

Diseño y Arquitectura de Solución



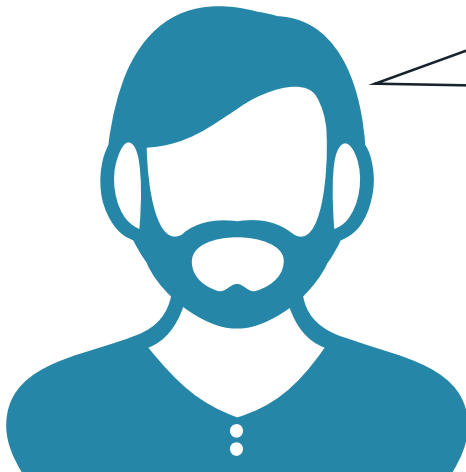
CE-4101 Especificación y diseño de software

Definición de arquitectura de software



Qué es arquitectura de software?

Definición



"The architecture of a software system is the shape given to that system by those who build it. The form of that shape is in the division of that system into components, the arrangement of those components, and the ways in which those components communicate with each other."

[Martin, Robert C. Clean Architecture]

Otras definiciones de arquitectura de software

Otras Definiciones

"The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution". [IEEE 1471 2000]

La arquitectura del software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software., las propiedades visibles externamente de esos componentes y las relaciones entre ellos. [Bass, Clement & Kazman]

Objetivo de la arquitectura de software?

Para qué invertimos en hacer un diseño y arquitectura?

"The goal of software architecture is to minimize the human resources required to build and maintain the required system."

"The measure of design **quality** is simply the measure of the effort required to meet the needs of the customer. If that effort is low, and stays low throughout the lifetime of the system, the design is good. If that effort grows with each new release, the design is bad. It's as simple as that.

[Martin, Robert C. Clean Architecture]



Requerimientos de calidad

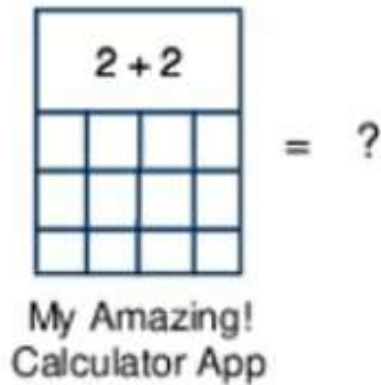
Nos recuerdan?

- ✓ Necesitamos una solución que escale fácilmente la cantidad de usuarios para pasar de Z a Y.
- ✓ Necesitamos que los tiempos de respuesta estén por debajo de X cantidad de segundos.
- ✓ Necesitamos que podamos hacer cambios al funcionamiento sin tener que evitar que los usuarios ingresen a la aplicación.
- ✓ Necesitamos que el nivel de servicio de la aplicación este en el rango de W%
- ✓ Debe poder hacer el procesamiento utilizando un hardware con estas características

Arquitectura y su relacion con los requerimientos no funcionales

7

Atributos de calidad



What if...

It takes 10 minutes to get an answer?

Sometimes you get the wrong answer?

You have to rewrite it to enable subtraction?



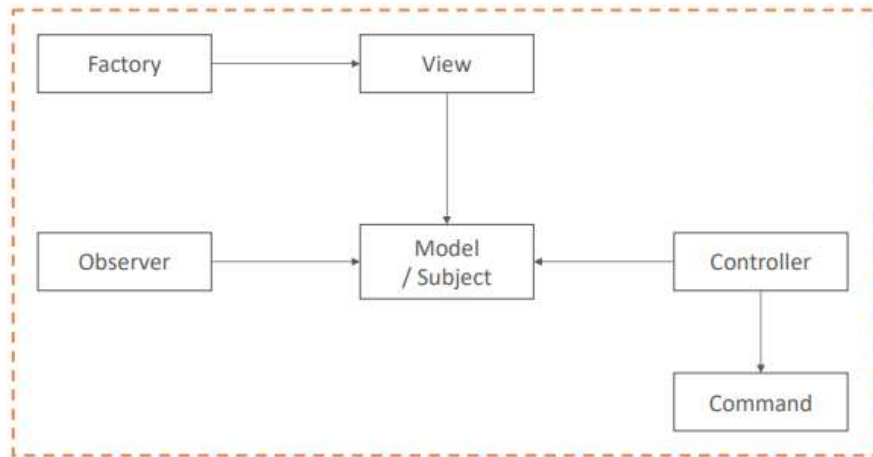
Por qué?

8

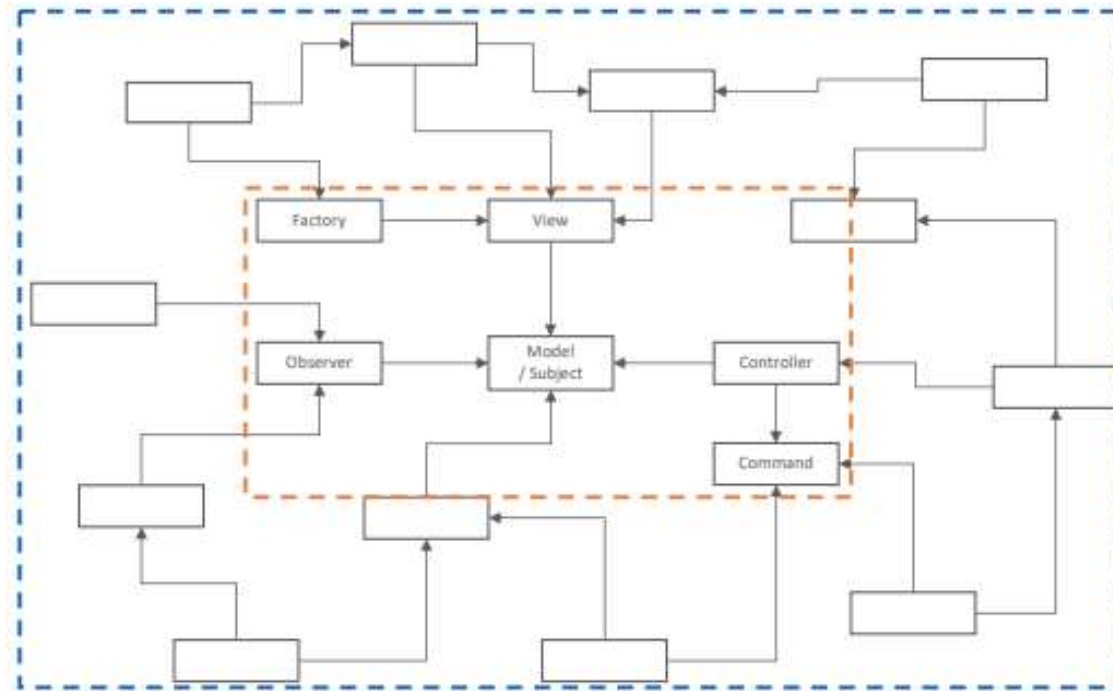


- ✓ La arquitectura y el diseño representan de manera temprana decisiones que se quieren implementar
- ✓ Ayuda en la comunicación con los stakeholders
- ✓ Dicta la estructura organizacional de la solución y permite que multiples equipos trabajen en diferentes componentes
- ✓ Permite trabajar en predecir los costos, calendarios de implementación
- ✓ Ayuda a determinar cómo los atributos de calidad serán habilitados o inhibidos

