

Tema 3: Diseño



A. Goñi, J. Ibáñez, J. Iturrioz, J.A. Vadillo



OCW
2013



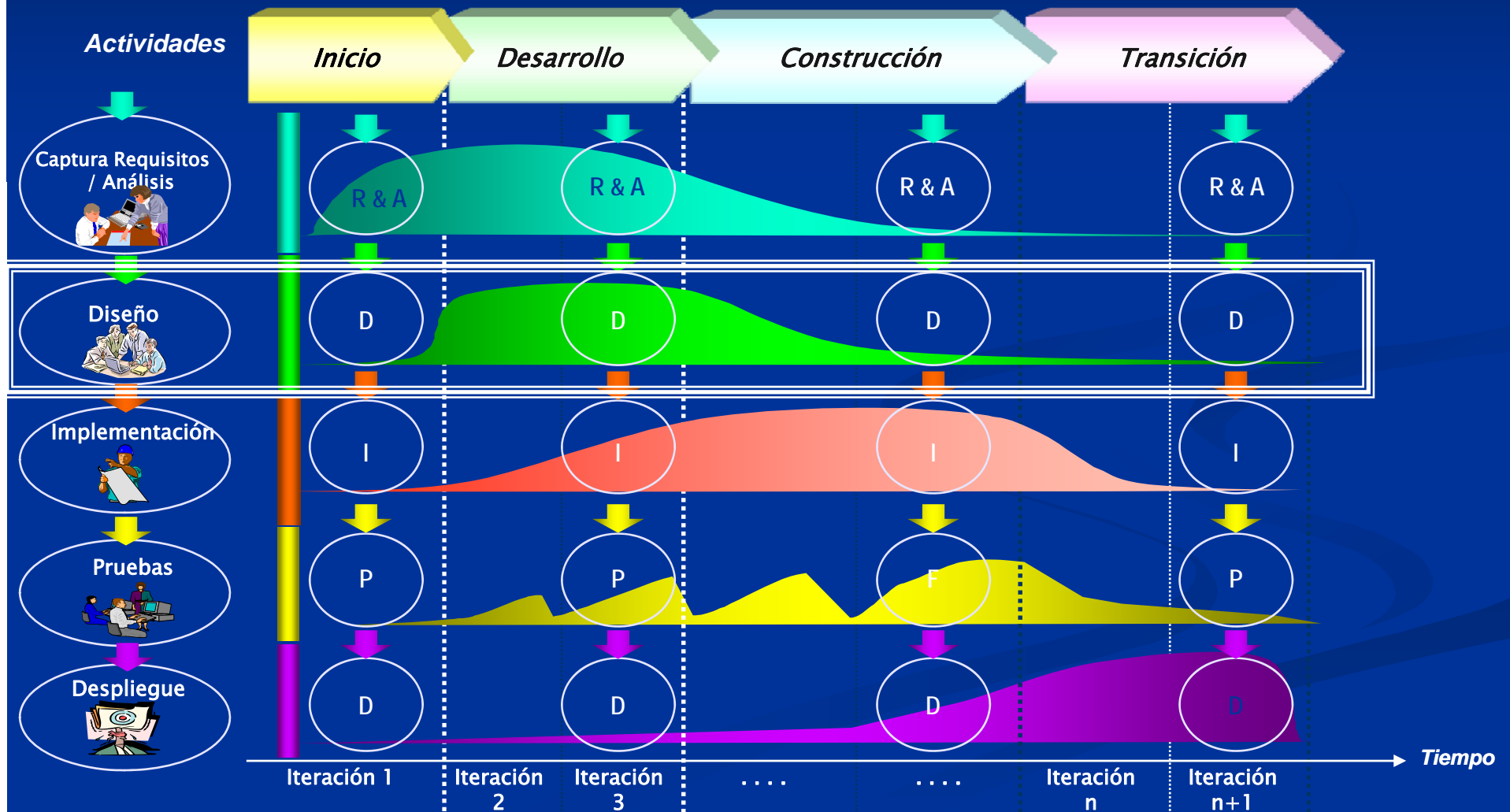
Índice

- 1.- Introducción
 - El rol del diseño en el CV
 - Artefactos a conseguir durante el flujo de trabajo del diseño
 - Decisiones de diseño tomadas en nuestro caso de estudio
- 2 - Diagramas de secuencia UML
- 3.- Realización de diagramas de secuencia
 - 3.1.- Capas lógicas en el software: presentación, lógica del negocio y acceso a datos
 - 3.2.- Patrones de responsabilidad GRASP
- 4 - Arquitecturas Software de varios niveles

