

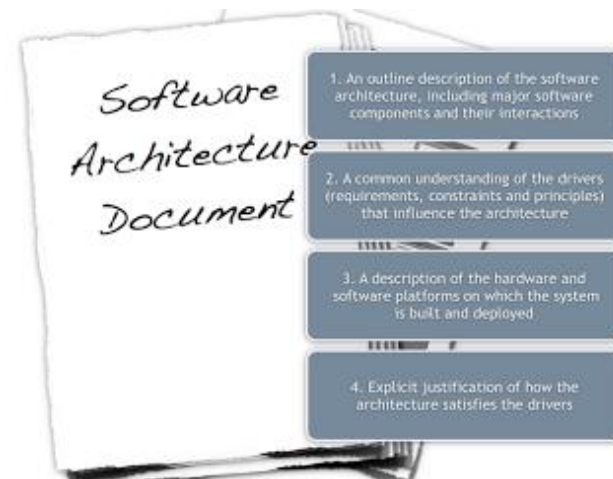


Arquitectura de Software

- ✓ Documento de Arquitectura de Software

Documento de Arquitectura de Software

- Es el documento que abarca a todo el sistema.
- No existe un estándar definido.
- Habitualmente se organiza por vistas.
- Especifica aspectos técnicos y funcionales.
- Se considera un documento 'vivo' que sufrirá modificaciones a lo largo del proyecto.

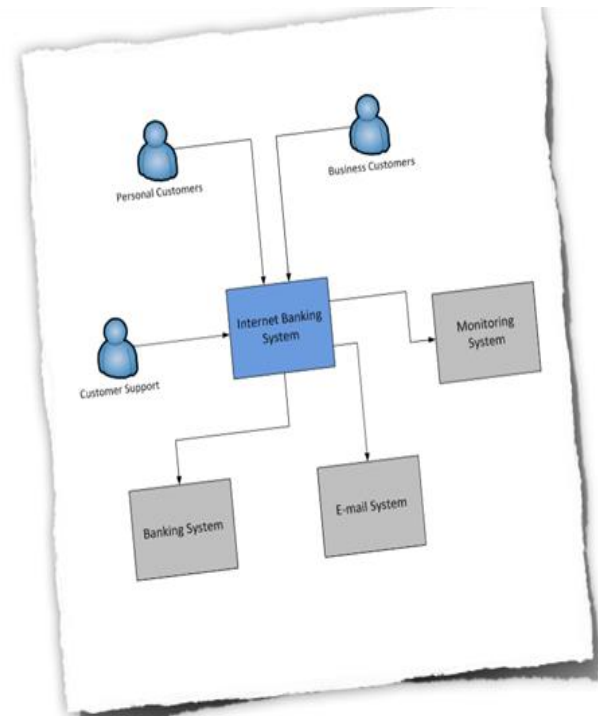
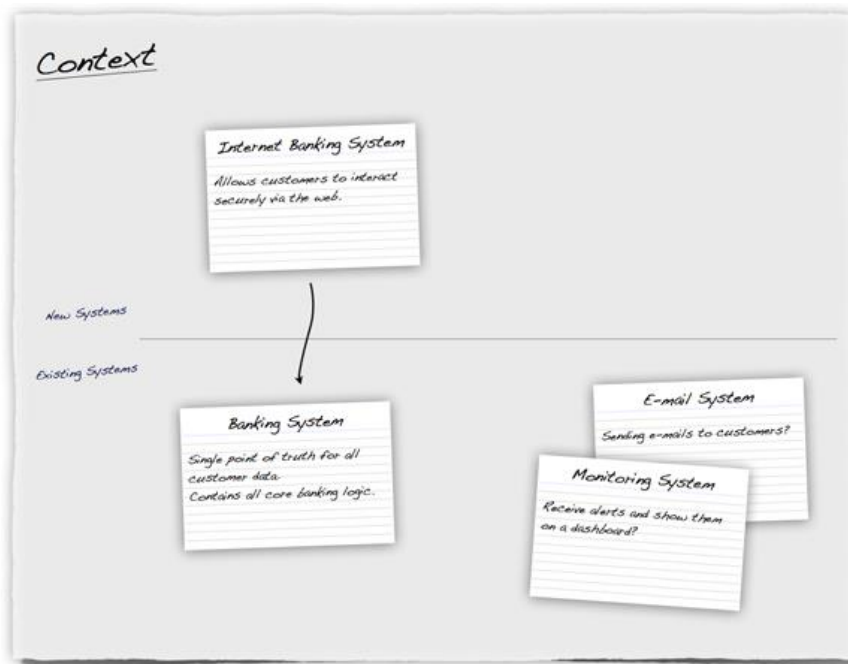


