que el sistema ejecuta y produce un resultado observable para un actor. Representa una funcionalidad del sistema

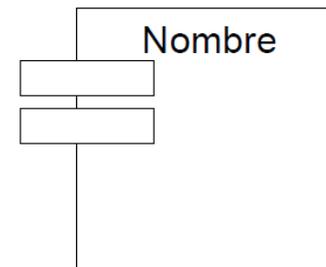


Elementos estructurales

- **Nodo.** Es un elemento físico que representa un recurso del sistema

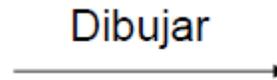


- **Componente.** Elemento estructural software del sistema que representa una parte autónoma del mismo. Empaqueta un conjunto de funcionalidades o datos utilizados en la implementación (tablas, ficheros, librerías, etc.)

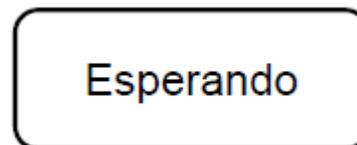


Elementos de comportamiento

- **Interacción.** Conjunto de mensajes entre elementos de los diferentes diagramas. El propósito del mensaje cambia según el contexto en el que se muestre

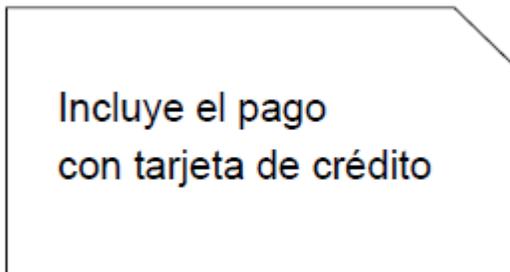


- **Estado.** Situación en un espacio y un tiempo determinado que tiene un elemento del sistema. Sirve para especificar el comportamiento de dicho elemento



Elementos de anotación

- **Nota:** Sirve para anotar comentarios en los modelos con la finalidad de describir, clarificar y hacer observaciones



Diagramas UML

- Diagramas Estructurales
 - Diagrama de Casos de Uso
 - Diagrama de Clases
 - Diagrama de Objetos
- Diagramas de Comportamiento
 - Diagrama de Estados
 - Diagrama de Actividad
- Diagramas de Interacción
 - Diagrama de Secuencia
 - Diagrama de Colaboración
- Diagramas de Implementación
 - Diagrama de Componentes
 - Diagrama de Despliegue/Distribución

Estáticos

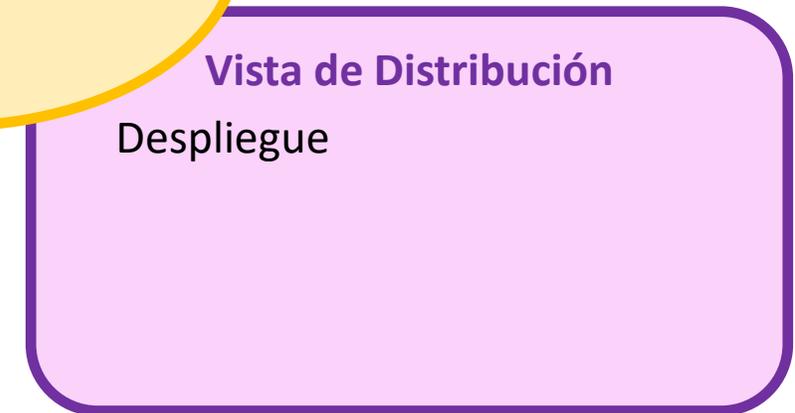
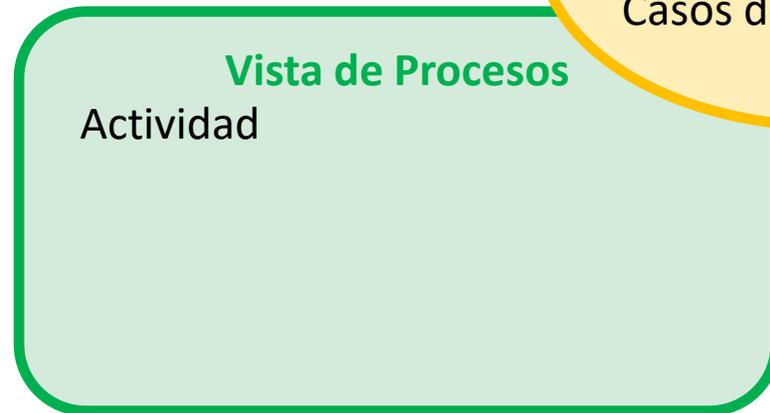
Casos de Uso
Clases
Objetos
Componentes
Despliegue/Distribución

Dinámicos

Estados
Actividad
Secuencia
Colaboración

4+1. Vistas de Krutchen

Casos de Uso
Clases
Objetos
Estados
Actividad
Secuencia
Colaboración
Componentes
Paquetes
Despliegue



Resto de temas. Objetivo

- Explicación de cada uno de los diagramas a través de su uso y ejemplos
- Conocer la utilidad del diagrama y por que se hace así
- Ejemplos prácticos de cada uno de los diagramas