

TEÓRICA

ESTILOS DE ARQUITECTURA

* Flujo de Datos (Pipes and filters - Batch BCs)

- Pipes and filters
- Uniform pipes and filters
- Estándarizado el formato de I/O

* Replicación { - RR (Replicación Replicado) - se replica la BD

- cache (se almacena el result. de algún request de usuario).

limitaciones físicas y lógicas

difícil depuración (LRU)

los que los le usuarios (menor tiempo de procesamiento)

* Jerarquías

Arquitectura

PRÁCTICA

"RUN-TIME ABSTRACTION"

FASE DE OP

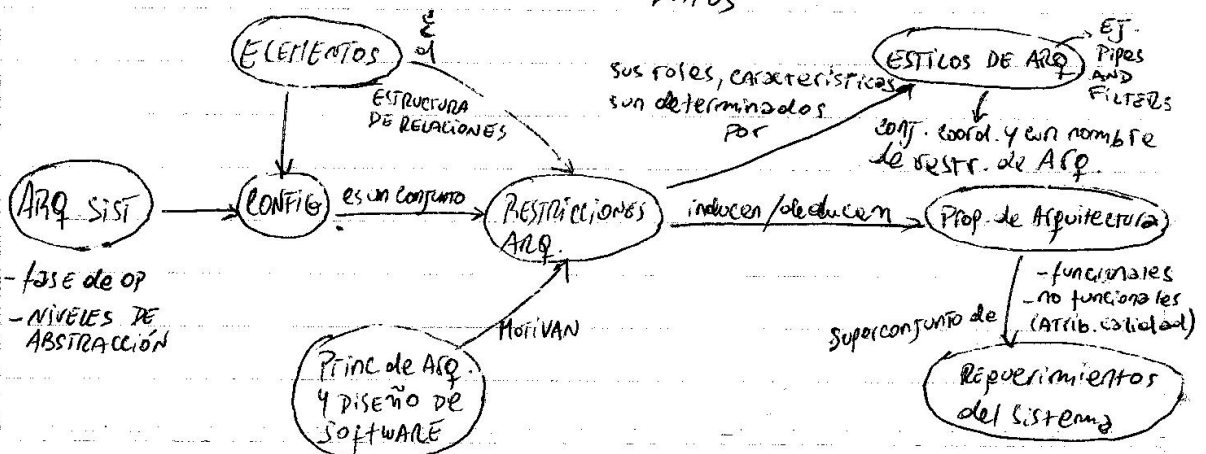
NIVELES DE ABSTRACCIÓN

ELEMENTOS ARQUITECTURALES

COMPONENTES

CONECTORES

DATOS



DISEÑO DE SOFT.

Arq. ~> Forma en que se comunica HW y SF ~> COMPONENTES
DB
COMPONENTES
INTERFAZ

TEÓRICA

PRÁCTICA

Attr. de Michael

- Performance \rightarrow Throughput \rightarrow transacciones que se pueden procesar x seg

\hookrightarrow Network

\hookrightarrow User perceived

\hookrightarrow Response time (completion time)

\hookrightarrow Responsiveness (cant de tiempo que le toma al S/S aceptar el req. y empezar a trabajar en el)

\rightarrow Latency $\begin{matrix} \text{(FLOWER)} \\ \text{Time to execute a request (includes network)} \\ \text{(FILLING)} \\ \text{Time entre estado inicial y primer indicación de respuesta} \end{matrix}$

- Scalability - Un S/S es capaz de aceptar un aumento y mejorar la performance
e.g. agregar otro server (horizontal)

Vertical

Horizontal

\hookrightarrow Dato un comp. \rightarrow Agregar más componentes del mismo tipo.
hacerlo más pesado.

- Simplicity (SRP)

- Maintainability: facilidad con la cual se puede hacer un cambio en el S/S

\hookrightarrow Evolvability: hab. un comp. sin impactar negativamente en otros.

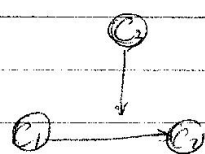
\hookrightarrow Extensibility: habilidad de la ap. de agregar nuevas func.

\hookrightarrow Customizability: hab. de poder especificar siempre y cuando ~~el~~ un elemento por el que puedes pensar a como un S/S. No hab. fixed. por el elemento.

