ARQUITECTURA DE SOFTWARE

¿Qué es arquitectura?

Arquitectura

Finalmente es arte …

Pero también son las miles de decisions, tanto grandes como pequeñas.

Algunas se toman en una etapa temprana donde el impacto es profundo, y otras se dejan para después dejando las slimitaciones para una etapa futura.

Arquitectura de Software ¿Qué es?

- Las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. (SEI).
- Se considera el tipo de arquitectura que adoptará el sistema, la estructura, y las propiedades de los componentes que lo constituyen y las interrelaciones que ocurren entre sus componentes arquitectónicos.
- La arquitectura de software es la estructura o estructuras del sistema, lo que comprende a los componentes de software, sus propiedades externas visibles y las relaciones entre ellos.

Arquitectura de Software ¿Qué es?

Es una representación que permite:

- Analizar la efectividad del diseño para cumplir los requerimientos establecidos
- Considera alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavia es relativamente fácil
- Reducir los riesgos asociados con la construcción del software

¿Quién lo hace?

La arquitectura es muy importante como para dejarla en manos de una sola persona. No importa cuán brillante sea.

El ingeniero de software es quien lo hace. Sin embargo cuando un sistema es grande y complejo, el trabajo lo realizan especialistas.

Un diseñador de base de datos o daatawarehouse.
Uno o varios arquitectos de software.

¿Por qué es importante?

Cásate con tu arquitectura de prisa,
arrepiéntete en tu tiempo libre.

¿Comenzarias a construir una casa sin planos?

¿Comenzarias con los planos de la plomeria?

¿Por qué es importante?

- Las representaciones de la arquitectura de software permiten la comunicación entre todas las partes participantes interesadas en el desarrollo del software
- La arquitectura resalta las primeras decisiones que tendrán un efecto profundo en todo el trabajo de ingeniería de software siguiente, y también importante, en el éxito último del sistema como entidad operacional.
- La arquitectura “constituye un modelo relativamente pequeño y asequible por la via intelectual sobre como está estructurado el sistema y la forma en que sus componentes trabajan juntos”

¿Por qué es importante?