- Este documento provee información complementaria al código fuente y podría incluir:

1. Una descripción de la arquitectura de software, incluyendo los principales componentes y sus interacciones.
2. Un entendimiento común de los aspectos que influyen la arquitectura.
3. Una descripción de las plataformas del hardware y software en el cual el sistema estará construido y desplegado.
4. Justificación de cómo la arquitectura satisface los aspectos que influyen la arquitectura.

Contexto

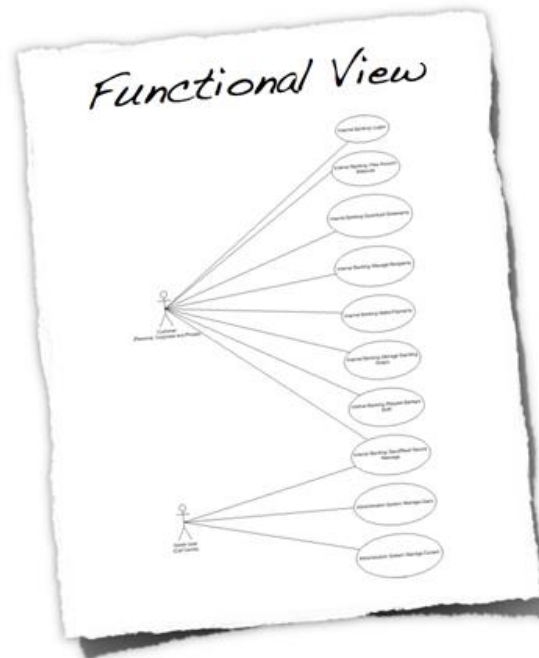
- Entender el 'paisaje' del sistema.
- Que es lo que estamos construyendo?
- Cómo calza en el ambiente existente?



Vista Funcional

- Permite sumarizar las principales funciones del sistema.
- La vista funcional debe responder la pregunta:

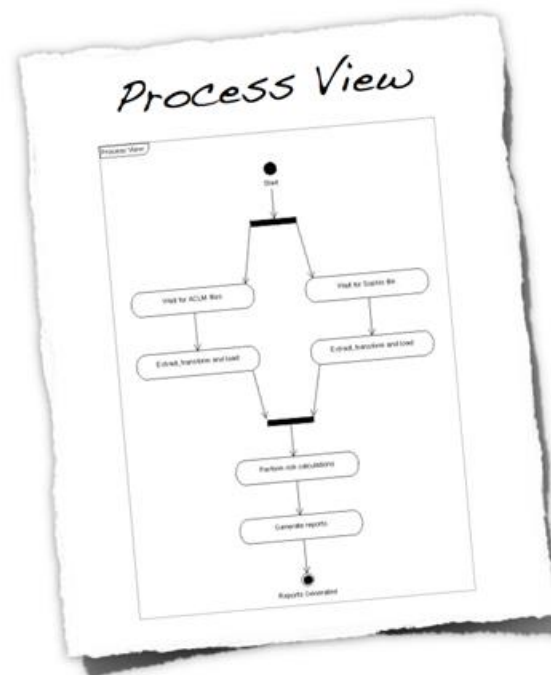
Esta claro lo que el sistema hace?



Vista del Proceso

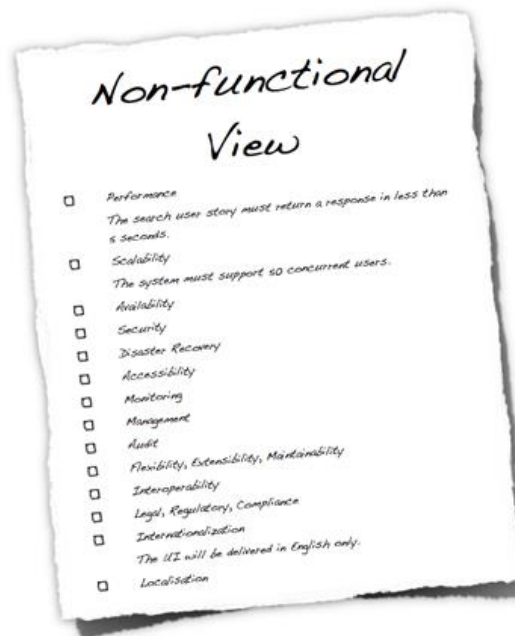
- Muestra lo que hace el sistema a un alto nivel.
- Muy útil cuando se está automatizando un proceso de negocios.
- La vista del proceso debe responder la pregunta:

Esta claro lo que el sistema hace en una perspectiva de proceso?