1. Introducción

- Objetivo del diseño
 - Encontrar la forma (o solución) del sistema que cumpla con todos los requisitos (funcionales y no funcionales)
 - REQUISITOS: Do the right things / Haz lo correcto
 - DISEÑO: Do the things right / Hazlo correcto
- Durante el diseño se escogen las herramientas: LP, SO, SGBD, GUI, concurrencia, distribución, componentes,...
- Se divide el trabajo para que se pueda implementar en paralelo por equipos distintos
 - Definir bien las interfaces, que permita que los equipos se sincronicen
 - Usar una notación común
- Durante el diseño debe crearse una abstracción de la implementación del sistema:
 - El diseño debe ser una buena entrada a la fase de implementación: que implementar sea directo a partir del diseño

1.1.-El rol del diseño en el CV



1.1.- El rol del diseño en el CV

- Diseño
 - Actividad principal en la fase de desarrollo, junto con el análisis
- El diseño ayuda a:
 - Encontrar una arquitectura estable del sistema
 - Crear un anteproyecto del modelo de implementación
- El modelo del diseño obtenido se debe mantener a lo largo de todo el proyecto
 - Esto es, seguramente cambiará algo durante la implementación, y hay que ocuparse en actualizarlo

1.2.- Artefactos a conseguir durante el flujo de trabajo del diseño

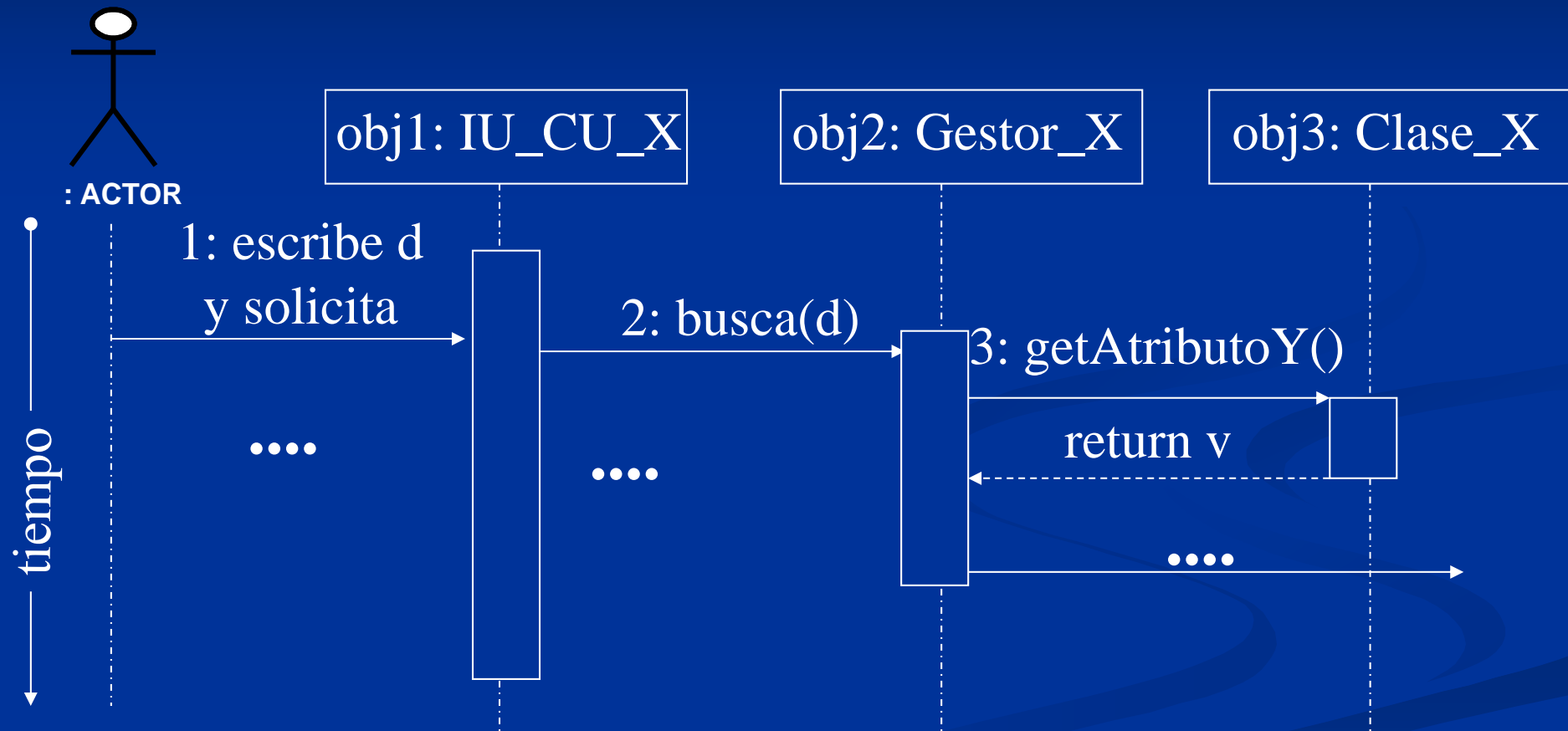
- 1.- Realización/Diseño de los casos de uso – Diagramas de secuencia
- 2.- Clases/Interfaces de diseño
- 3.- Descripción de la arquitectura