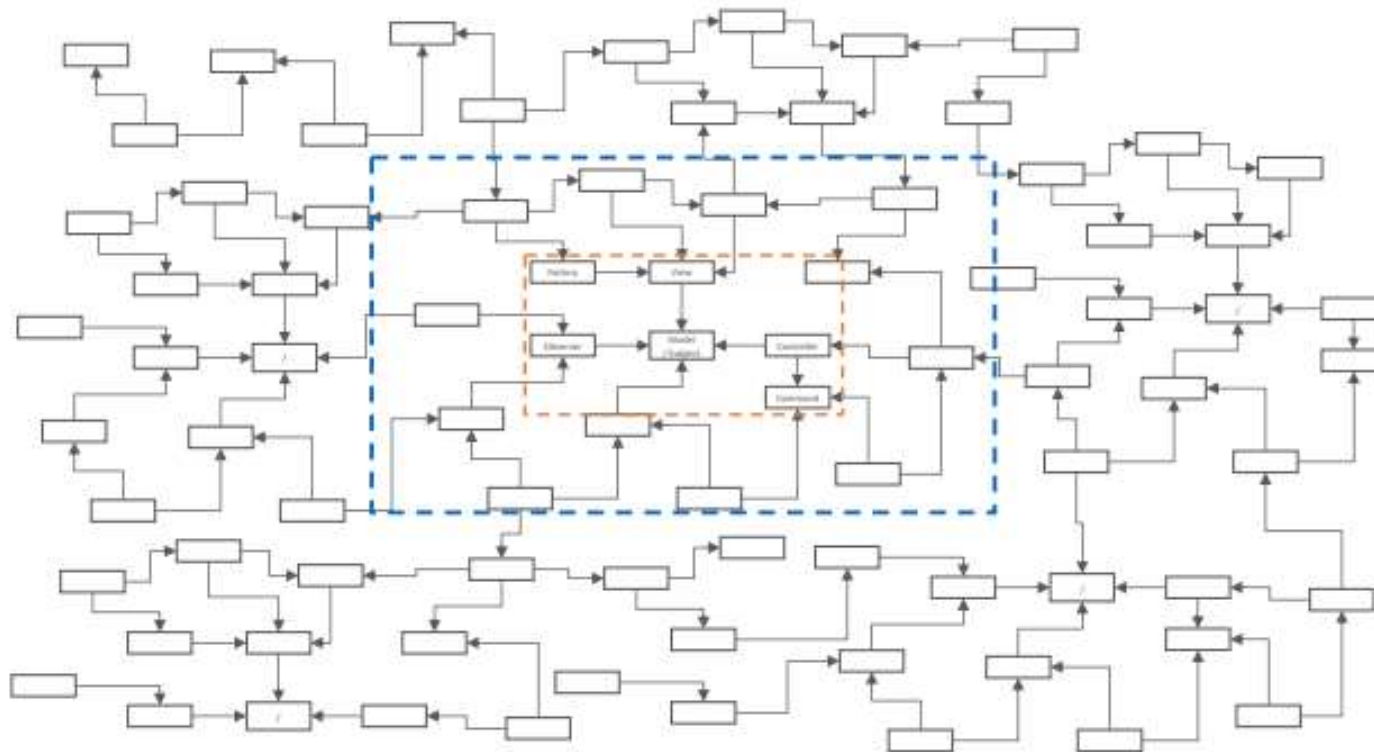
Patrón de diseño



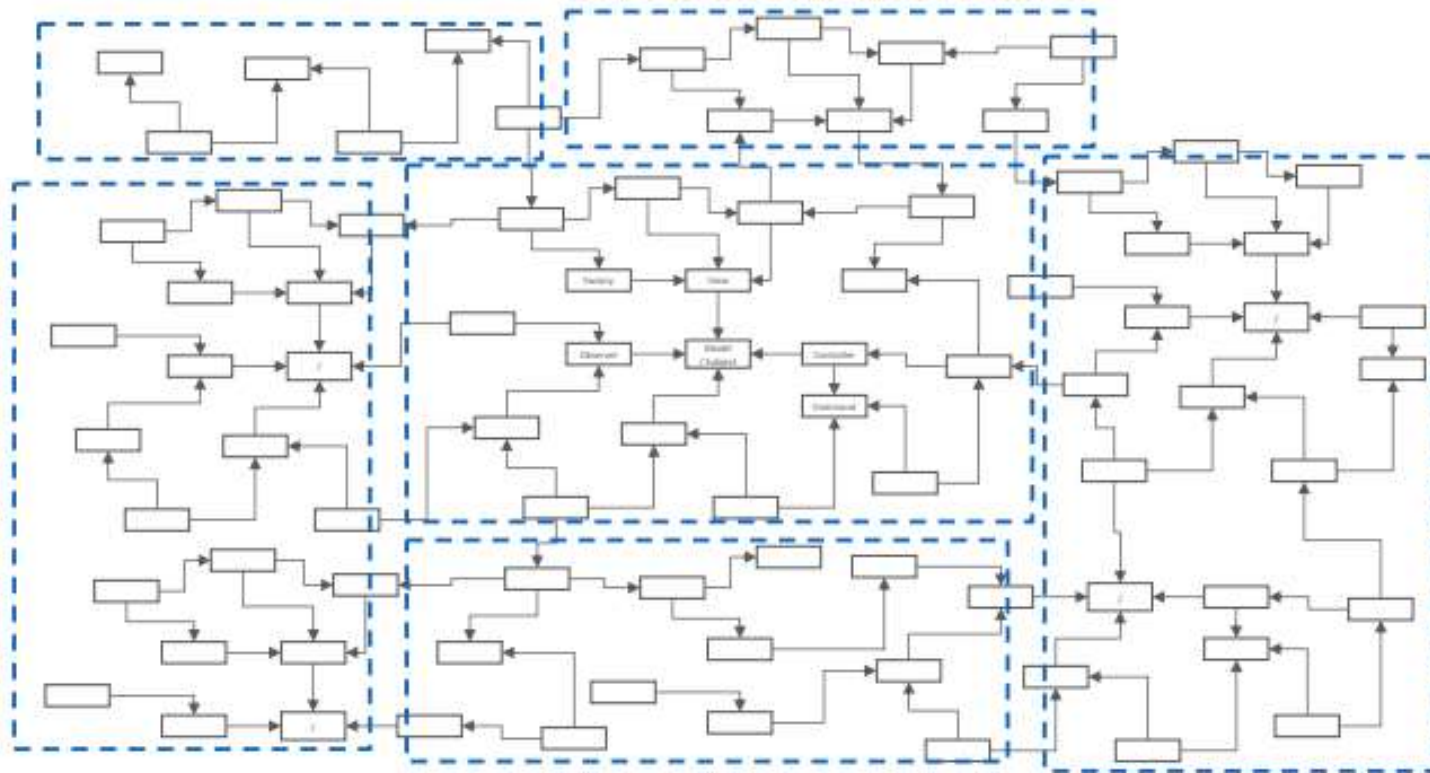
Patrón de diseño



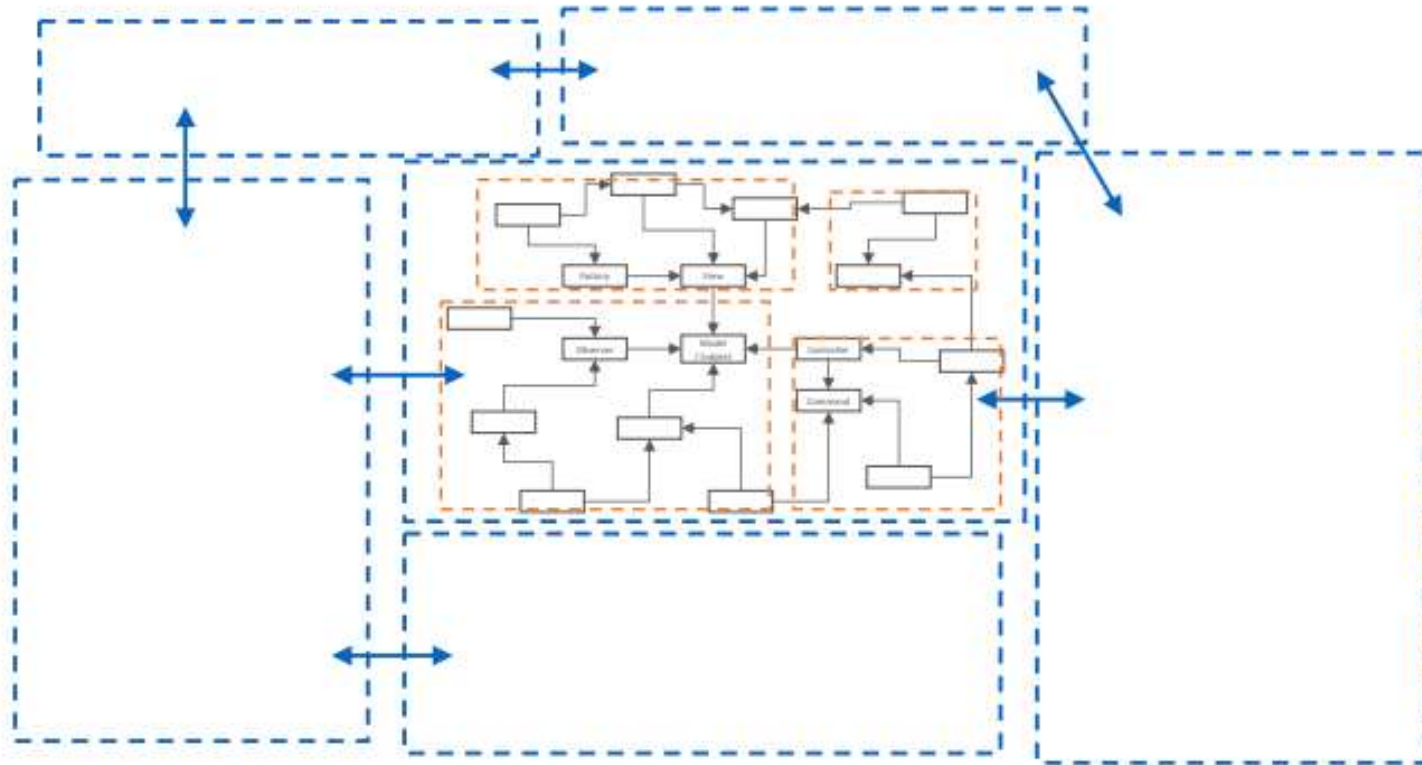
Patrón de diseño



