

Representación y explicación de arquitecturas cliente servidor

Trabajo realizado por Jesus
Fernandez Rodriguez.



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. Conceptos de arquitectura

- **Arquitectura de dos capas:** es un modelo más simple donde el cliente (capa de presentación) y el servidor (datos/BD) interactúan directamente.

La lógica de negocio suele estar contenida en uno de los dos lados. El cliente accede prácticamente de forma directa a la base de datos.

- **Arquitectura de tres capas:** cada capa tiene responsabilidades bien definidas y se comunican de forma ordenada. Estas son:

Presentación (Cliente): lo que el usuario ve e interactúa (páginas web, apps).

Lógica de negocio (Aplicación): procesa reglas del negocio, coordina operaciones.

Datos: capa encargada de guardar y gestionar la información.

- **Arquitectura de n capas:** extensión de la de tres capas, con más niveles intermedios que mejoran rendimiento, seguridad o modularidad.

Algunas capas adicionales pueden ser: API, balanceador de carga, servicios de integración, microservicios especializados, etc.



2. Arquitectura de dos capas

En este modelo, el cliente se comunica directamente con el servidor de base de datos.

Flujo de información:

- 1- El cliente solicita información (ejemplo: “mostrar lista de productos”).
- 2- El servidor consulta la base de datos y responde.
- 3- El cliente muestra la información.

Ventajas:

Diseño simple y económico.
Requiere menos recursos de infraestructura.

Desventajas:

Dificultad para crecer con muchos usuarios.
Baja seguridad: cliente muy próximo a la base de datos.
Poco flexible ante cambios.

Ejemplo: Una **web de reservas de restaurante**: el usuario entra en la página, rellena un formulario con su nombre, fecha y número de personas. El navegador envía esos datos directamente al servidor, que los guarda en la base de datos.

Cuando el usuario consulta sus reservas, el navegador pide la información y la base de datos responde sin que exista un servidor de aplicaciones intermedio.



3. Arquitectura de tres capas

El sistema se divide en cliente, servidor de aplicaciones y servidor de base de datos. El servidor intermedio contiene la lógica de negocio y evita que el cliente acceda directamente a los datos.

Flujo de información:

- 1- El cliente pide información (ejemplo: catálogo).
- 2- El servidor de aplicaciones procesa la solicitud.
- 3- Este consulta la base de datos.
- 4- La respuesta se devuelve al cliente organizada.

Ventajas:

Separación clara entre presentación, lógica y datos.
Mayor seguridad y mantenimiento sencillo.
Escalabilidad adecuada.

Desventajas:

Más compleja que la de dos capas.
Requiere más recursos de servidor.

Ejemplo: Una **tienda online como Amazon**: el usuario entra desde su navegador y busca un producto (capa cliente). La petición llega al servidor de aplicaciones, que aplica la lógica de negocio (verifica disponibilidad, calcula precios y organiza los resultados). Luego, el servidor consulta la base de datos y devuelve la información al cliente ya procesada y presentada en la web.



4. Arquitectura de n capas

Es una evolución de la de tres capas. Agrega niveles especializados, como balanceadores, APIs, microservicios o capas de autenticación, que permiten mejorar la escalabilidad y la seguridad.

Flujo de información:

- 1- El cliente realiza la petición.
- 2- La solicitud pasa por diferentes servicios especializados (autenticación, API).
- 3- Cada capa procesa su parte hasta llegar a la base de datos.
- 4- La respuesta se reconstruye y se devuelve al cliente.

Ventajas:

Muy escalable y flexible.

Permite modularidad y especialización.

Desventajas:

Desarrollo y mantenimiento complejos.