↳ Configurability: hab. de hacer modificaciones ~~en~~ Post-Deploy.
/ esos comp. son capaces de usar nuevos servicios.

↳ Reusability: hab. de poder usar sus elementos en otras
aplic. o en modificación.

↳ Visibility: hab. que tiene un comp. de la exp. para
mejorar entre otros 2 comp.
HTTP > Visibilidad que TCP



Exponer la Interfaz de un componente
nos lleva a una mayor visibilidad.

Web service

↳ SOAP



getArdoz()
 calculate()
 XML

PRÁCTICA

↳ Portability: hab. de poder ejecutarse en dif. ambientes.

↳ Reliability (confiabilidad): ~~hab.~~ ~~proble~~ es
susceptible a fallar cuando se producen fallas parciales
de los componentes.

- SPOF (single point of failure) (EVITAR)

- Redundancia

- Monitoring

- Recoverable Failures (que fallas del sist. pueden ser
recuperables)

PRÁCTICA

Attr. de calidad:

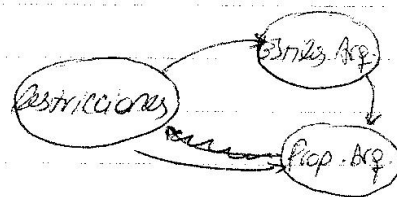
- Usability ~ UX: Buena experiencia del usuario en prod.
Interfaces - interacción con el usuario.
- Testability: Medida de cuán fácil es crear criterios de test de una función y sus componentes y cuán fácil es crear esos test.
- Security: Modo del sist. de prevenir acciones maliciosas o acc. indebidas fuera del fin para el cual fue diseñado / incluye pap. relativos de info o datos.
- Interoperability: Cómo interactúa con el resto de los sistemas.

Categorías de estilos de Arq.:

Data Flow

Los Pipes and Filters

- + extensibilidad
- + performance
- + Testability
- + simplicity
- + scalability
- + config.
- + visib.
- + exp.



↳ Uniform Pipes and Filters (todas las interfaces son iguales)

- + visib.
- Simp.
- Performance
- + Config.

TEÓRICA

Correct. del App.

PERFORMANCE (Rendimiento)

ESCALABILIDAD (Con cambios menores responde a cambios de Volumen)

SIMPLICIDAD

MODIFICABILIDAD (Cuanto es fácilmente modificable)

EVOLUCIONABILIDAD (Comp. tienen que hacer apps distintas)

EXTENSIBILIDAD (Más funciones)

PERSONALIZACIÓN (Personalizar app. para que responda a necesidades de cada persona)

CONFIDABILIDAD (Post-deploy - Es confiable?)

VISIBILIDAD (Tráfico entre procesos se puede controlar)

PORTABILIDAD (Se puede pasar a otra plataforma - el concepto)

CONFIDABILIDAD (Si falla alguno de los componentes, no falla todo el sist.)

No se

~~JOHN~~

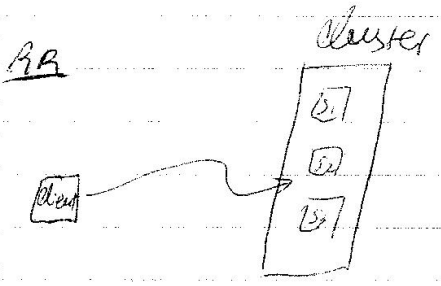
PRACTICA

Architecture Styles

Replication

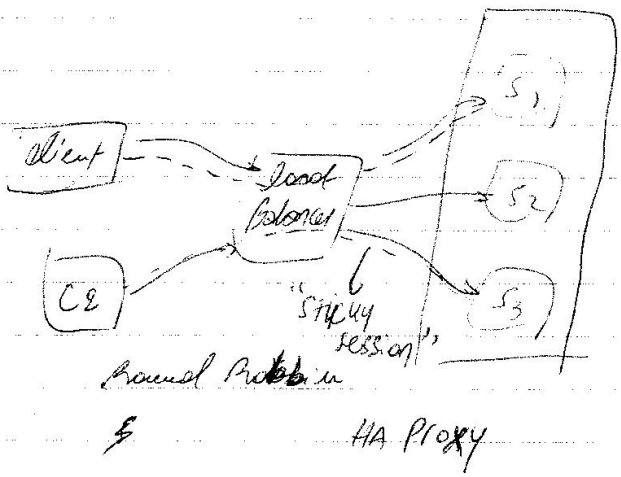
↳ RR = Replicated Repository

↳ \$ = cache



Serv. descentralizados → Client Cree fue el Serv. esta Centralizado.

