



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 17

Arquitectura de Software

IIC2143 - Ingeniería de Software
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

16 de mayo de 2018

¿Qué es la arquitectura?

- “Técnica y estilo con los que se diseña, proyecta y construye un edificio o un monumento.”



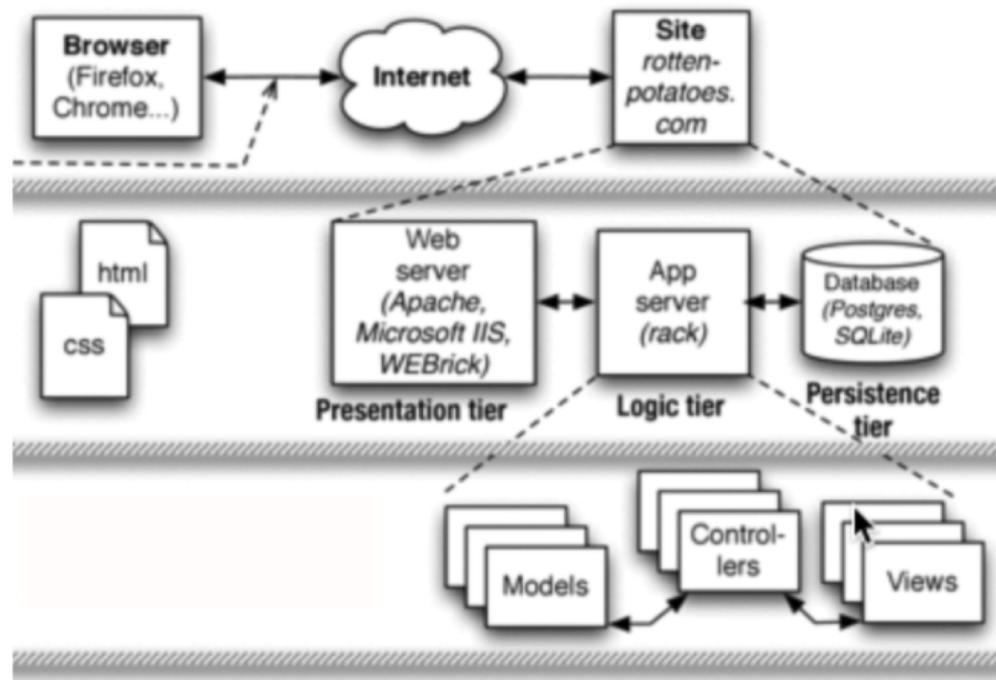
- La arquitectura está definida por los elementos principales de un edificio, su composición y organización.

¿Qué es la arquitectura?

- En el *software* es similar: “conjunto de las principales decisiones de diseño realizadas sobre un sistema”
- Es la estructura de los componentes y sus interacciones
- Sin embargo, tiene diferencias:
 - Las edificaciones son un concepto más intuitivo
 - Solamente con mirar un edificio se puede desprender mucha información de este
 - Un *software* con buena arquitectura se adapta a los cambios, los edificios son estáticos

¿Qué es la arquitectura?

- Sus decisiones son a nivel de sistemas, mientras que el diseño es a nivel de módulos
- Por este motivo es más costosa de modificar



¿Qué es un componente?

- “A software component is an architectural entity that (1) encapsulates a subset os the system’s functionality and/or data, (2) restricts access to that subset via an explicitly defined interface, and (3) has explicitly defined dependencies on its required execution context.”

¿Qué es un componente?

- Partes del contexto encapsuladas por los componentes son:
 - Interfaces expuestas para interactuar con otros componentes
 - Recursos (como archivos o directorios) de los cuales depende el componente
 - Dependencias de *software* para ejecutarse (como lenguajes, sistemas operativos, *drivers*, etc...)
 - Configuraciones de *hardware* para ejecutar el componente

Factores que inciden en la arquitectura

- Requisitos no funcionales

- Seguridad
- Confidencialidad
- Desempeño
- Escalabilidad
- Disponibilidad
- Interoperabilidad
- Portabilidad
- Otros...

Factores que inciden en la arquitectura

- Requisitos no funcionales
- Principios del sistema (consistencia y claridad del código)
- Restricciones del proyecto

Restricciones en la arquitectura

- Tiempo y presupuesto
- Tecnológicas
 - lista de tecnologías aprobadas
 - sistemas existentes e interoperabilidad
 - uso de sistemas *open source*
 - madurez de tecnologías
 - relación con los proveedores
- Relativas a personas
 - tamaño y experiencia del equipo
 - disponibilidad de especialistas
 - ¿quién dará mantenición al sistema?

