

INGENIERÍA DE SOFTWARE 2 – CLASE 8, PARTE 1: DISEÑO ARQUITECTÓNICO

- El diseño arquitectónico define la relación entre los elementos de estructura del sistema, para lograr los requisitos del mismo.
 - Es un proceso de identificación de los subsistemas dentro del sistema y establecer el marco de control y comunicación entre ellos.
 - Grandes sistemas son divididos en subsistemas que proporcionan algún conjunto de servicios relacionados.
 - Se desarrolla en base a componentes y los vínculos entre ellos.

Diseño arquitectónico

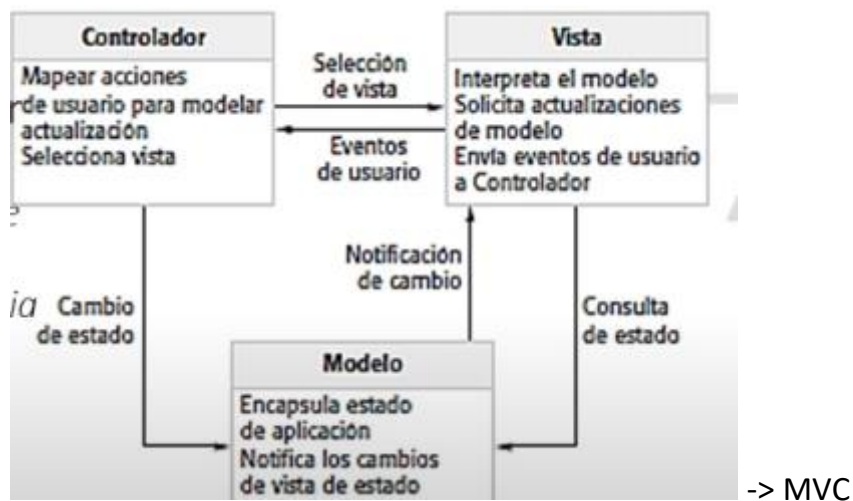
- La arquitectura afecta directamente a requerimientos no funcionales.
 - Los más críticos
 - Rendimiento, protección, seguridad, disponibilidad, mantenibilidad.
 - Rendimiento
 - Se agrupan las operaciones críticas en un grupo reducido de sub-sistemas (componentes de grano grueso, baja comunicación).
 - Seguridad
 - Se utiliza una arquitectura en capas, protegiendo los recursos más críticos en las capas más internas.
 - Protección
 - La arquitectura debe ser diseñada para que las operaciones relacionadas con la protección sean localizadas en un único sub-sistema (o grupo pequeño), de forma que se reducen los costos y problemas de validación de la protección.
 - Disponibilidad
 - La arquitectura se deberá diseñar con componentes redundantes para que sea posible el reemplazo sin detener el sistema, esto nos otorga una arquitectura muy tolerante a fallas.
 - Mantenibilidad
 - La arquitectura del sistema debe diseñarse con componentes autocontenidos de grano fino que puedan modificarse con facilidad.
 - Grano fino: una alta cantidad de componentes pequeños.
 - Componentes pequeños -> más fáciles de modificar y corregir errores, pero va en contra del rendimiento.

INGENIERÍA DE SOFTWARE 2 – CLASE 8, PARTE 1: DISEÑO ARQUITECTÓNICO

Aspectos del diseño arquitectónico

Organización del sistema

- Representa la estrategia básica usada para estructurar el sistema.
 - Los subsistemas de un sistema deben intercambiar información de manera efectiva.
 - Los datos compartidos se guardan en una base de datos central.
 - Cada subsistema mantiene su información y los intercambia entre subsistemas.
 - Estilos organizacionales (patrones de arquitectura):
 - Repositorio, cliente-servidor, capas o combinaciones entre ellos.



Patrón de Repositorio

- La mayoría de sistemas que usan grandes cantidades de información se organizan alrededor de una BD central y compartida (repositorio).
- Los datos son generados por un subsistema y utilizados por otros subsistemas.
- Ejemplos
 - Sistemas de gestión, Sistemas CAD, herramientas Case, etc.

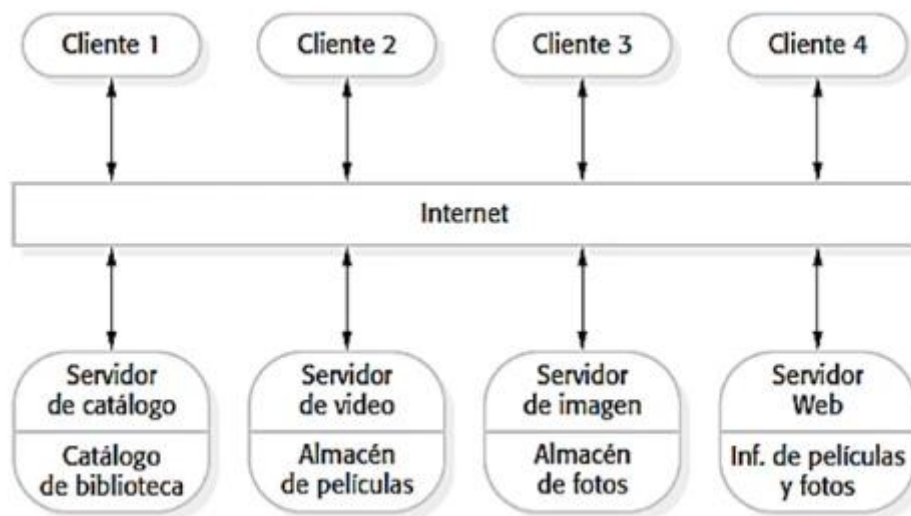
INGENIERÍA DE SOFTWARE 2 – CLASE 8, PARTE 1: DISEÑO ARQUITECTÓNICO