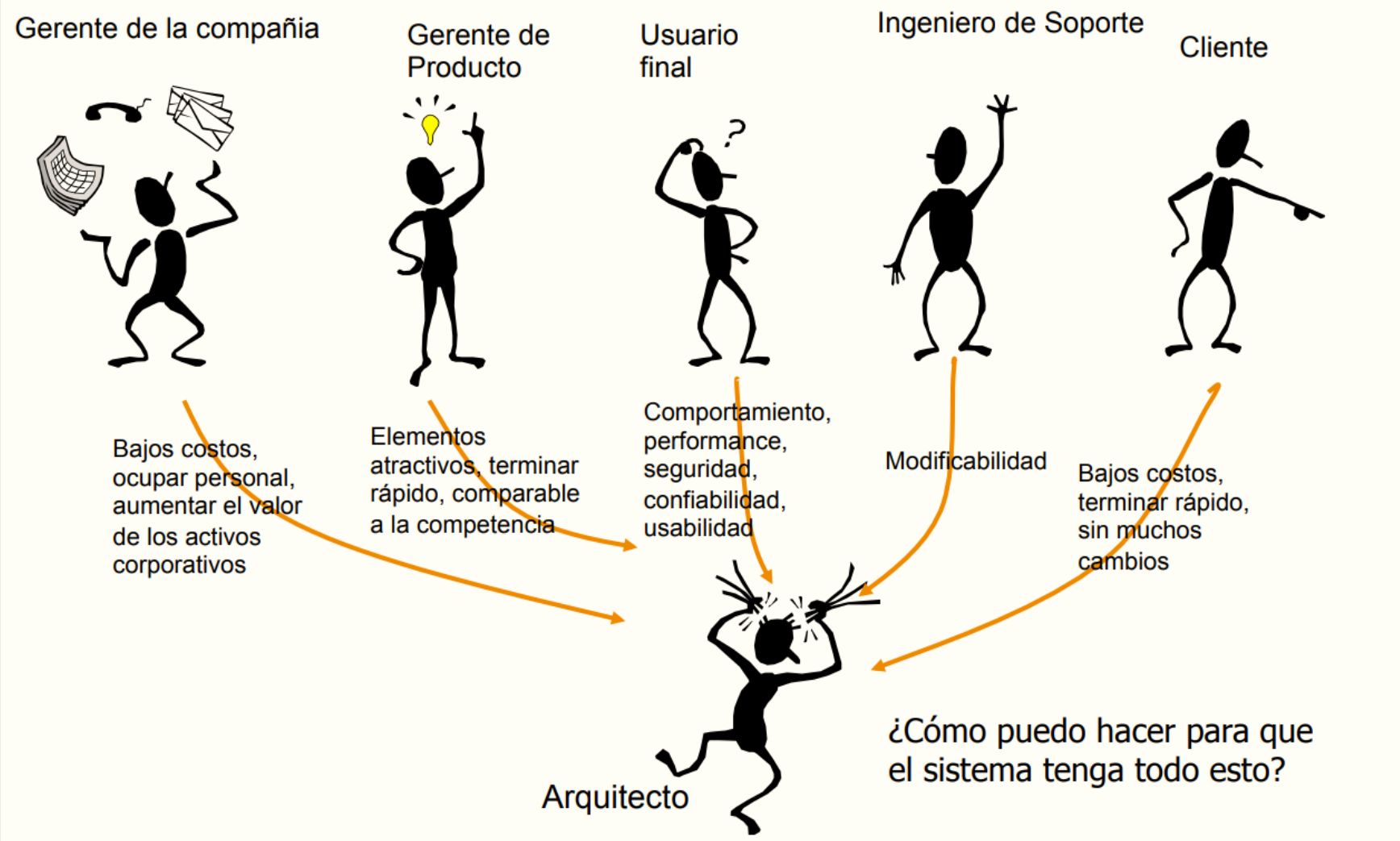
- ¿Cómo van los usuarios a utilizar la aplicación?
- ¿Cómo afectará la aplicación cuando se vaya a producción?
- ¿Cuáles son los requisitos de atributos de calidad de la aplicación, tales como la seguridad, rendimiento, concurrencia, la internacionalización y la configuración?
- ¿Cómo puede la aplicación ser diseñada para ser flexible y fácil de mantener en el tiempo?
- ¿Cuáles son las tendencias arquitectónicas que puedan afectar a la aplicación ahora o después de que se ha implementado?

¿Por qué es importante?

Tenga en cuenta que la arquitectura debe:

- Exponer la estructura del sistema, pero esconder los detalles de implementación.
- Darse cuenta de todos los casos de uso y escenarios.
- Tratar de responder a las necesidades de los diversos grupos de interés.
- Manejar los requisitos funcionales y de calidad.

¿Por qué es importante?



Interesados involucrados

- Los objetivos organizacionales y las propiedades del sistema requeridas por el negocio raramente se comprenden y menos aún se articulan completamente.
- Los requisitos de calidad del cliente casi nunca se documentan, lo cual resulta en:
  - *objetivos que no se alcanzan*
  - *conflicto inevitable entre los interesados*
- Los arquitectos deben identificar e involucrar activamente a los interesados de modo que:
  - *comprender las restricciones reales del sistema*
  - *administrar las expectativas de los interesados*
  - *negociar las prioridades del sistema*
  - *tomar decisiones de compromiso*