Vista No Funcional

- Permite sumarizar los principales requerimientos no funcionales.
- Resalta a aquellos que son significativos para la arquitectura.
- Los más comunes requerimientos no funcionales son:
Performance, escalabilidad, disponibilidad, seguridad, etc.



Restricciones

- El software vive en el mundo real y este tiene restricciones.
- En esta sección definir las restricciones y que queden claras.
- Los más comunes restricciones son:
Tiempo, presupuesto, recursos, estándares, protocolos, etc.

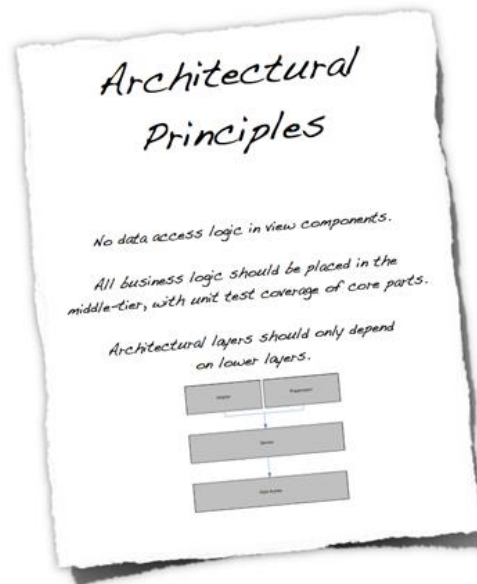


Principios

- Permite especificar aquellos principios que debemos seguir para definir la arquitectura.

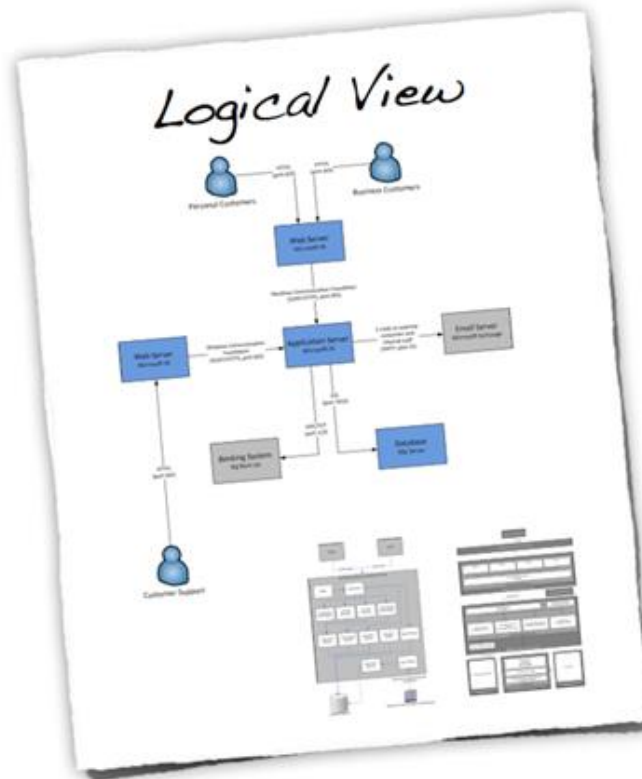
- Algunos ejemplos de principios son:

Arquitectura por capas, uso de librerías y frameworks, patrones de diseño, etc.