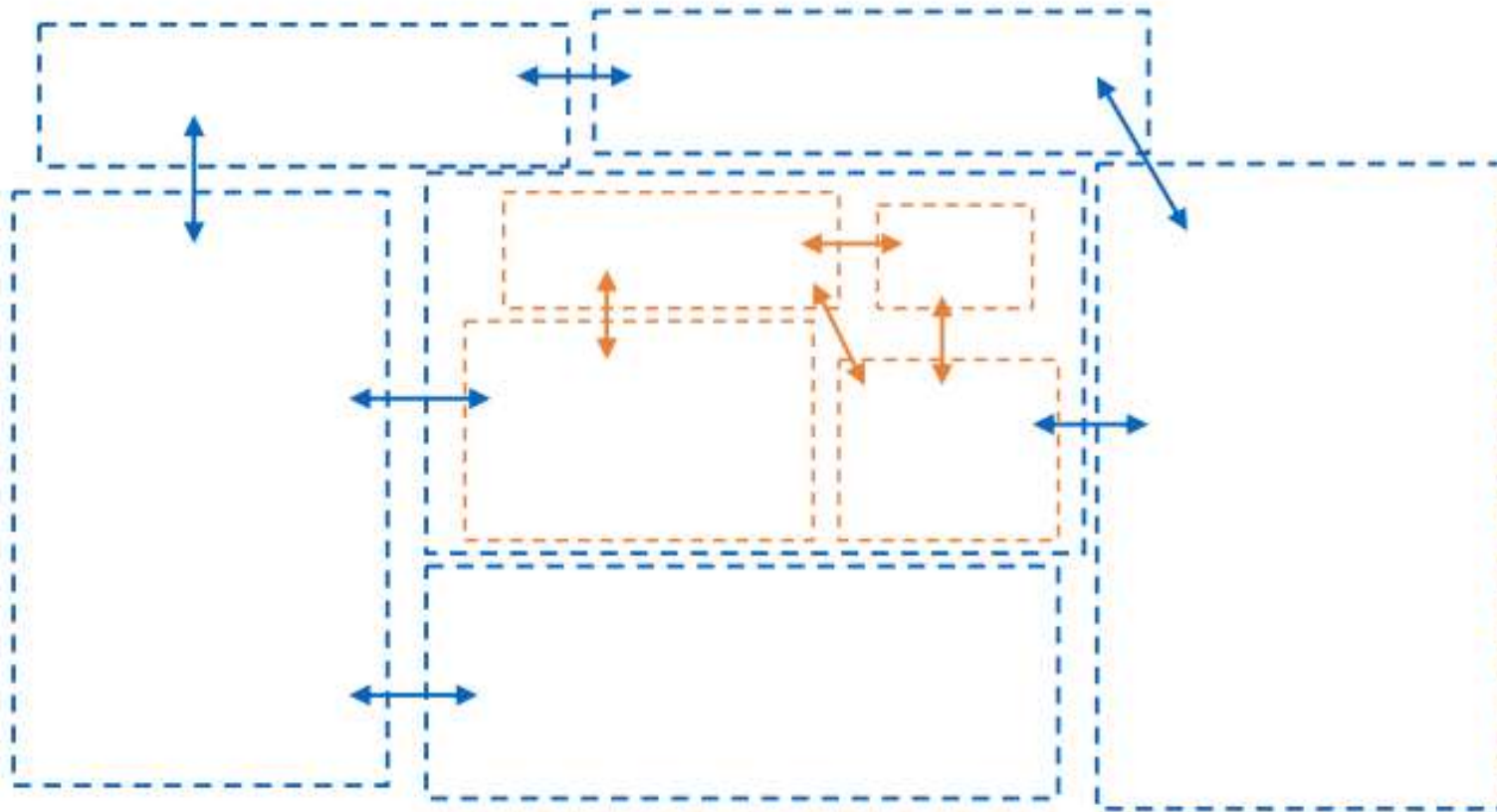
Arquitectura de Aplicación



Arquitectura de Aplicación



Arquitectura de Aplicación



Objetivo de la arquitectura de software?