Interesados involucrados

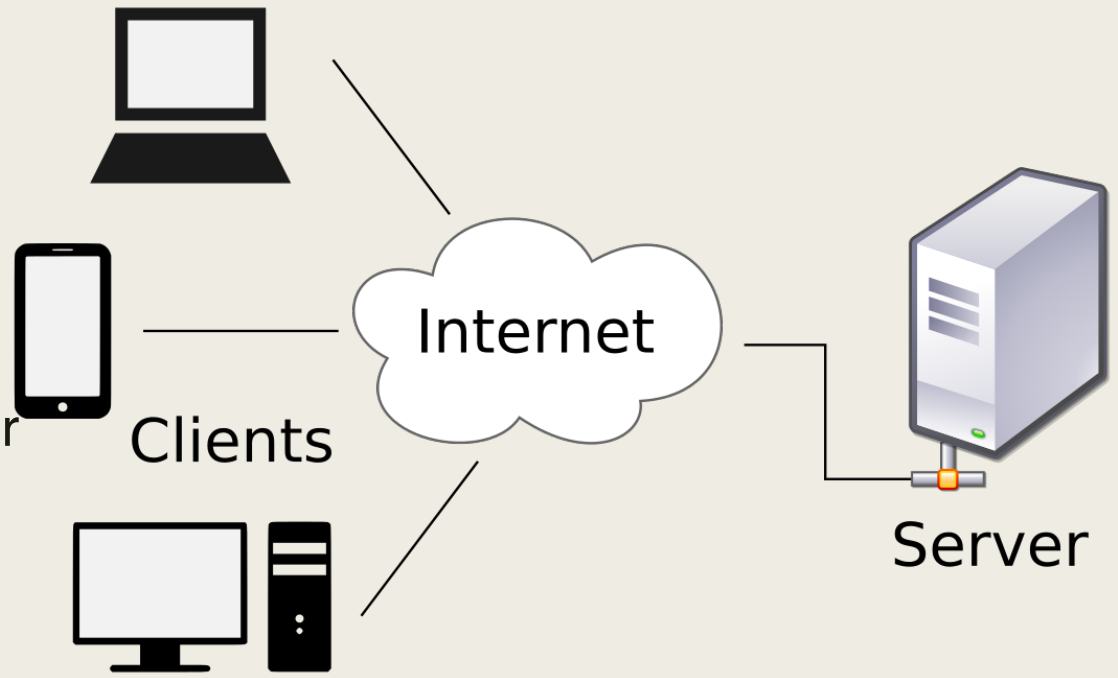
CLIENTE - SERVIDOR

Cliente - servidor

Definición

La arquitectura cliente / servidor es un modelo informático en el que el servidor aloja, entrega y administra la mayoría de los recursos y servicios que consumirá el cliente.

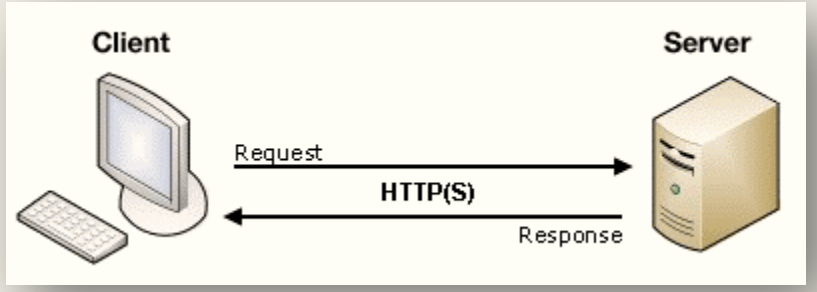
Este tipo de arquitectura tiene una o más computadoras cliente conectadas a un servidor central a través de una red o conexión a Internet. Este sistema comparte recursos informáticos.



Cliente - servidor

Desdes el punto de vista funcional:
La arquitectura cliente/servidor es una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

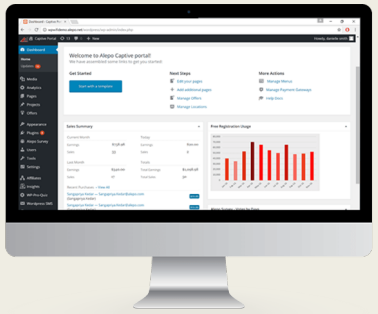
En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio)



Cliente - servidor

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son mas adecuadas a sus características.

Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PC’s a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe.