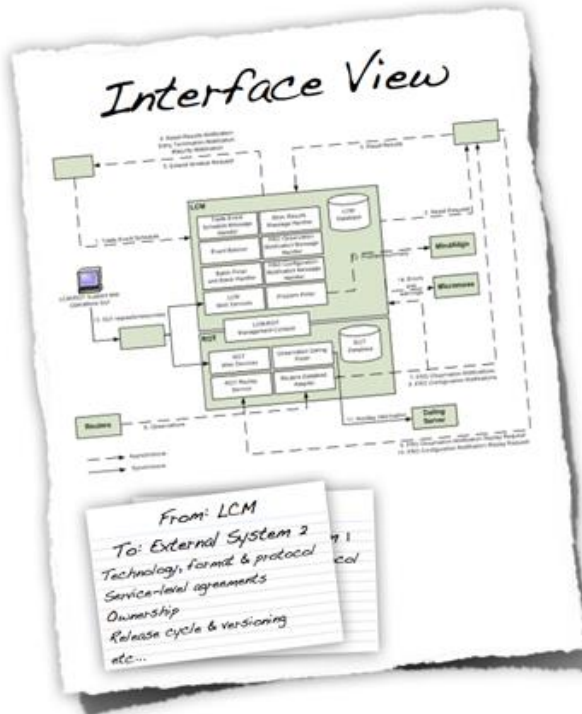
Vista Lógica

- Es la gran figura que permite presentar la estructura del sistema a través de sus componentes e interacciones.
- Incluye las elecciones de tecnología a un alto nivel:



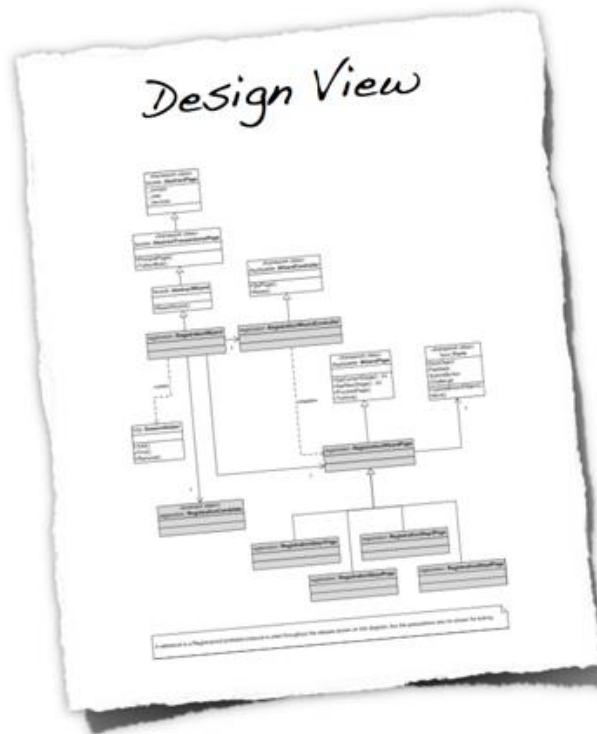
Vista de las interfaces

- Relacionada con la vista lógica.
- Las interfaces son una de las partes mas riesgosas de un sistema, así que es muy importante mostrar las interfaces internas y externas y cómo se trabajan con ellas.



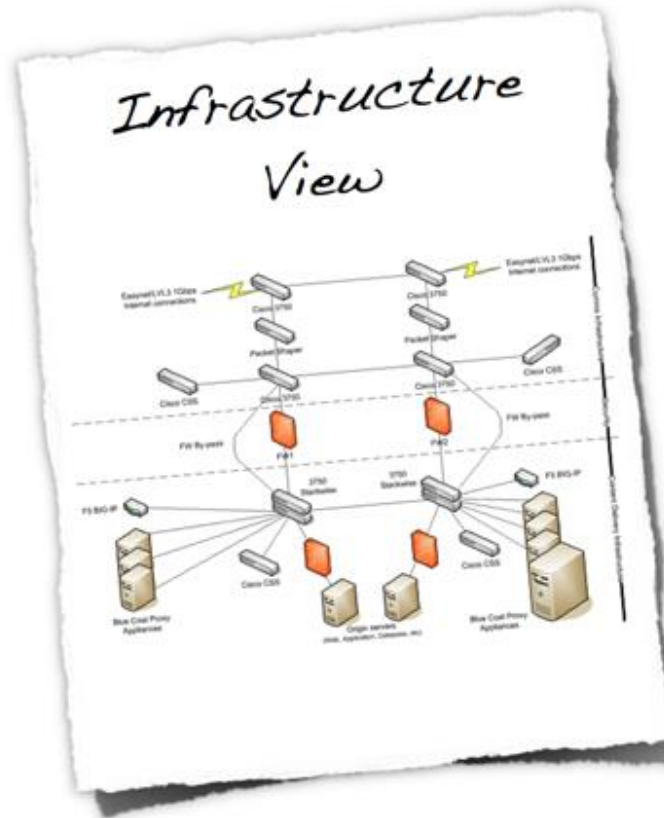
Vista de Diseño

- La vista de diseño es donde se especifica a un nivel mas bajo los detalles de la implementación del sistema.
- Por ejemplo, podría incluir la documentación de los patrones, de tecnologías, frameworks, etc.