Realización/Diseño de los casos de uso

– Diagramas de secuencia

- Es una colaboración que indica cómo se realiza/ejecuta un CU, en términos de las clases de diseño y sus objetos
- Para cada CU habrá que añadir
 - El diagrama de secuencia
 - Flujo de eventos (en el diseño)
 - Requisitos de implementación

Diagrama de Secuencia en UML



Clases / Interfaces de diseño

- Una clase de diseño es una abstracción de una clase (o constructor similar existente) en el LP
 - Operaciones, parámetros, atributos y tipos
 - Visibilidad de atributos y operaciones
 - Asociaciones y agregaciones (aunque luego se implementen añadiendo atributos)
 - Generalizaciones (con la semántica del LP utilizado)

Clases / Interfaces de diseño

- Los métodos se especifican en lenguaje natural o en pseudocódigo
- Pueden especificarse requisitos de implementación de una clase
- Pueden ponerse estereotipos
 - "class module", "form", "user control" en VB
 - "interface" en Java

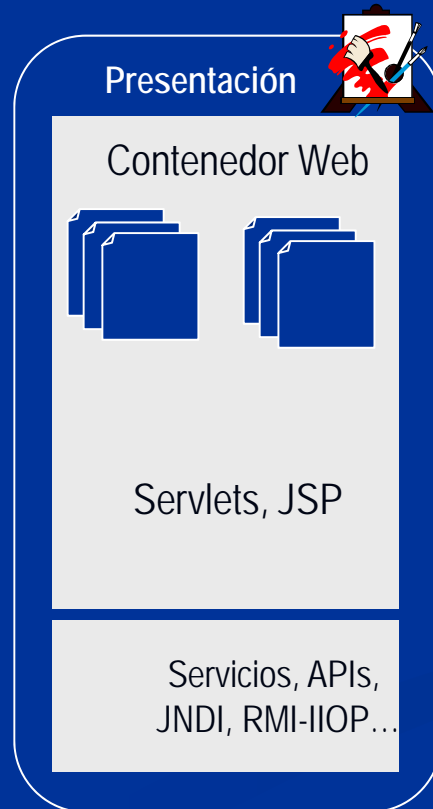
Descripción de la arquitectura

- Separación del sistema en N niveles/capas
- Definición de las relaciones entre niveles cliente, servidor y de datos

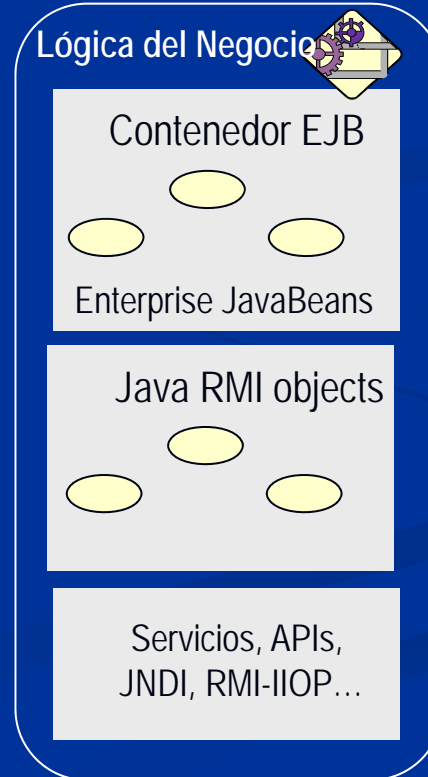
Nivel Cliente



Nivel Servidor



Lógica del Negocio



Nivel de Datos



1.3.- Decisiones de diseño tomadas en nuestro caso de estudio

- El lenguaje de programación a usar es OO (Java)
 - Las clases del modelo del dominio serán clases OO en el diagrama de clases del sistema (con sus atributos)
 - Las asociaciones, generalizaciones, etc. las mantendremos en el diagrama de clases, siendo conscientes de que en la implementación se transformarán en atributos de clases, o en nuevas clases.
 - Al ir construyendo los diagramas de secuencia, SE IRÁN AÑADIENDO nuevos métodos a las clases del diseño.
- Usaremos un sistema de gestión de bases de datos OO (DB4o)
 - Las clases/asociaciones/generalizaciones del modelo del dominio se transformarán directamente en clases con persistencia.
 - Será necesario controlar la persistencia de cada uno de los objetos, tanto al consultar/recuperar como al insertar, modificar o borrar objetos.