



Programación II

Arquitecturas de software

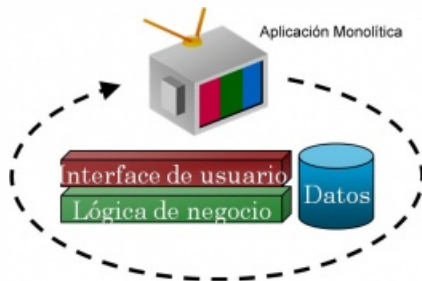
La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel.

Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

Kruchten, Philippe

Arquitectura Monolítica

- El software se estructura en unidades funcionales muy acopladas.
- No hay separación física ni lógica.
- El sistema final consiste de módulos compilados separadamente y luego enlazados.



Ventajas.

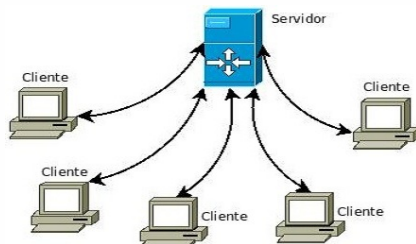
- Eficientes ya que se producen pocos cambios de contexto

Desventajas.

- Baja reutilización
- Poca escalabilidad
- Los cambios afectan a todo el sistema

Arquitectura Cliente-Servidor

- Modelo de aplicación distribuida donde se reparten las tareas entre los proveedores de servicios (servidores) y quienes los usan (clientes)
- La separación entre cliente y servidor puede ser solo a nivel lógico, ambos programas corren en el mismo equipo; o físico, donde cada uno de los programas corre en equipos diferentes. Esta última es la disposición más común de encontrar.



Ventajas.

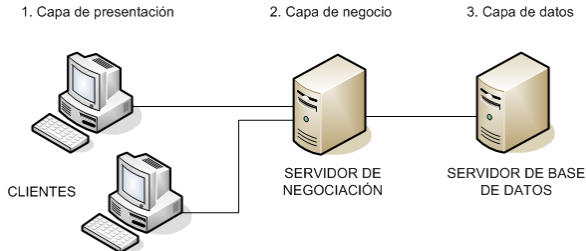
- Centralización del control: accesos, recursos e integridad de datos
- Escalabilidad independiente en cliente y servidor
- Mantenimiento más sencillo

Desventajas.

- Cuellos de botella en el servidor

Arquitectura MultiCapas

- El objetivo principal de este estilo de programación es separar la lógica de negocio de la lógica de diseño.
- Cada capa tiene asignada una funcionalidad simple e interactúa con las demás capas a través de una API. Si hay un problema con alguna capa en particular se corrige teniendo en cuenta solo de mantener la API.



El diseño más utilizado actualmente es el de 3 capas:

- **Capa de presentación:** es la que ve el usuario. Esta capa se comunica únicamente con la capa de negocios.
- **Capa de negocios:** se reciben peticiones del usuario y se da respuesta. Aquí se establecen todas las reglas que deben cumplirse a nivel de negocio.
- **Capa de datos:** es la encargada de la persistencia de datos. Recibe peticiones para almacenar y recuperar datos.

Los términos capas y niveles no significan lo mismo, una capa hace referencia a como es segmentada una solución a nivel lógico, mientras que un nivel hace referencia a la segmentación física.

Otro modelo muy usado en el desarrollo de aplicaciones web es MVC

Arquitectura Orientada a servicios (SOA)

- Utiliza servicios para dar soporte a los requisitos del negocio.
- Brinda una forma bien definida de exposición e invocación de servicios (generalmente servicios web) para facilitar la interacción entre procesos propios y de terceros
- Los servicios son debilmente acoplados y altamente escalables.

