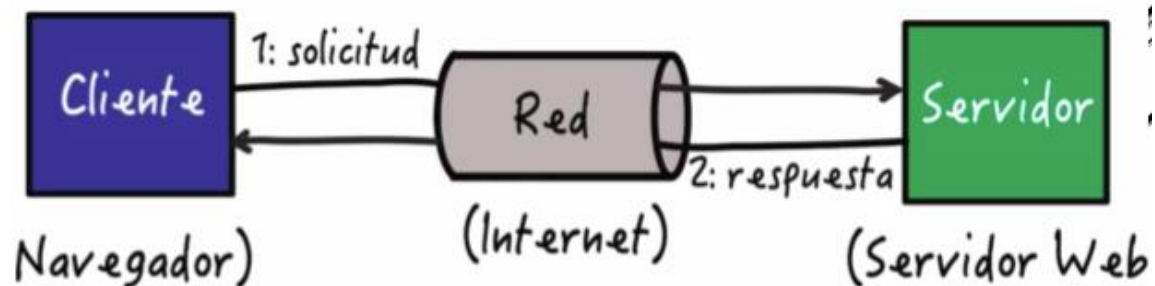


Nombre: Ricardo Antonio De los Santos Manzueta

Matricula: 100306344

Sección: 01

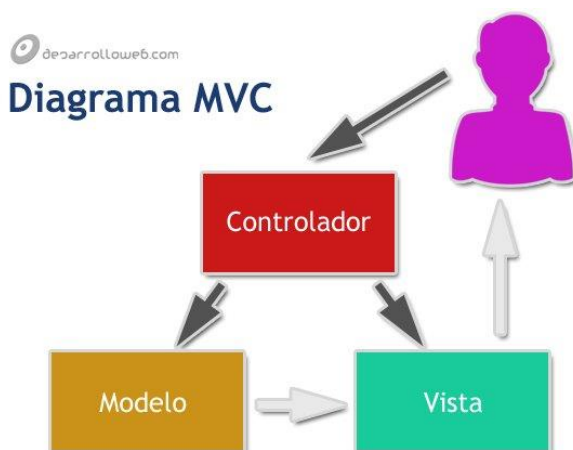
Arquitectura Cliente – Servidor



El cliente accede al servidor por medio de un navegador. Hace peticiones y espera las respuestas del servidor. La red gestiona la transferencia de datos, por medio de HTTP.

En esta arquitectura el cliente suele ser estaciones de trabajo que solicitan varios servicios al servidor, mientras que un servidor es una máquina que actúa como depósito de datos y funciona como un sistema gestor de base de datos, este se encarga de dar la respuesta demandada por el cliente.

Modelo – Vista – Controlador (MVC)

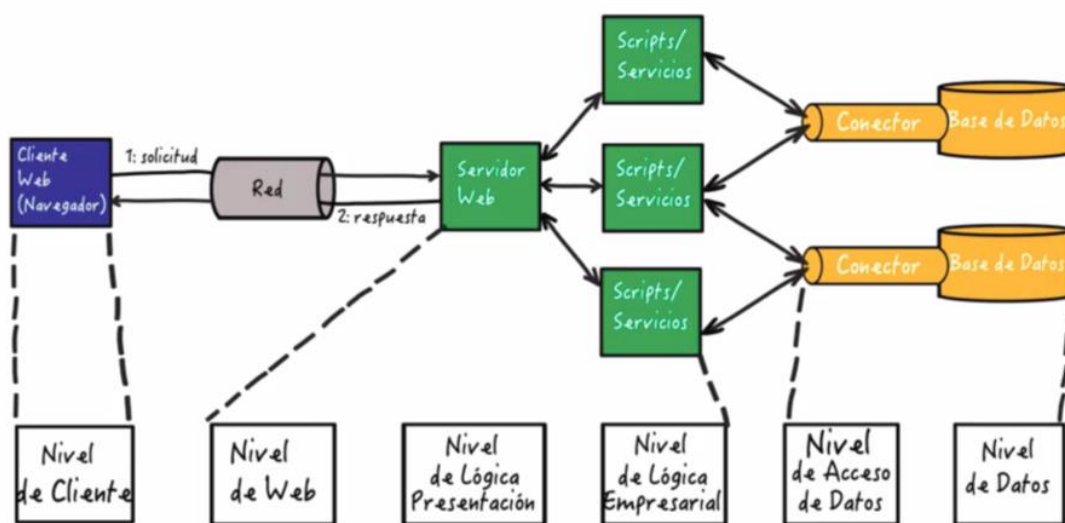


Este modelo separa la capa de datos, la capa visual y la capa de control. Cada capa debe ser independiente de otra.

En nuestro caso, la vista sería la página web (HTML), el modelo sería el software que ya existía (programa en JAVA) y el controlador sería el intermediario que debemos crear para conectar una con otra.

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Arquitectura de 6 capas



Este modelo es algo similar a una ampliación del anterior. Detalla de manera más específica los componentes del modelo de 3 capas.

- **Divide la capa de Vista en dos sub-capas:**
 - **Nivel de cliente:** componentes de interfaz del lado del cliente.
 - **Nivel de lógica de presentación:** programas del lado del servidor para generar páginas web.

- **Divide la capa de Modelo en dos sub-capas:**
 - **Nivel de datos:** los datos usados por la aplicación.
 - **Nivel de acceso a datos:** accede los datos y los pasa a la capa de controlador.

- **Divide la capa de Controlador en dos sub-capas:**
 - **Nivel de lógica de negocio:** modela los objetos de negocio.
 - **Nivel de web:** el servidor web.

Ventajas y desventajas de la arquitectura en capas

Ventajas

- **Modularidad:** facilita el desarrollo, la prueba y el mantenimiento de cada capa de forma independiente.

- **Separación de responsabilidades:** permite la separación de responsabilidades y una mayor claridad en la arquitectura, ya que cada capa tiene una función específica.

- **Reutilización:** posibilita que las capas se reutilicen en diferentes proyectos, lo que reduce el tiempo y los costos de desarrollo.

- **Escalabilidad:** significa que se pueden agregar o quitar capas según las necesidades del sistema.

- **Flexibilidad:** es adaptable a diferentes entornos y requisitos del sistema.

Desventajas:

- **Complejidad:** aumenta la complejidad del sistema debido a la necesidad de interfaces y comunicación entre las capas.
- **Tiempo y esfuerzo de desarrollo:** requiere más tiempo y esfuerzo de desarrollo debido a la necesidad de diseñar, desarrollar y probar cada capa por separado.
- **Consumo de recursos:** aumenta el consumo de recursos del sistema, como el uso de memoria y CPU, debido a la necesidad de comunicación entre las capas.
- **Costos:** eleva los costos del desarrollo debido a la necesidad de diseñar y desarrollar varias capas separadas.
- **Rendimiento:** disminuye el rendimiento del sistema debido a la necesidad de comunicación entre las capas.