Cliente - servidor

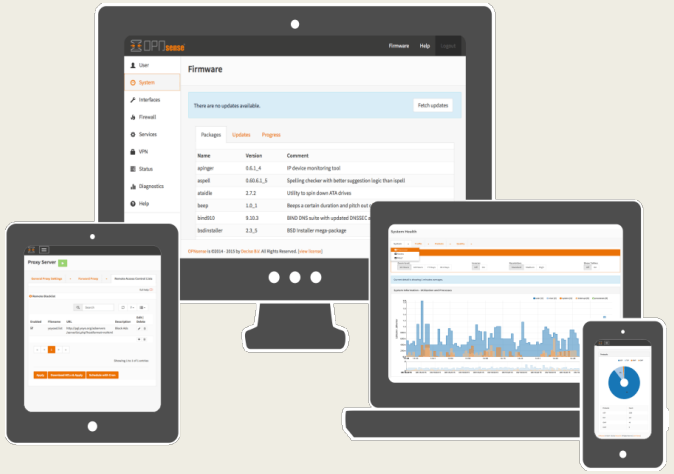
Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario (aunque esto puede variar).

En otras palabras la arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento

Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end.

El cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.



Cliente

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end.

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Servidor

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para trasmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Arquitectura en capas

El aumento de las computadoras personales en las empresas durante la década de 1980 y la mayor fiabilidad del hardware de red fueron causa que los sistemas de dos capas y de tres capas se convirtieran en algo común.

En un sistema de dos niveles, se requiere un software diferente para el servidor y para el cliente. Ilustra el modelo de servidor de cliente de dos niveles. En las primeras etapas, el modelo de cómputo del servidor del cliente se llamaba modelo de computación de dos niveles en el que el cliente se considera como nivel de captura y validación de datos y el servidor se consideraba como nivel de almacenamiento de datos. Este escenario está representado.

Arquitectura en capas

La necesidad de escalabilidad empresarial desafió este modelo tradicional de cliente y servidor de dos niveles. A mediados de la década de 1990, cuando la aplicación se volvió más compleja y pudo implementarse en cientos o miles de usuarios finales, el lado del cliente ahora sufría los siguientes problemas:

- Un cliente robusto que requiere recursos considerables en la computadora del cliente para funcionar de manera efectiva. Esto incluye espacio en disco, RAM y CPU.
- Las máquinas cliente requieren administración, lo que genera costos mas altos.

Arquitectura en capas

En 1995, la arquitectura de tres niveles aparece como una mejora sobre la arquitectura de dos niveles.

Tiene tres capas, que son:

- Primera capa: interfaz de usuario que se ejecuta en la computadora del usuario final (el cliente).
- Segunda capa: Servidor de aplicaciones Es una lógica de negocios y capa de procesamiento de datos. Este nivel medio se ejecuta en un servidor que se llama servidor de aplicaciones.
- Tercera capa: servidor de base de datos Es un DBMS, que almacena los datos requeridos por el nivel medio. Este nivel puede ejecutarse en un servidor separado llamado servidor de base de datos.

Arquitectura en capas

Como se describió anteriormente, el cliente ahora es responsable de la interfaz de usuario de la aplicación, por lo que requiere menos recursos computacionales, ahora los clientes se llaman 'thin client' y requiere menos mantenimiento.

Ventajas de la arquitectura en capas

- El entorno cliente/servidor facilita el trabajo más productivo de los usuarios y hace un mejor uso de los datos existentes.
- Flexibilidad en el sistema de base de datos cliente / servidor.
- El tiempo de respuesta y el rendimiento es alto.
- El servidor de base de datos se puede personalizar a los requerimientos del DBMS y así proporcionar un mejor rendimiento.
- El cliente podría ser una estación de trabajo personal, adaptada a las necesidades de los usuarios finales y, por lo tanto, capaz de proporcionar mejores interfaces, alta disponibilidad, respuestas más rápidas y facilidad de uso general mejorada para el usuario.
- Una única base de datos (en el servidor) se puede compartir a través de varios sistemas distintos de clientes (aplicaciones).

Arquitecturas más utilizadas

- Microservicios
- En capas (4):
  - Capa de presentación (también conocida como capa UI)
  - Capa de aplicación (también conocida como capa de servicio)
  - Capa de lógica de negocios (también conocida como capa de dominio)
  - Capa de acceso a datos (también conocida como capa de persistencia)
-