Patrones arquitectónicos

- Similar a patrones de diseño:

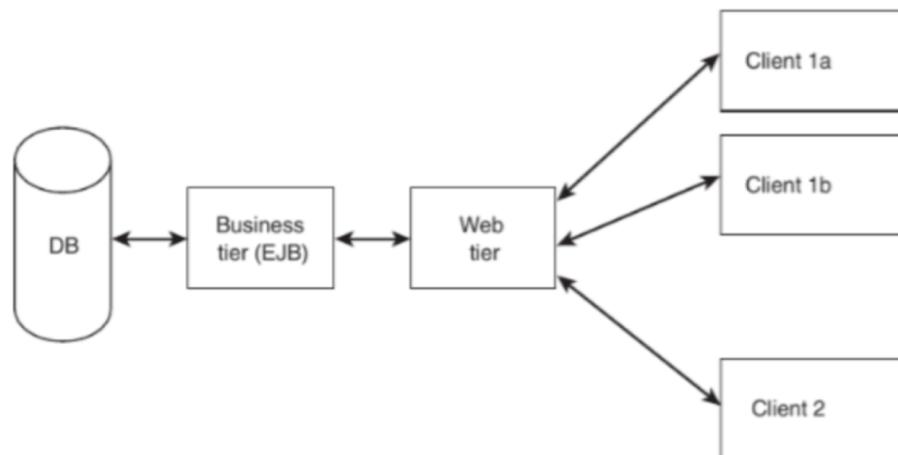
“An architectural pattern is a named collection of architectural design decisions that are applicable to a recurring design problem, parametrized to account for different software development contexts in which that problem appears.”

Patrones arquitectónicos

Client – Server

- Dos grandes componentes: cliente y servidor
- Servidor recibe y procesa solicitudes de clientes