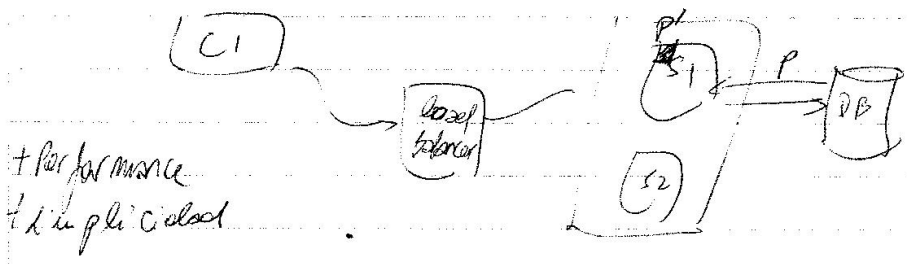
- + Confiability = Reliability
- + Performance
- + scalability



Round Robin

HA Proxy

\$



- + Performance
- + Replicated

Hierarchical (CS) Client Server

PRACTICA

+ Evolucionabilidad

+ Escalabilidad

+ Simplicidad

(CSS) → Client Stateless servers (sin estado de sesión del cliente)

- Performance

+ Simplicidad

+ EVO

+ Escalability

+ Versatility

TEORIA

Modelo 4+1 (Forma de documentar hoy en día una org.) (Philippe Krutchen)

Visión lógica {
 D. de cosas
 D. de comunicaciones
 D. de secuencias

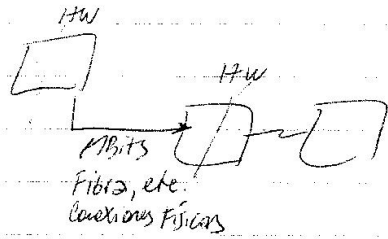
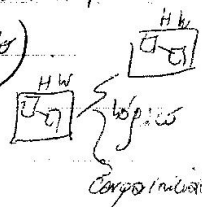
Visión desarrollo {
 D. de comp (Artefactos)
 D. de paquetes (Artefact dentro de un nodo)

Visión PROCESO
 (cómo func. la org.
 una vez que está terminada)
 {
 D. de actividad

Visión Física {
 D. de despliegue

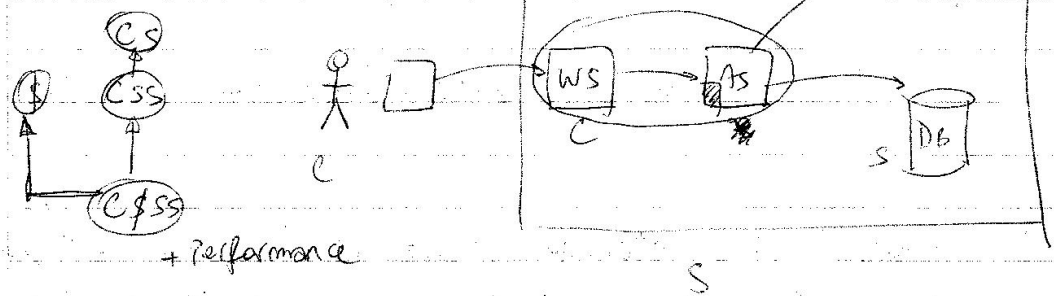
C.V. seleccionados (en principio)