Atributos



Análisis de indicadores para una empresa desarrolladora

16

Crecimiento del equipo de desarrollo



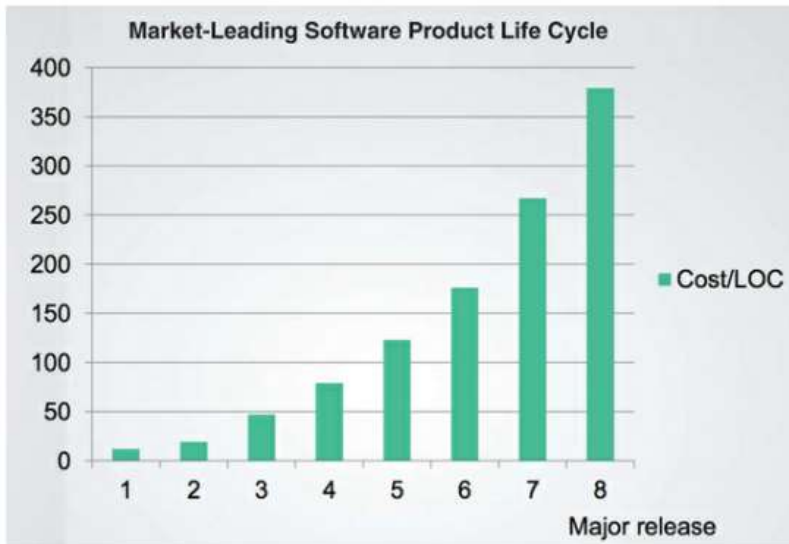
Productividad en KLOC para el equipo de desarrollo



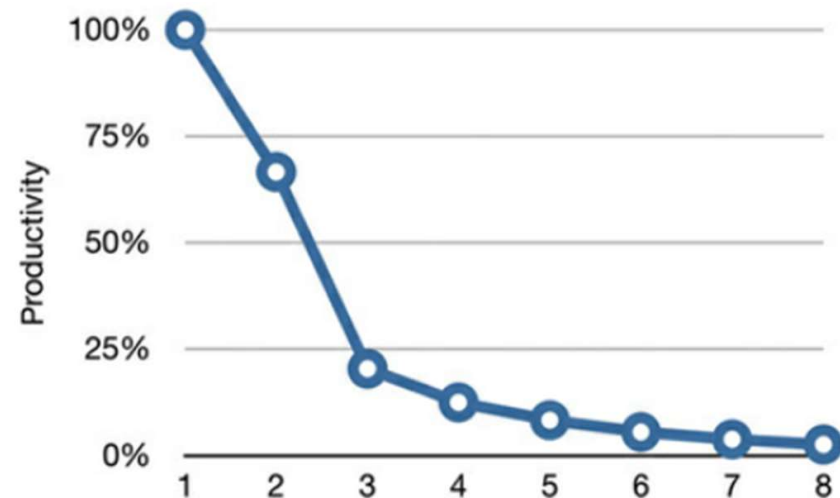
Análisis de indicadores para una empresa desarrolladora

17

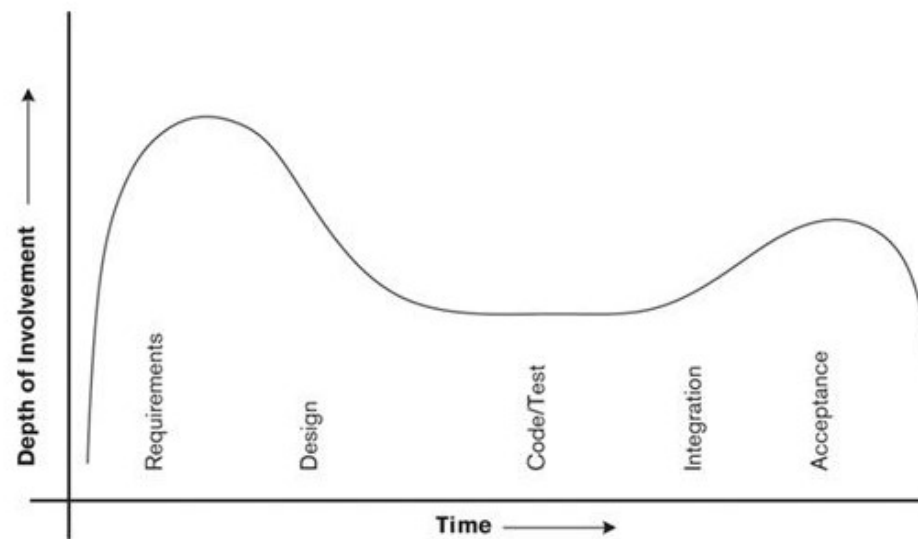
Costo por LOC para el equipo de desarrollo



Productividad por release para el equipo de desarrollo



La arquitectura en el ciclo de vida del proceso de desarrollo de software



Rozanski, Nick; Woods, Eoin.. (2012). Software System Architecture: Working with Stakeholders using viewpoints and Perspectives

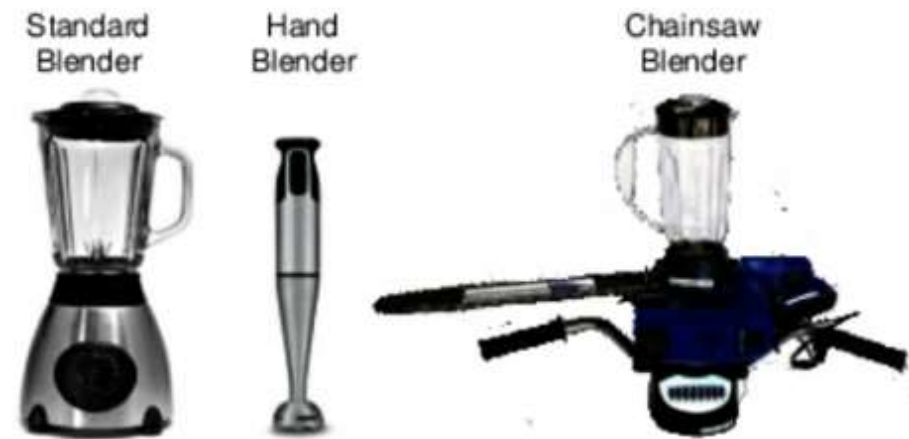
Impacto de la arquitectura de software



Escoger una arquitectura

Definición

- ✓ Cada sistema tiene una arquitectura.
- ✓ Realizar un proceso activo en la definición de esa arquitectura mejora las posibilidades de que cumpla nuestras necesidades
- ✓ Las decisiones van a estar llenas de intercambios de dejar cosas buenas para evitar cosas malas



	Standard Blender	Hand Blender	Chainsaw Blender
Cleanability	Neutral	Positive	Neutral
Counter top-ability	Positive	Negative	Strongly Negative
Quietness	Neutral	Positive	Strongly Negative
Power	Neutral	Negative	Strongly Positive
Portability	Strongly Negative	Positive	Strongly Positive
Safety	Neutral	Neutral	Negative

Impacto de la arquitectura en un caso real

Rackspace



“Rackspace® (NYSE: RAX) is the global leader in hybrid cloud and founder of OpenStack®, the open-source operating system for the cloud.”