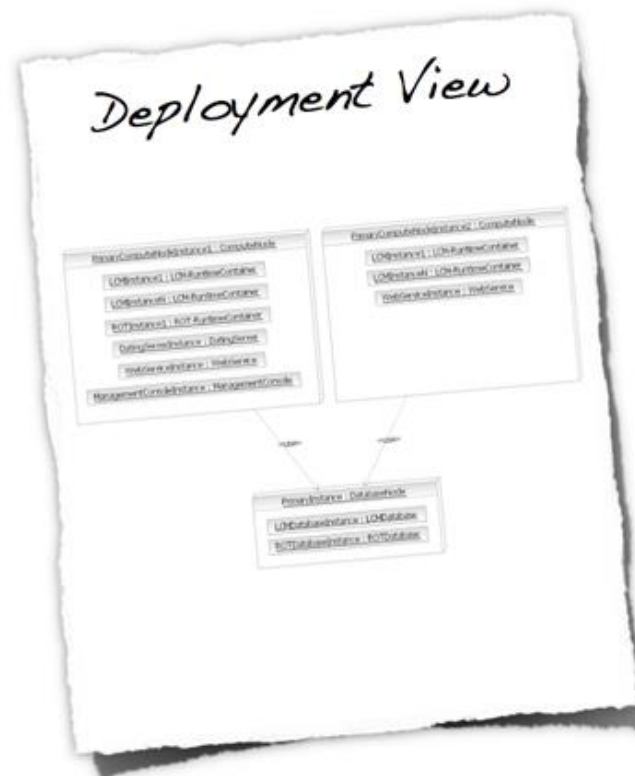
Vista de Infraestructura

- La vista de Infraestructura es usada para describir el hardware físico y las redes en la cual el sistema será desplegado.



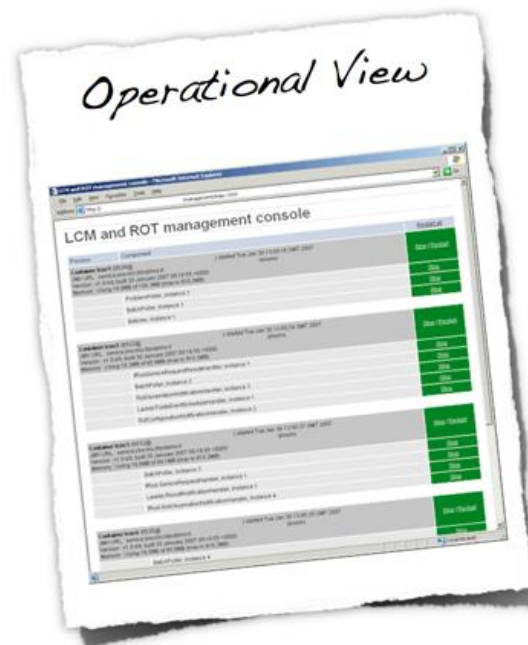
Vista de Despliegue

- La vista de Despliegue detalla cómo el software será desplegado en la infraestructura física.



Vista Operacional

- La vista Operacional se especifica cómo el sistema se ejecuta, monitorea y administra.
- Se debe especificar cómo el sistema se realizará el monitoreo en todas las capas, ejm. Desde el cliente hasta la base de datos. Cómo se diagnosticará los problemas o dónde se encuentran los errores o logs del sistema?



