MEDAC
Instituto Oficial de Formación Profesional

Representación y explicación de arquitecturas cliente-servidor

Esther Ruiz Rodríguez

2º WEB

Índice

El objetivo de esta tarea es comprender y representar gráficamente los diferentes tipos de arquitectura cliente-servidor que existen en el desarrollo de aplicaciones web.

1.Investigar los conceptos de arquitectura de dos capas, tres capas y n capas, basándote en los apuntes del tema y fuentes complementarias fiables.

2.Realizar un diagrama para cada tipo de arquitectura utilizando una herramienta

3.Para cada diagrama, redacta un breve texto (máximo 8-10 líneas) en el que describas:

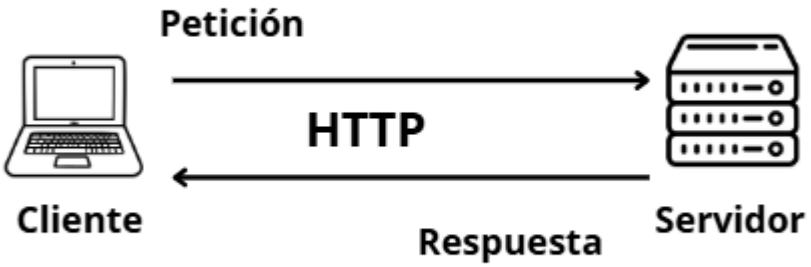
a.Cómo fluye la información entre los diferentes componentes (cliente, servidor, base de datos, etc.).

b.Qué ventajas y desventajas tiene cada tipo de arquitectura.

c.Ejemplos de aplicaciones reales o servicios web que usen esa arquitectura.

4.Entregar PDF con los diagramas y los textos.

Arquitectura de dos capas:



a. **a.El flujo de información de la arquitectura de dos capas se basa en:**

   El cliente hace una petición al servidor. El servidor procesa la solicitud y accede a los recursos necesarios. El servidor responde y envía al cliente la respuesta.

b. **Ventajas - Desventajas:**

Ventajas: Las herramientas para el desarrollo con dos capas son robustas y ampliamente evaluadas.

Desventajas: Requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios.

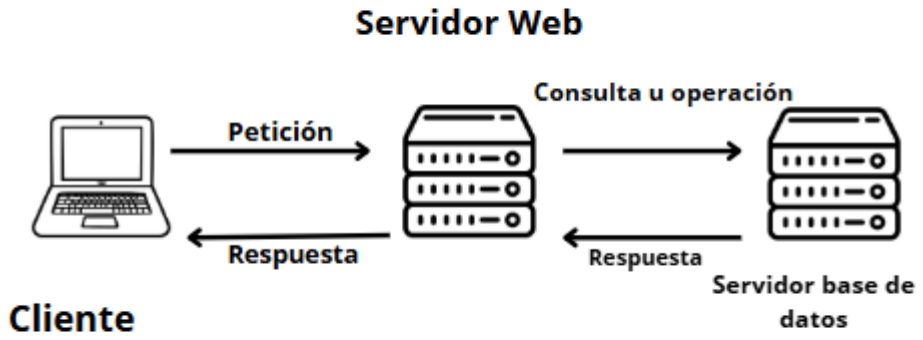
La seguridad del sistema es compleja y a menudo requiere administración de las bases de datos.

c. **Ejemplos de aplicaciones:**

Outlook, Thunderbird (Correos electrónicos)

Access , conectado a SQL Server (Sistema de gestión de bases de datos)

Arquitectura de 3 capas:



a. **El flujo de información de la arquitectura de tres capas se basa en:**

El cliente enviará una solicitud al servidor web. El servidor web recibe la solicitud del cliente, la procesa y decide qué hacer. El servidor web envía y consulta al Servidor de base de datos. El Servidor de Base de datos responde con los datos solicitados. El

servidor Web procesa los datos recibidos y prepara la respuesta. El servidor Web devuelve la respuesta y muestra la información al Cliente.

b. Ventajas - Desventajas:

Ventajas:

La separación de responsabilidades. Cada capa tiene sus funciones específicas.

Tiene mejor seguridad, ya que la capa lógica (Servidor Web) controla el acceso a los datos, evitando que el cliente interactue directamente con la base de datos.

El mantenimiento es más sencillo. Al estar modularizada, los errores y las mejoras son más fáciles de localizar y aplicar.

Desventajas:

Mayor complejidad. Requiere un diseño y desarrollo más estructurado.

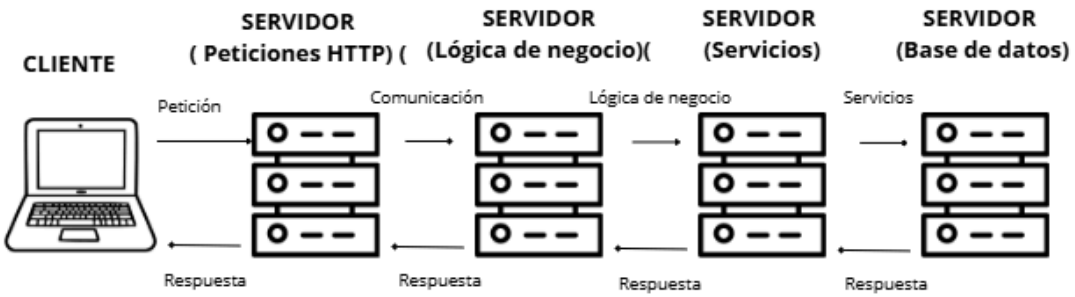
El costo. Puede implicar mayor inversión en infraestructura y recursos.

El rastreo de errores puede ser más difícil al haber múltiples capas.

c. Ejemplos de aplicaciones:

Amazon, Netflix, Facebook, Spotify, Salesforce (CRM).

Arquitectura de n capas:



a. El flujo de información de la arquitectura de n capas:

El cliente realiza una acción (ej: solicitar perfil, hacer login). El cliente envía la petición al servidor web.

El servidor web recibe la solicitud, la autentica, valida y decide a qué parte de la lógica de negocio debe enviarla.

En la capa 3,aquí se aplican las reglas y decisiones del negocio. Si se necesitan datos, esta capa coordina con la capa de servicios.

En la capa 4, se gestionan los microservicios o módulos especializados que hacen tareas específicas, como llamadas a bases de datos, cache, servicios externos o procesos adicionales.

En la capa 5 (BD), es donde se almacenan y recuperan los datos definitivos.

b. Ventajas - Desventajas:

Ventajas:

- Gran modularidad y separación de responsabilidades.
- Facilita escalabilidad horizontal: cada capa puede escalarse o desplegarse de forma independiente.
- Mejor control de seguridad, porque las capas intermedias pueden validar y filtrar las solicitudes.
- Soporta arquitecturas modernas (microservicios, APIs, servicios externos).

Desventajas:

- Complejidad considerable: requiere mayor planificación y coordinación.
- Latencia aumentada por la comunicación entre varias capas.
- Dificultad para detectar y depurar errores, porque puede involucrar varios componentes.
- Más recursos e infraestructura necesaria para desplegar y administrar las capas.

c. Ejemplos:

Uber (plataforma de transporte)

Cliente: Web y apps móviles.

Comunicación: API Gateway con autenticación.
Lógica de negocio: Gestión de viajes, usuarios, precios dinámicos.
Servicios: Microservicios para geolocalización, pagos, notificaciones.
Base de datos: MySQL, Redis, sistemas NoSQL.