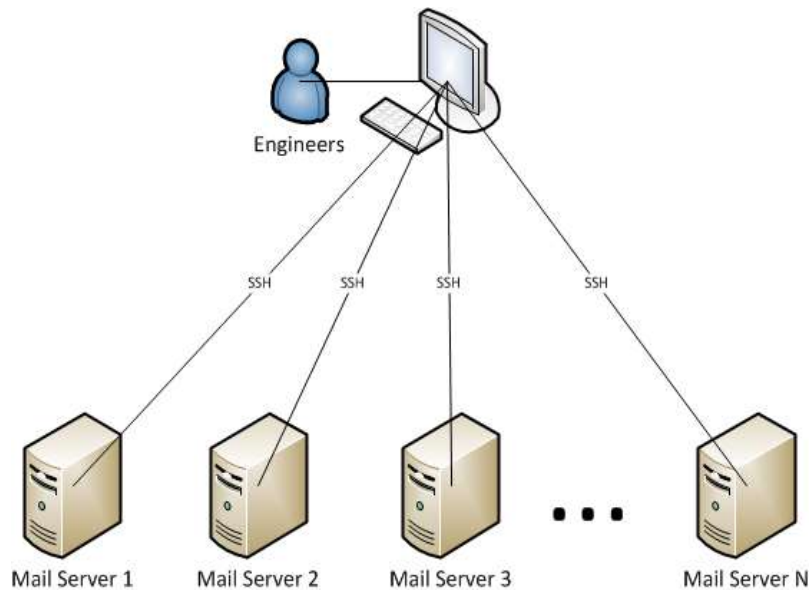
Datos importantes:

- 10+ años
- 80.000 + Clientes
- Rackspace tiene más de 500+ dispositivos y 7 data centers
- Parte de los servicios incluyen hosting de servicios de correo
- El sistema almacena mas de 800 millones de objetos
- Los logs de datos registran cerca de 180 GB cada día
- Utilizan Noteworthy y MS Exchange como productos para sus servidores de correo
- Necesidad: compañía enfocada en su cliente . Desea utilizar la información que generaban en sus logs para mejorar el servicio a sus clientes reduciendo los tiempos de respuesta a la hora de hacer troubleshoot

Impacto de la arquitectura en un caso real (cont)

Rackspace

Rackspace v1



Archivos almacenados en archivos planos en los discos locales de cada servidor de correo mantenidos en el servidor por 14 días

El equipo técnico se conectaba por SSH a cada servidor de correo a buscar la información

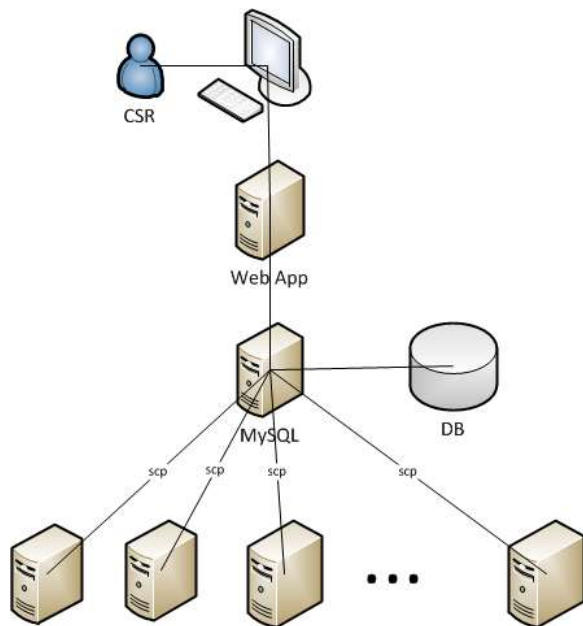
Impacto:

- Requiere buscar cada servidor cada vez que un cliente llama
- 25%-30% del tiempo del servidor dedicado a las búsquedas en lugar de dedicar esos recursos a los servicios de correo

Impacto de la arquitectura en un caso real (cont)

Rackspace

Rackspace v2



Crearon una aplicación Web que acceden los CSR.

Centralizaron los log files. Utilizaron scp para hacer la transferencia de cada servidor de correo al servidor de logs

Los archivos se cargan cada 10 minutos a la BD de MySQL

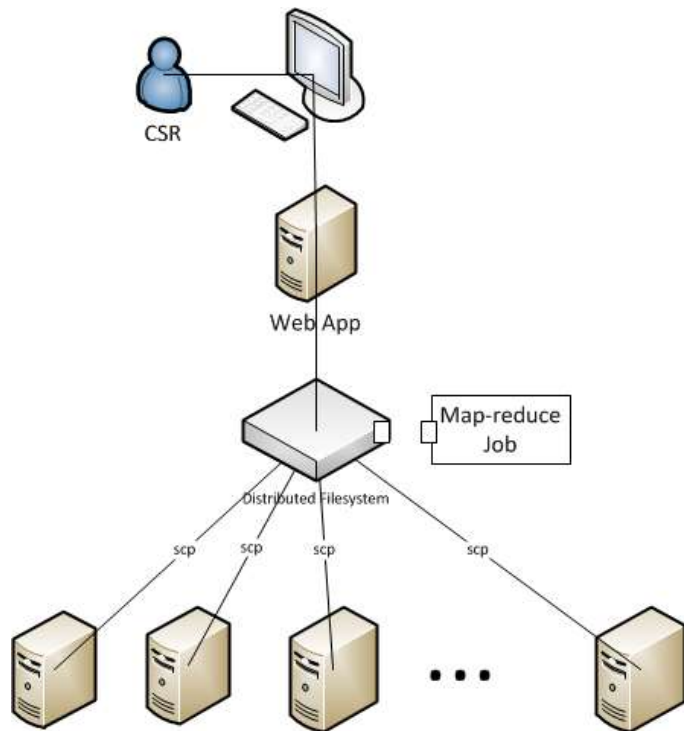
Impacto:

- Esfuerzo significativo para cargar estos archivos al servidor
- Estrategia de Scale-up provocaba que las cargas se fueran haciendo más y más lentas y el rendimiento de los indexes decrecimiento conforme las tablas crecían

Impacto de la arquitectura en un caso real (cont)

Rackspace

Rackspace v3



Map-Reduce Cluster

Hadoop es un file system distribuido que está diseñado específicamente para procesamiento a gran escala de datos distribuidos.