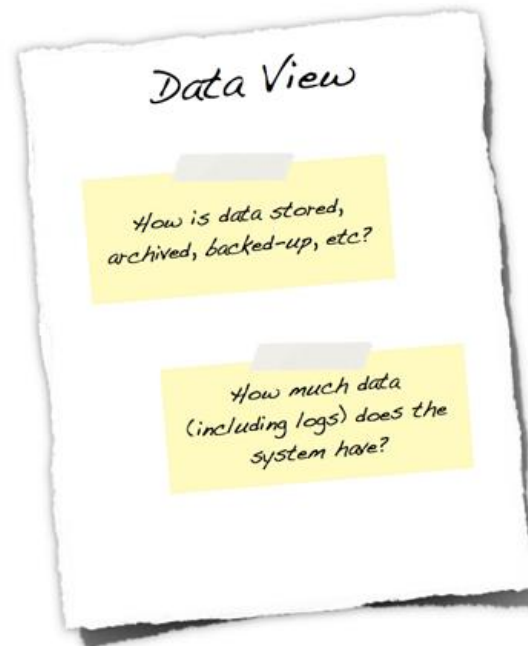
Vista de Seguridad

- La vista de Seguridad especifica cómo el sistema implementa los requerimientos de seguridad como autenticación, autorización, confidencialidad de los datos, etc.



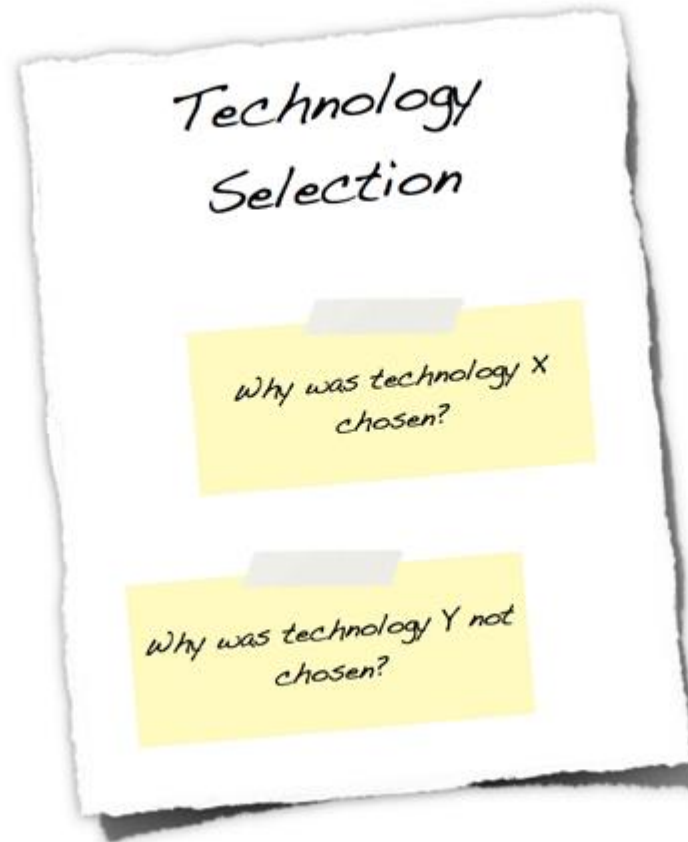
Vista de los Datos

- La vista de Datos es importante si el sistema esta centrado en los datos, manejan do una gran cantidad de datos o trabajando con complejos flujos de datos.
- Incluir información del tamaño, capacidad de planeamiento usando archivos o backup.



Selección de la Tecnología

- En esta sección del documento se especifica las decisiones de por qué se escogió o no una tecnología.



Justificación de la Arquitectura

- En esta sección se especifica cómo los requerimientos no funcionales son cumplidas por la arquitectura.