-
- Todos los que tocan al repositorio son subsistemas.
- Ventajas de este patrón
 - Forma eficiente de compartir grandes cantidades de datos, no hay comunicación entre subsistemas.
 - Todos acceden al repositorio compartido.
 - Los subsistemas que generan datos no saben como son usados esos datos.
 - Independizar los sistemas entre sí.
 - Las actividades de backup, protección y control de acceso están centralizadas en el repositorio.
 - El modelo compartido es visible a través del esquema del repositorio.
 - Las nuevas herramientas se integran de manera directa, ya que estas son compatibles con el modelo de datos.
- Desventajas de este patrón
 - Los subsistemas deben estar acordes a los modelos de datos del repositorio. Puede afectar el rendimiento.
 - La evolución puede ser difícil a medida que generamos un gran volumen de información de acuerdo con el modelo de datos establecido. La migración de estos modelos puede ser muy jodida, a veces imposible.
 - Diferentes subsistemas pueden tener distintos requerimientos de protección o políticas de seguridad y el modelo de repositorio impone las mismas para todos.
 - Es difícil distribuir el repositorio en varias máquinas, existen repositorios centralizados lógicamente pero pueden ocasionar problemas de redundancia e inconsistencias.

INGENIERÍA DE SOFTWARE 2 – CLASE 8, PARTE 1: DISEÑO ARQUITECTÓNICO

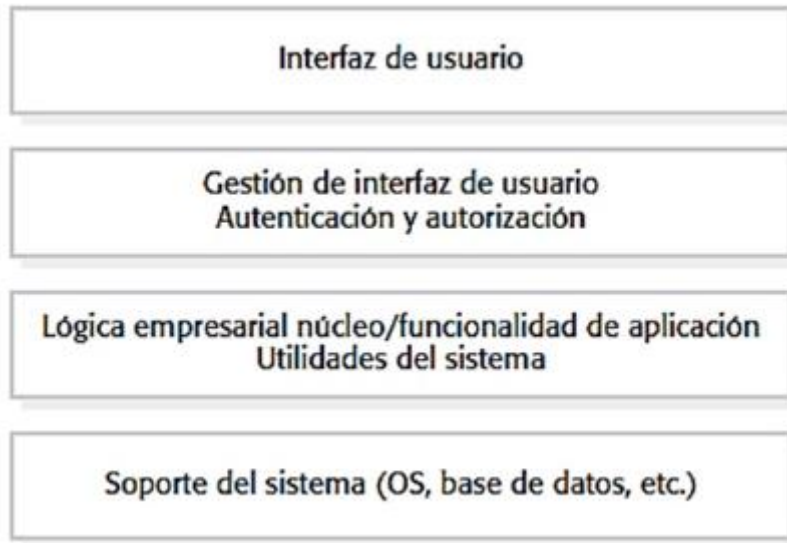
Patrón Cliente-Servidor



- Es un modelo donde el sistema se organiza como un conjunto de servicios y servidores asociados a esos servicios, más unos clientes que utilizan los servicios.
 - Componentes
 - Conjunto de servidores que ofrecen servicios, otros sistemas.
 - Conjunto de clientes que llaman a los servicios.
 - Una red que permite a los clientes acceder a los servicios.
 - El caso particular es cuando los servicios y el cliente corren en la misma máquinas. Para pruebas.
 - Los clientes conocen el nombre del servidor y el servicio que brinda, pero el servidor no necesita conocer al cliente.

Patrón de arquitectura en capas

INGENIERÍA DE SOFTWARE 2 – CLASE 8, PARTE 1: DISEÑO ARQUITECTÓNICO



- El sistema se organiza en capas, donde cada una de ellas presenta un conjunto de servicios a sus capas adyacentes.
 - Ventajas
 - Desarrollo incremental.
 - Portable y resistente a cambios.
 - Una capa puede ser reemplazada siempre que se mantenga la interfaz, y si varía la interfaz se genera una capa para adaptarlas.
 - Permite generar sistemas multiplataforma, ya que solamente las capas más internas son dependientes de la plataforma (una capa interna para cada plataforma)
 - Desventajas
 - Difícil de estructurar, dividir en capas.
 - Las capas internas proporcionan servicios que se requieren por todos los niveles.
 - Los servicios que requieren el usuario pueden estar brindados por las capas internas teniendo que atravesar varias capas adyacentes.
 - Si hay muchas capas, un servicio solicitado de la capa superior puede tener que ser interpretado varias veces en diferentes capas.
- Ejemplo del patrón

INGENIERÍA DE SOFTWARE 2 – CLASE 8, PARTE 1: DISEÑO ARQUITECTÓNICO