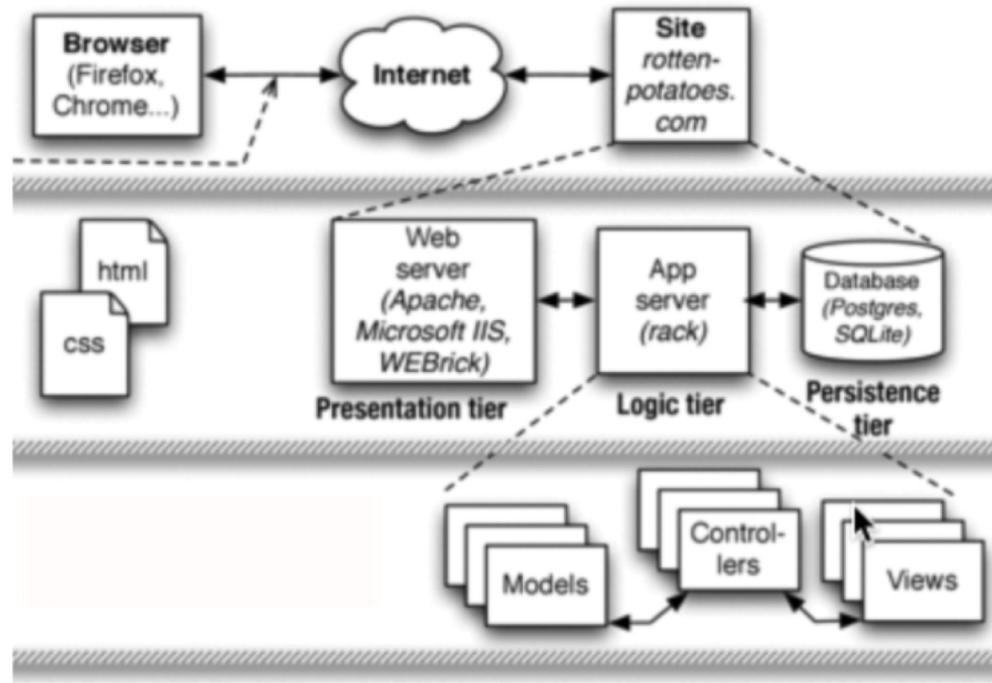
Arquitectura de referencia J2EE



Patrones arquitectónicos

Client – Server

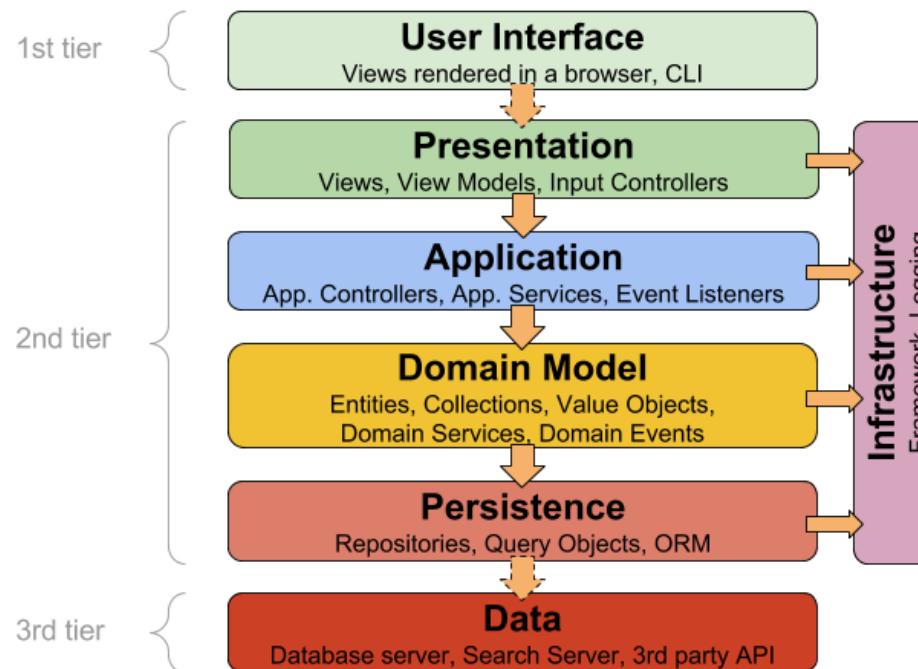
- Dos grandes componentes: cliente y servidor
- Servidor recibe y procesa solicitudes de clientes



Patrones arquitectónicos

Layered (en capas)

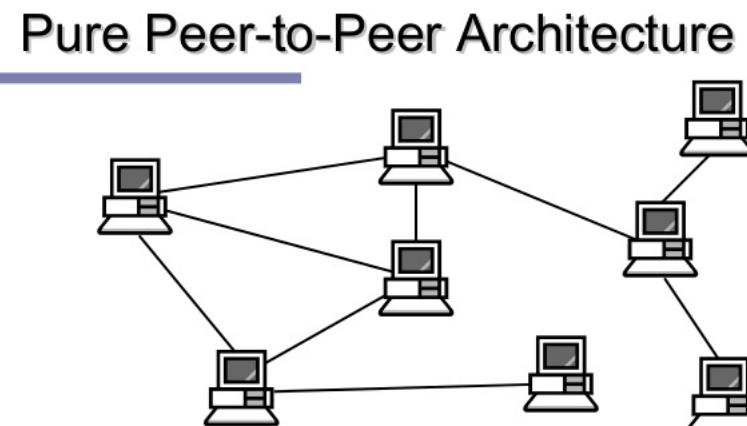
- Una capa es una colección coherente de componentes



Patrones arquitectónicos

Peer-to-peer

- Red de nodos conectados con las mismas capacidades y responsabilidades
- Permite el intercambio directo de información entre nodos



- No central server
- Clients connected to one or more peers
- Resilient
- Large #messages