Commodity boxes

Map-Reduce ejecutado cada 20 minutos

Impacto:

- Estrategia de Scale-out
- Mantienen las ventajas de indexamiento de la version anterior

Impacto de la arquitectura en un caso real (cont)


Rackspace

Versión/Atributo	Disponibilidad de datos	Escalabilidad	Acceso a la información
V1	Real-time	Impacto en el rendimiento de los servidores de correo	Expresiones regulares
V2	10 minutos	Incremento en tiempos de carga Scale-in	SQL queries
V3	10-20 minutos	Sin problemas aún	Map-Reduce Job

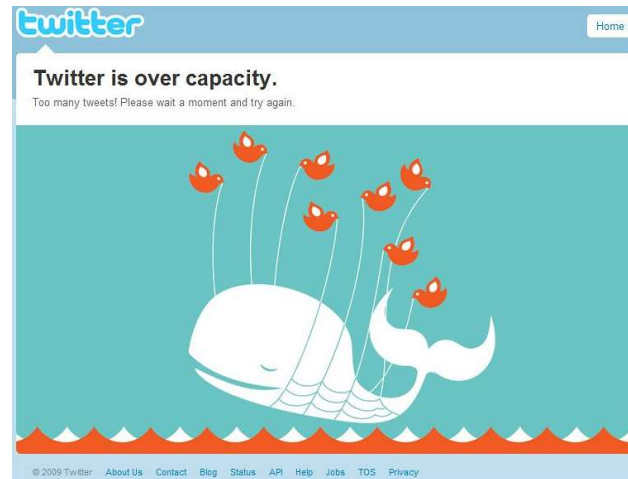
Impacto de la arquitectura en un caso real (cont)


Rackspace

Versión/Atributo	Disponibilidad de datos	Escalabilidad	Acceso a la información
V1	Real-time	Impacto en el rendimiento de los servidores de correo	Expresiones regulares
V2	10 minutos	Incremento en tiempos de carga Scale-in	SQL queries
V3	10-20 minutos	Sin problemas aún	Map-Reduce Job

Qué tienen en común estas 3 cosas?

27



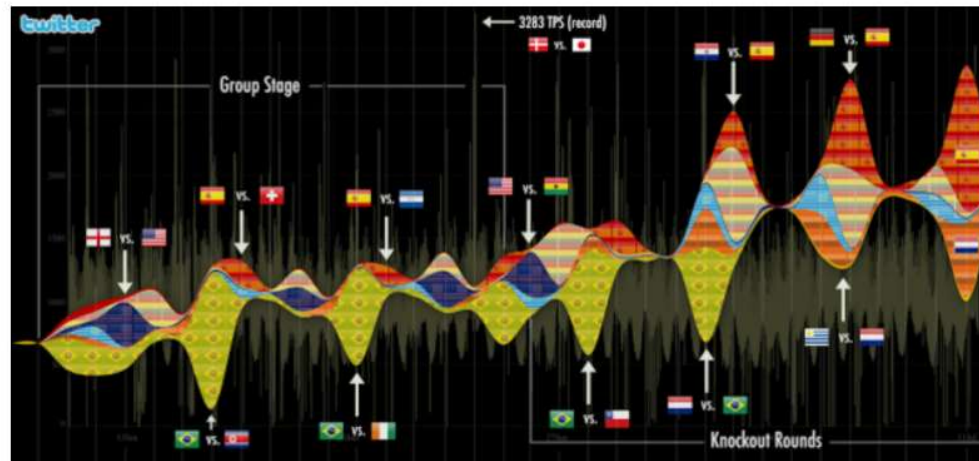
Mundial 2010

28

The 2010 World Cup: a Global Conversation

By @mgrooves

Friday, 16 July 2010



2010 World Cup: a Twitter timeline

El estado inicial



- Twitter tenía una de las implementaciones más grandes de Ruby on Rails. La operación era mantenida por cerca de 200 ingenieros que permitieron un crecimiento explosivo de la aplicación.
- La aplicación estaba organizada de manera monolítica en donde todas las actividades: manejo de BD, memcache, rendering del sitio, API's etc. Se encontraban en el mismo codebase.
- La estructura de la solución hacía sumamente difícil que los ingenieros tuvieran un conocimiento profundo de la aplicación y presentaba serios retos para organizar el trabajo y paralelizar actividades en los equipos de ingenieros.
- La organización había alcanzado un límite en el rendimiento de los sistemas de almacenamiento (MySQL). El sistema estaba con problemas en la ingesta de nuevos tweets a la velocidad en que estaban siendo creados.
- Para mantener la operación, la organización continuaba agregando hardware en lugar de pensar en soluciones ingenieriles.
- Finalmente la organización empezó a realizar cambios en el código que sacrificaban la flexibilidad y facilidad de lectura del código para tener beneficios en rendimiento y eficiencia.

El estado ideal



- Mejorar la media de latencia que experimentaban los usuarios en Twitter.
- Reducir el número de máquinas requeridas para ejecutar la operación en 10x.
- Aislar los fallos a lo largo de la infraestructura para prevenir largos tiempos de no disponibilidad.
- Límites más limpios agrupando la lógica relacionada en un solo lugar. Tratar de aplicar las mejores prácticas de modularidad y encapsulamiento.
- Hacer release de nuevas funcionalidades más rápido.

La ruta del cambio



Análisis del front-end usando métricas de CPU, RAM y Red

Ruby VM vs JVM

- Rails servers manejaban 200-300 request/sec/host
- Usando experiencias pasadas con Scala en JVM como base realizan un cambio.
- La nueva versión permite 10-20k Request/sec/host

Modelo de programación

- Ruby manejaba la concurrencia a nivel de proceso, cada solicitud era encolada y manejada por un proceso
- El JVM tenía operaciones primitivas que soportaban crear aplicaciones realmente concurrentes
- Se descompusieron los components en servicios
- Librerías comunes para el manejo de pools de conexiones, estrategias de failover, balanceo de carga

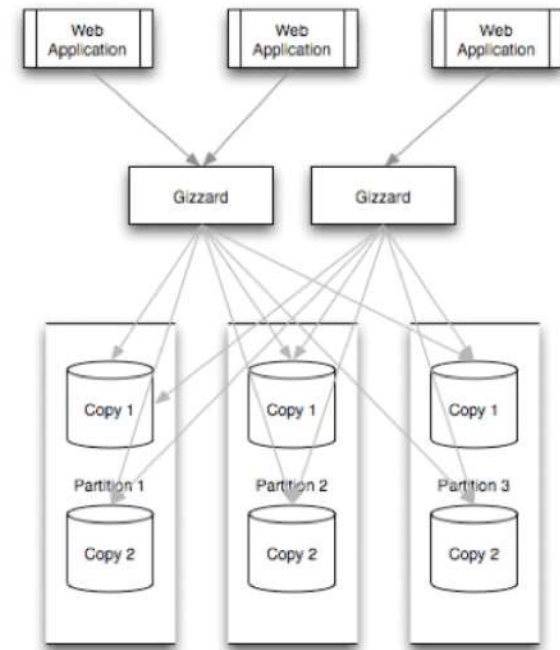
La ruta del cambio

Almacenamiento:

A pesar de que la aplicación monolítica fue partida en servicios seguía existiendo un gran cuello de botella en la BD

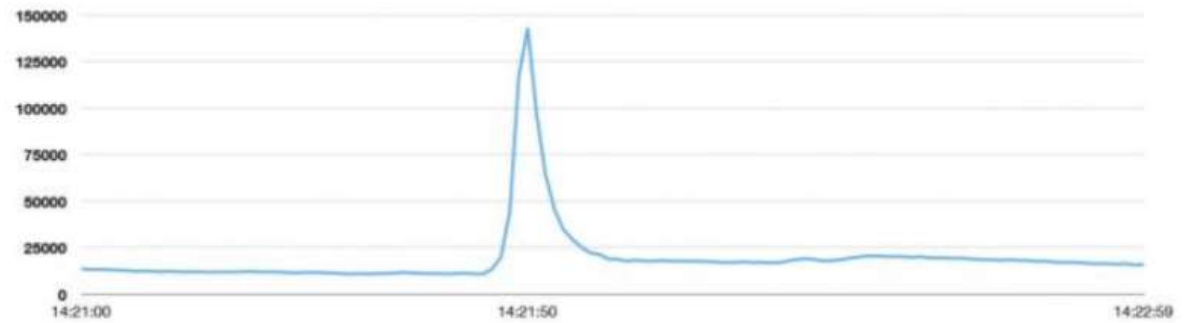
El esquema tradicional creaba una fila en la BD por cada tweet y esos se almacenaban en orden en la BD

Se cambia a Gizzard que toma cada tweer le aplica un hash y luego escoge la BD apropiada



El final de nuestra historia

33



“ New Tweets per second (TPS) record: 143,199 TPS. Typical day: more than 500 million Tweets sent; average 5,700 TPS. ”

Arquitectura de software y atributos de calidad del software

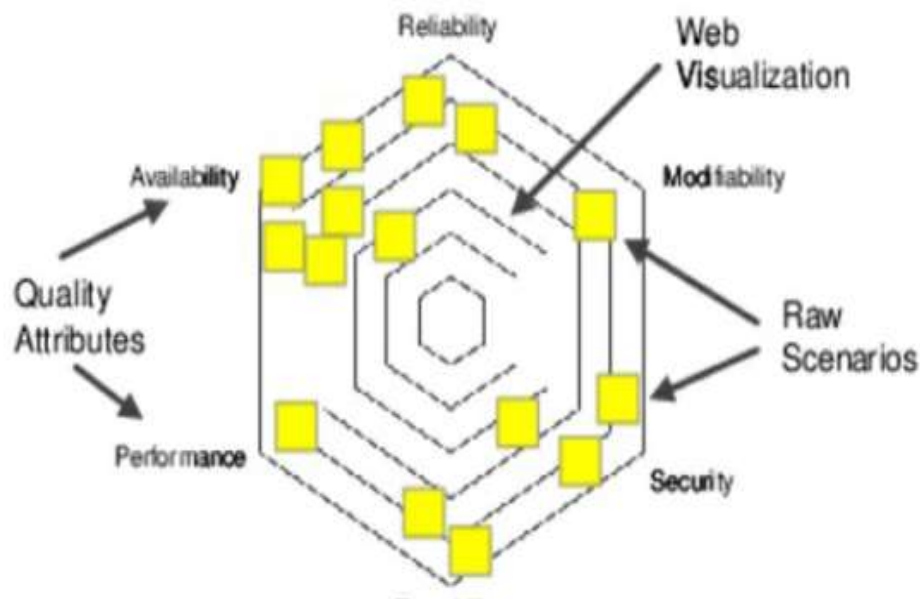


Herramientas en el proceso de creación de la arquitectura

35

Telaraña de atributos de calidad

Radar Chart



- Lluvia de ideas realizada con los stakeholders para capturar, categorizar, refinar y priorizar las preocupaciones principales
- Captura todas las preocupaciones de los stakeholders
- Cada preocupación se escribe en una nota y se coloca en el gráfico de radar
- Permite que los stakeholders se enfoquen en los atributos de calidad y no en las funcionalidades
- Permite visualizar como cada sistema es diferente en sus características
- Permite priorizar los atributos de calidad
- Los stakeholders realizan evaluaciones utilizando una escala de 1 a N y presentamos los resultados en un dashboard visual con sliders

Herramientas en el proceso de creación de la arquitectura

36

Ejemplos de la telaraña de atributos de calidad

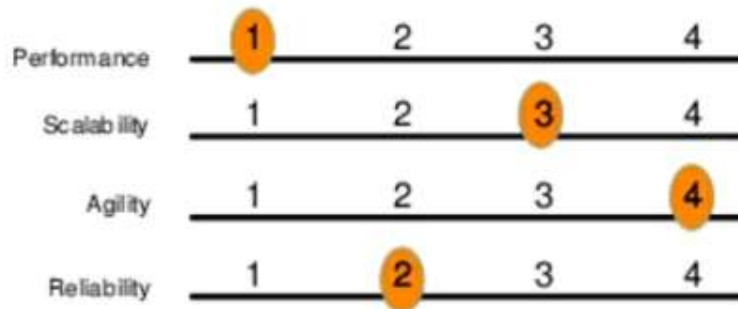


Herramientas en el proceso de creación de la arquitectura

37

Trade-off sliders

Evaluación de atributos

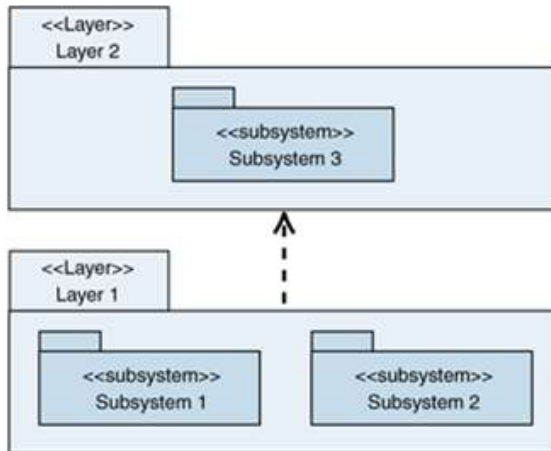


- Permite múltiples comparaciones entre los atributos de calidad del sistema
- Los stakeholders realizan evaluaciones utilizando una escala de 1 a N y presentamos los resultados en un dashboard visual con sliders

