



Universidad de Oviedo



## Arquitectura del Software Definiciones



Curso 2019/20

Jose Emilio Labra Gayo

# Esquema

Definiciones de arquitectura básicas

¿Qué es arquitectura del software?

Stakeholders, atributos de calidad, restricciones

Documentación de arquitectura del software

# ¿Qué es arquitectura del software?

“Conjunto de estructuras necesario para razonar sobre un Sistema, que comprenden los elementos de software, las relaciones entre ellos y las propiedades de ambos.”

## Estructura básica del sistema

“Decisiones de diseño principales del sistema”

Si hay que cambiarlas  $\Rightarrow$  Coste elevado

# Proceso de diseñar una arquitectura

Dominio del problema

Dominio de la solución

Objetivos de  
Diseño

Requisitos  
Funcionales

Atributos de  
Calidad

Restricciones

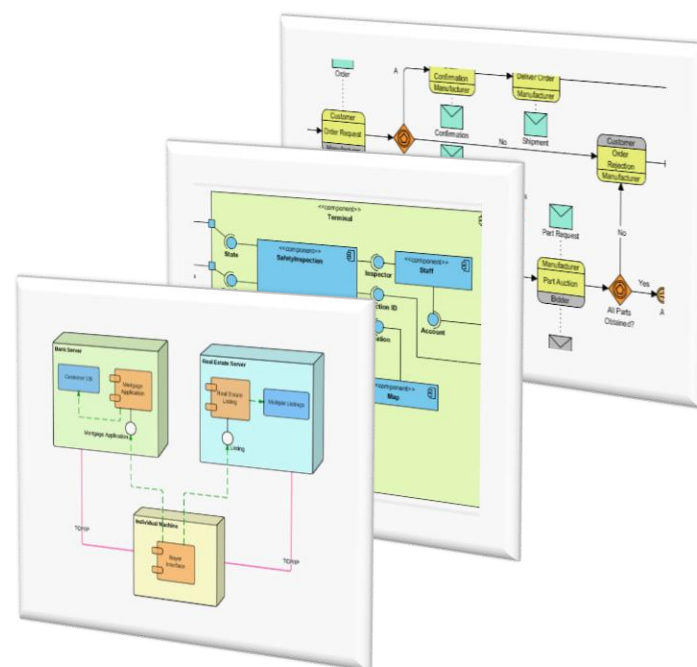
Preocupaciones  
(Concerns)

Entradas



Arquitecto

Actividad de diseño



Diseño de  
La arquitectura  
(salida)

# Motivaciones proceso de arquitectura

## Entradas

Objetivos de diseño

Requisitos funcionales

Atributos de calidad

Restricciones

Preocupaciones

# Objetivos de diseño

Aclarar ***porqué*** se diseña un sistema

Ejemplos:

**Propuesta pre-venta:** diseño rápido de una solución inicial para obtener una estimación

**Sistema a medida** con un tiempo y coste establecido que no puede variar mucho una vez enviado

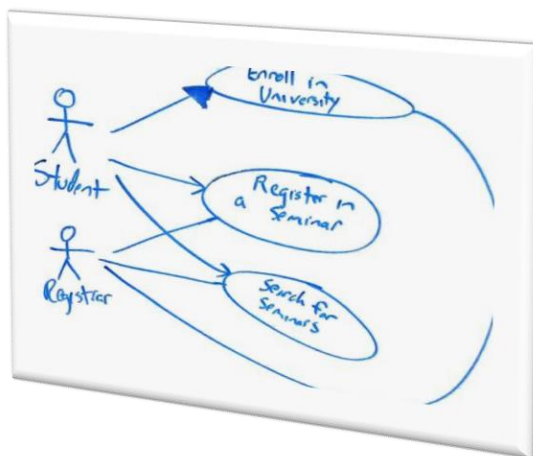
**Incremento Nuevo ó versión** de un sistema que está continuamente evolucionando

# Requisitos funcionales

Funcionalidad que debe soportar los objetivos de negocio

Lista de requisitos como casos de uso o historias de usuario

## Use cases



## User stories

**As a** recruiter  
**I want** to be able to  
manage resumes,  
**so that** I can process  
the resumes from  
job-seekers.

# Atributos de calidad

Características medibles de interés para usuarios o desarrolladores

También conocidos como requisitos no-funcionales

Rendimiento, disponibilidad, modificabilidad, testabilidad,...