Agenda

✓ Técnica de modelamiento

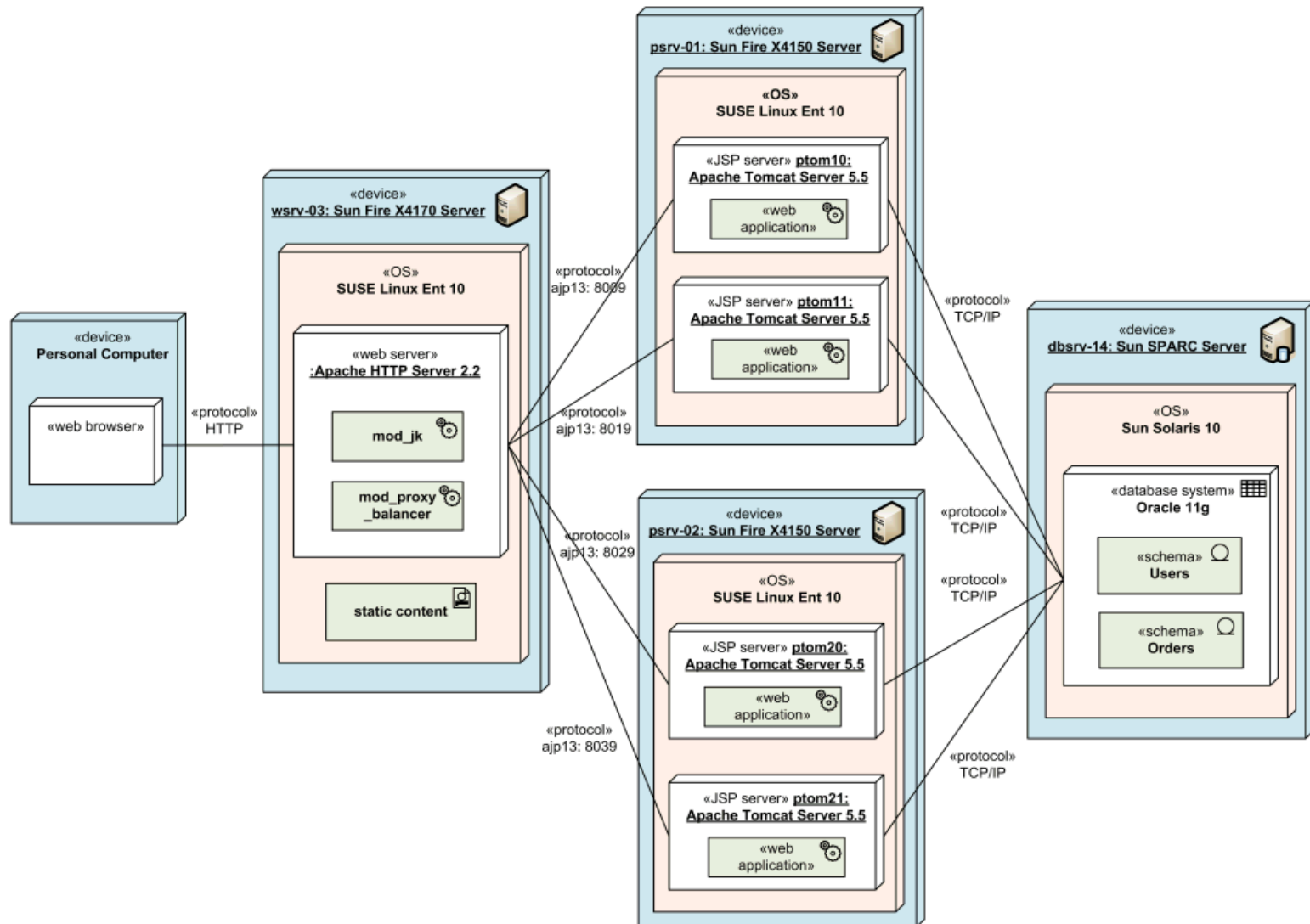
Pasos para modelar

- ❑ Leer la descripción de la arquitectura, entender los requerimientos.
- ❑ Elegir un estilo de arquitectura.
- ❑ Identificar los componentes primarios o servidores. Ejm. Servidor Web, servidor Legacy
- ❑ Identificar cómo se relacionan los componentes primarios Ejm. protocolos.
- ❑ Identificar si existen requerimientos no funcionales Ejm. Seguridad, alta disponibilidad, etc.
- ❑ Completar la arquitectura con información adicional que se haya encontrado. Ejm. Sistemas operativos, tecnologías, etc.
- ❑ Analizar o evaluar la arquitectura.

Caso Practico

- ❑ El banco RDMR está expandiendo sus servicios a sus clientes, para lo cual está implementando una arquitectura que soporte las operaciones en línea usando smartphones de plataformas distintas (iphone y android),
- ❑ Estas aplicaciones cliente reutilizan los mismos servicios que la aplicación web de home banking.
- ❑ El core principal se encuentra en un servidor legacy AS400 donde se ejecutan las transacciones que son solicitadas desde la aplicación web de home banking, esta aplicación web además utiliza una base de datos DB2 para registrar las operaciones que envía al AS400 usando el protocolo JDBC.

Caso Practico



Descanso ...