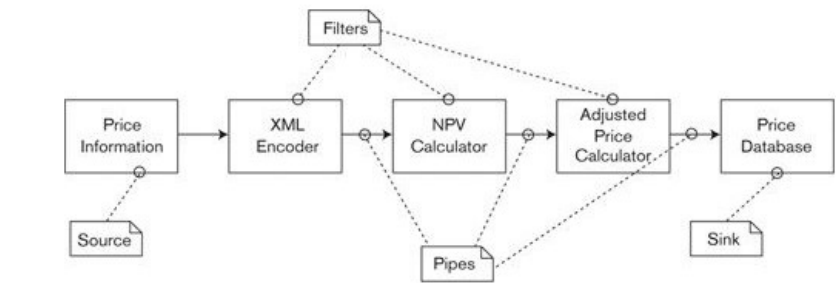
La arquitectura correcta

Diferentes estilos

Arquitectura en capas



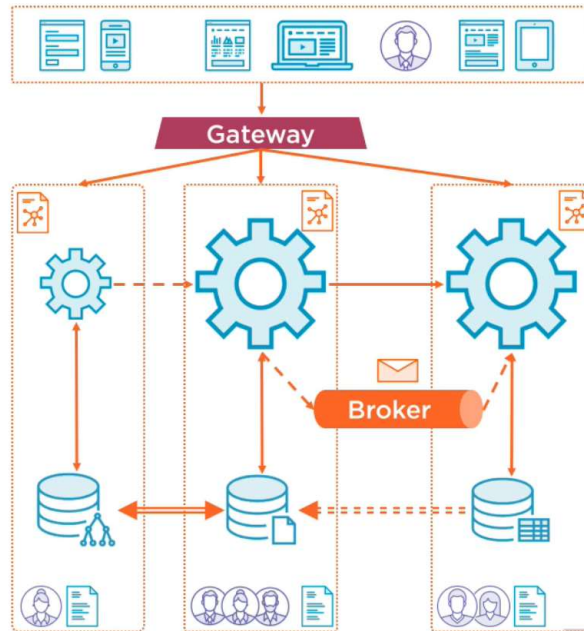
Arquitectura Pipes and Filters



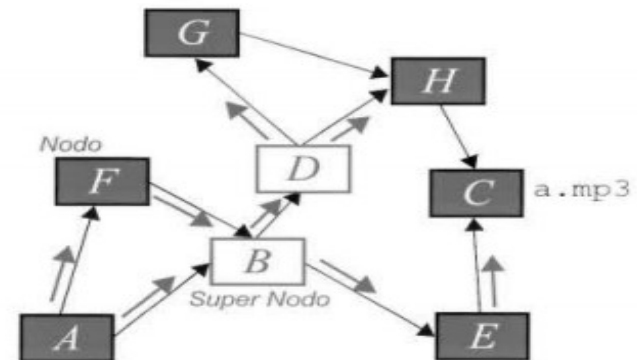
La arquitectura correcta

Diferentes estilos

Arquitectura en Microservicios



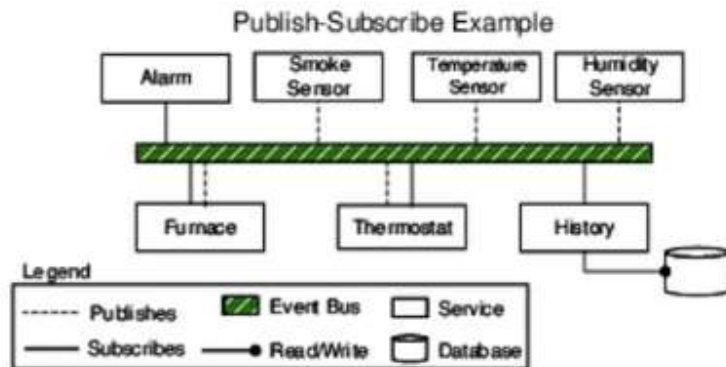
Arquitectura Peer-to-Peer



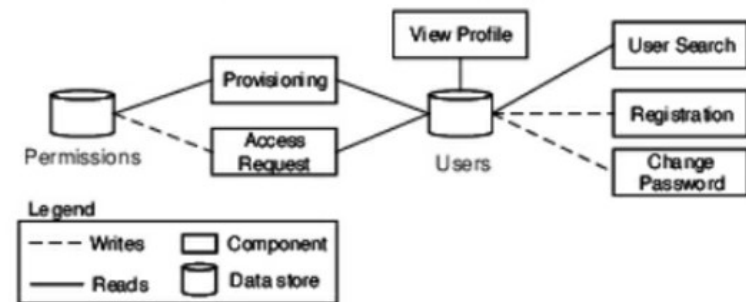
La arquitectura correcta

Diferentes estilos

Arquitectura en Publish-Subscribe

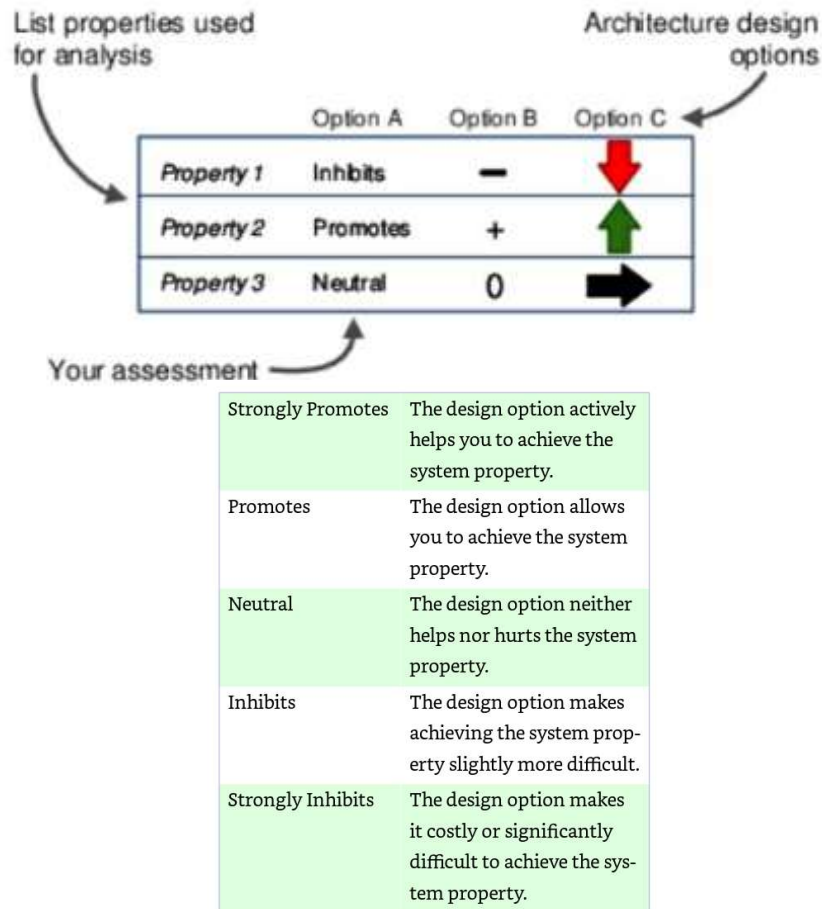


Arquitectura Centralizada de Datos



Herramientas en el proceso de creación de la arquitectura

Matriz de decisión



	3-Tier	Publish - Subscribe	Service Oriented
Availability (Database unavailable)	+	○	+
Availability (Uptime requirements)	○	○	○
Performance (5-second response time)	○	—	+
Security	○	—	○
Scalability (5% annual growth)	○	○	+
Maintainability (Team knowledge)	+	—	○
Buildability (Implementation risks)	++	—	--

Legend

Strongly Promotes	++	Strongly Inhibits	--	Neutral	○
Promotes	+	Inhibits	—		

Preguntas?



daniel.madriz.h@gmail.com



8324-1684



www.tec.ac.cr