También conocidos como –idades (-ities en inglés)

Pueden especificarse mediante escenarios

Técnica estímulo-respuesta

*“Si ocurre un fallo interno durante la operación normal, el sistema reanuda la operación en menos de 30 segundos y no se pierden datos”*

Priorizados por:

El cliente de acuerdo al éxito del sistema

El arquitecto de acuerdo al riesgo técnico

ISO 25010: lista de algunos requisitos no funcionales

Lista: [https://en.wikipedia.org/wiki/List\\_of\\_system\\_quality\\_attributes](https://en.wikipedia.org/wiki/List_of_system_quality_attributes)



# Atributos de calidad

Los atributos de calidad determinan la mayoría de las decisiones de diseño en arquitectura

Si la única preocupación fuese la funcionalidad, un sistema monolítico sería suficiente

Sin embargo, es habitual ver:

- Estructuras redundantes para fiabilidad

- Estructuras concurrentes para rendimiento

- Capas para modificabilidad

...

Priorizados por:

- El cliente de acuerdo al éxito del sistema

- El arquitecto de acuerdo al riesgo técnico

# Restricciones

Restricciones del sistema que vienen impuestas

Muy poco software tiene libertad total

Pueden ser técnicas u organizativas

Pueden surgir del cliente o también de la organización de desarrollo

Limitan las alternativas a considerar para decisiones de diseño particulares

Ejemplos:

Marcos de aplicaciones (*frameworks*)

Lenguajes de programación, ...

Normalmente son tus “amigos”

# Preocupaciones

Decisiones de diseño que deben tomarse aunque no estén enunciadas explícitamente en los objetivos o requisitos

Ejemplos:

- Crear una estructura física o lógica consistente

- Validar campos de entrada

- Gestión de excepciones y logging

- Migración de datos y backup

- Organización del código fuente

- ...

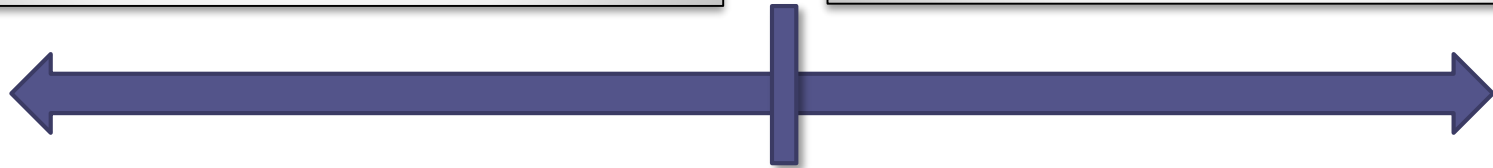
# Arquitectura = solución de compromiso

## Creatividad

Divertido  
Arriesgado  
Puede ofrecer soluciones nuevas  
Puede ser innecesario

## Método

Eficiente en terrenos familiares  
Resultado predecible  
No siempre es lo mejor  
Técnicas de calidad contrastada



Arquitecto

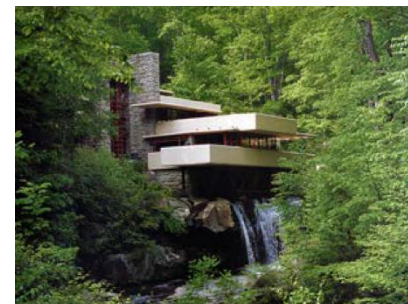


# Tipos de sistemas

Sistemas *greenfield* en dominios nuevos

Ejemplo Google, Whatsapp

Dominios innovadores poco conocidos



Sistemas *greenfield* en dominios maduros

Ejemplo: aplicaciones empresariales tradicionales, aplicaciones móviles, ...

Dominios conocidos, menos innovadores



Dominios *brownfield*

Cambios a sistemas existentes



*Greenfield*: no urbanizado  
*Brownfield*: antigua zona industrial

# Arquitecto del software

La disciplina evoluciona

Arquitecto debe conocer:

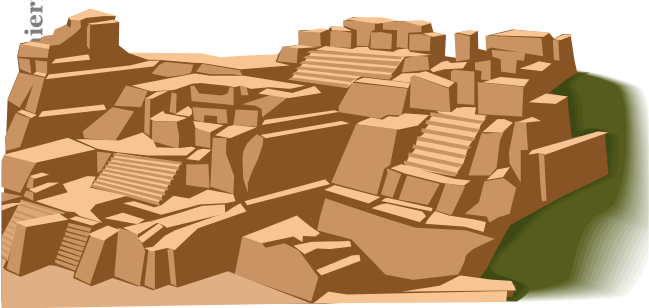
Avances en técnicas de construcción

Estilos y patrones

Mejor herramienta = experiencia (*no silver bullet*)

Experiencia propia

Experiencia de la comunidad



Arquitecto



# Papel del arquitecto de software