Componentes
 Artefact: Soft
 Node: Hard



PRACTICA

Hierarchical

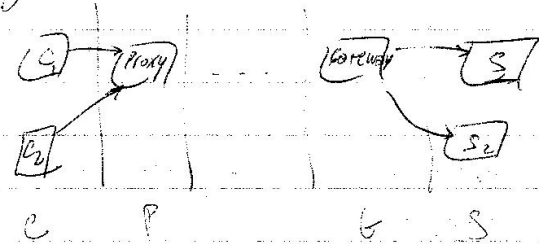


+ performance

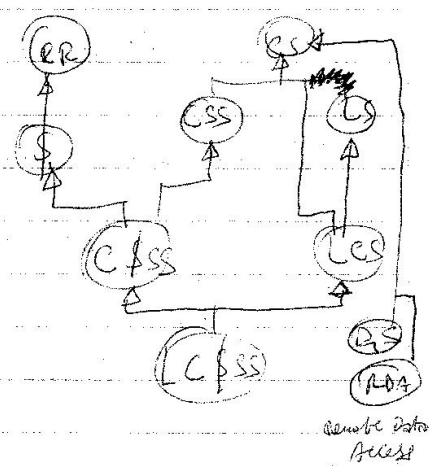
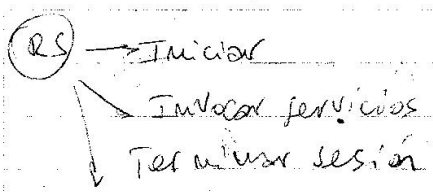
Layered system (LS)

- + Evolu
- + scalability
- + extensibility
- + portability
- performance

(LCS) Layered Client Server



LC\$SS

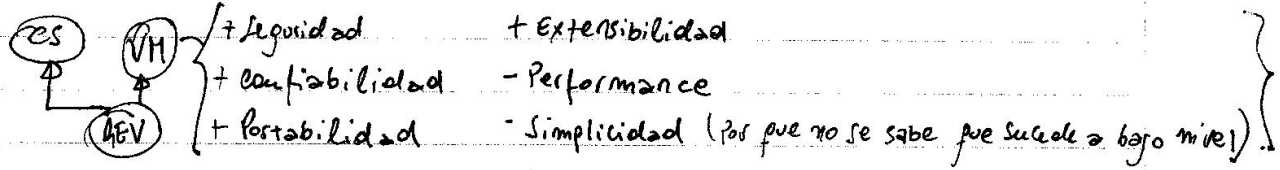


- + simplicity
- + dependability
- visibility
- extensibility
- confidentiality

RS = Remote Session
RPA: Remote Data Access

Mobile Code

PRÁCTICA



VM: Ejecutar código en un amb. controlado. Seguridad y Confiabilidad.

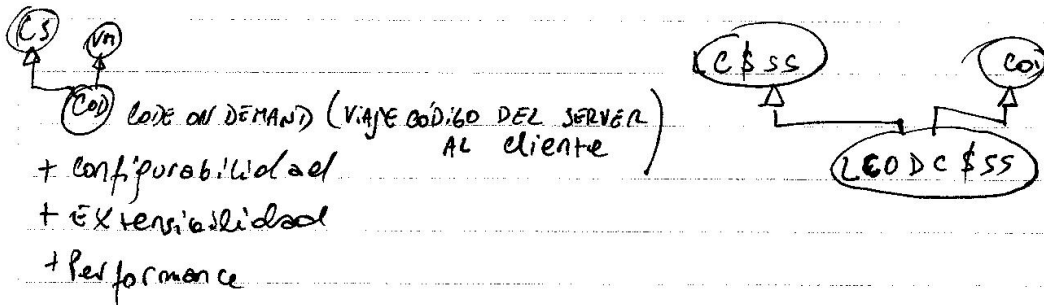
REV (REMOTE EVALUATION): El cliente tiene el conocimiento para ejecutar un servicio, pero no tiene los recursos necesarios para hacerlos. (Algoritmo)

+ Extensibilidad.

+ Personalización.

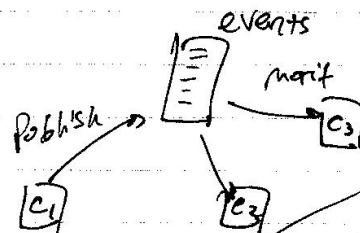
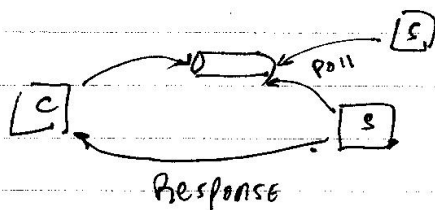
- Simplicidad

- Visibilidad (El monitor tiene que saber del código).



Peer to Peer (No hay un rol determinado de quien es el cliente y quien es el server)

EBI EVENT-BASED INTEGRATION



Funcionalidades ante un evento.

+ Extensibilidad

+ Evolucionabilidad