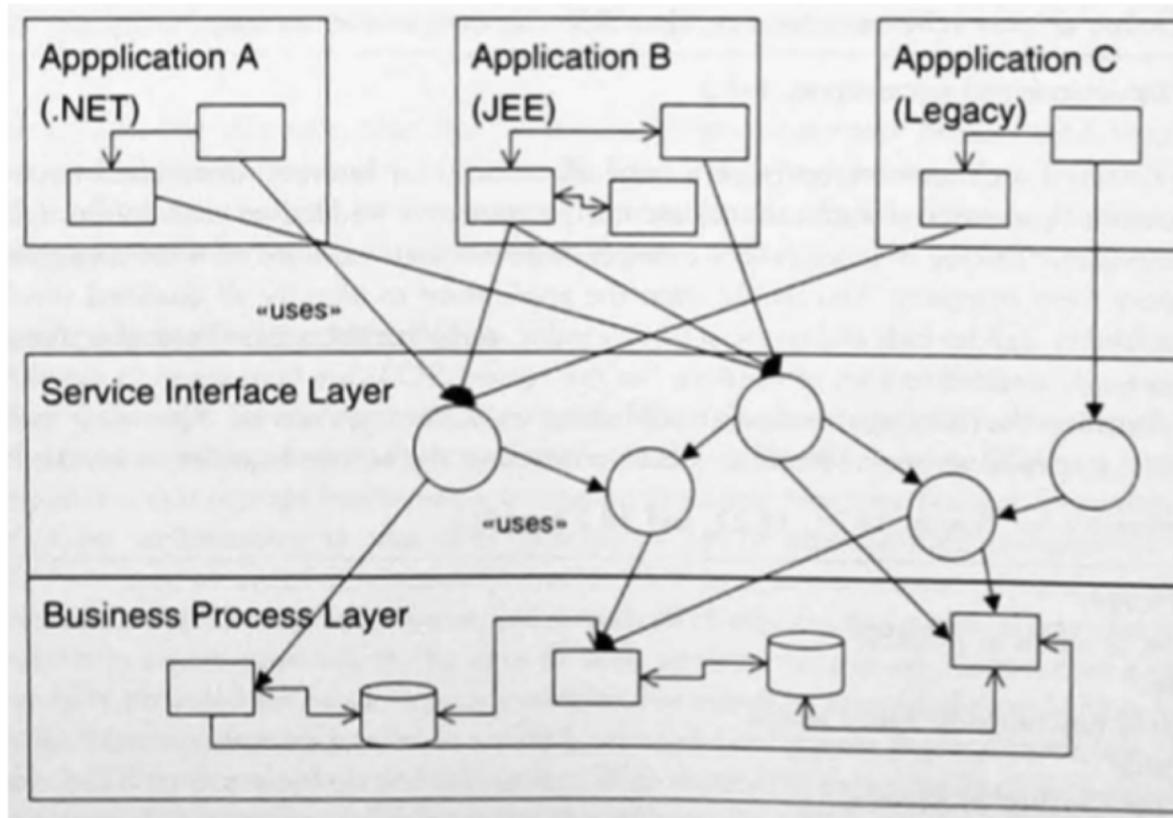
Patrones arquitectónicos

Service Oriented Architecture (SOA)

- Sistema se concibe como una combinación de servicios
- Servicios proveen funcionalidades a través de interfaces específicas
- Pueden ser combinados en forma dinámica
- Altamente relacionado con los servicios y APIs web

Patrones arquitectónicos

Service Oriented Architecture (SOA)

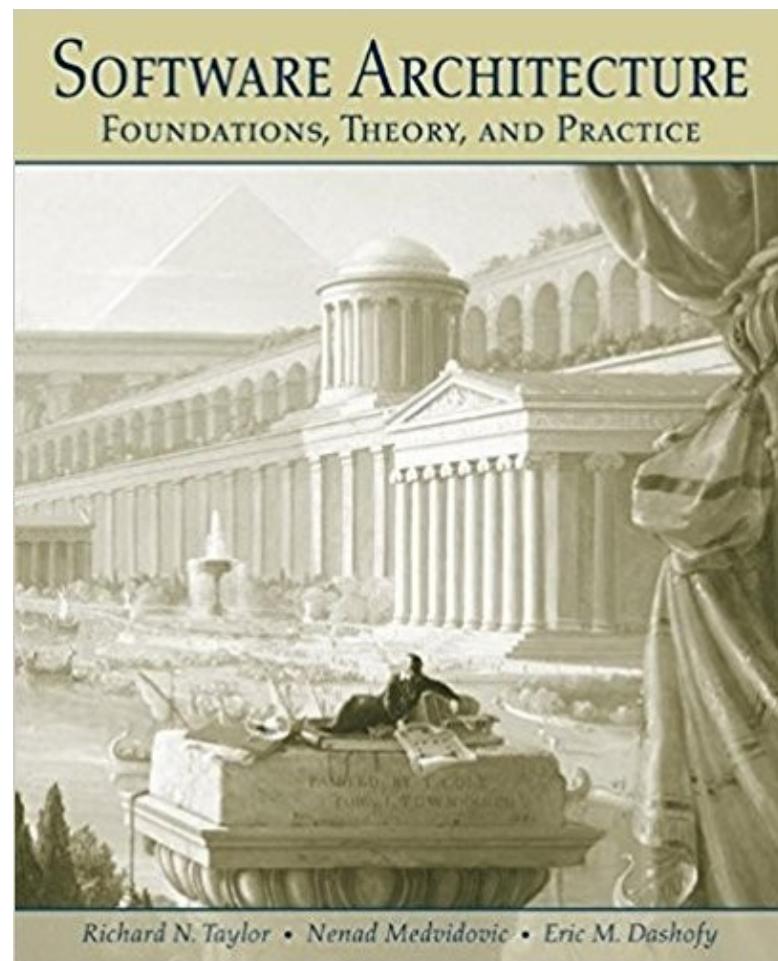


Componentes como Servicios

- Unidades de *software* que pueden ser reemplazadas o mejoradas en forma independiente
 - Ejemplos son módulos o librerías
- Un servicio es una componente que no corre en el mismo proceso o sistema, y se comunica a través de un protocolo de red (como *http* o *rpc*)

Componentes como Servicios

- Ventajas:
 - *Deployable* de manera independiente
 - Interfaces explícitas
 - Heterogeneidad de tecnologías
 - Resiliencias
 - Escalamiento flexible





Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 17

Arquitectura de Software

IIC2143 - Ingeniería de Software
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

16 de mayo de 2018