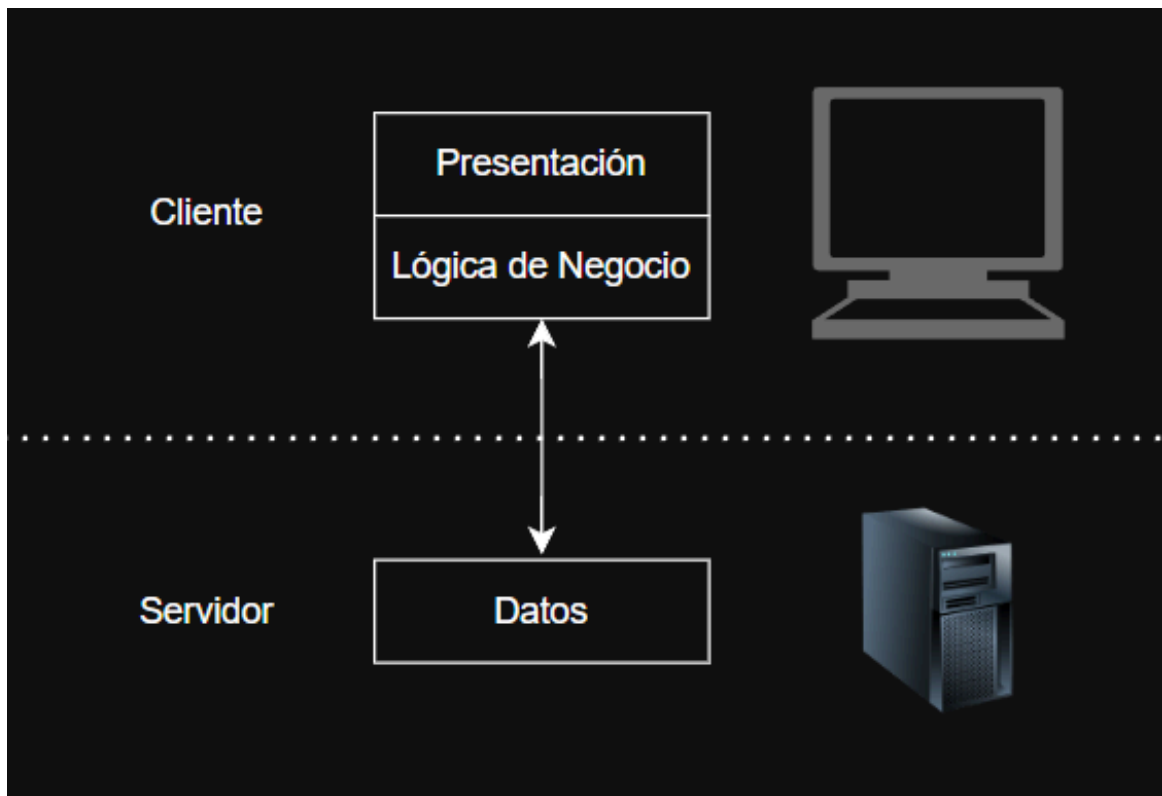
Mayor consumo de recursos e infraestructura.

Puede generar latencia por tantos pasos intermedios.

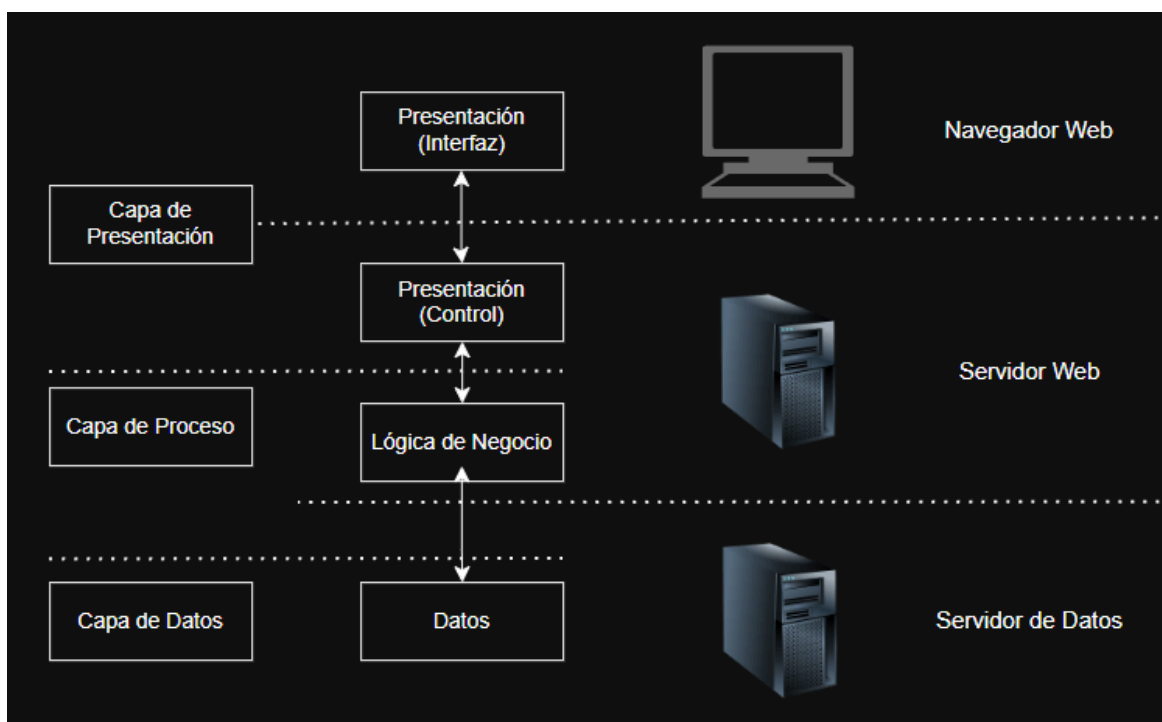
Ejemplo: Un servicio como **Netflix**: el usuario abre la app web y solicita una película (cliente). La petición pasa por la **capa de autenticación** que valida la cuenta, luego por una **API de recomendación** que analiza el historial, y finalmente llega a los **servidores de streaming**, que envían el vídeo desde el servidor más cercano. Balanceadores de carga reparten las conexiones entre millones de usuarios y, al final, la película se reproduce en el navegador.

5. Diagrama de cada arquitectura.

- Arquitectura de 2 capas:

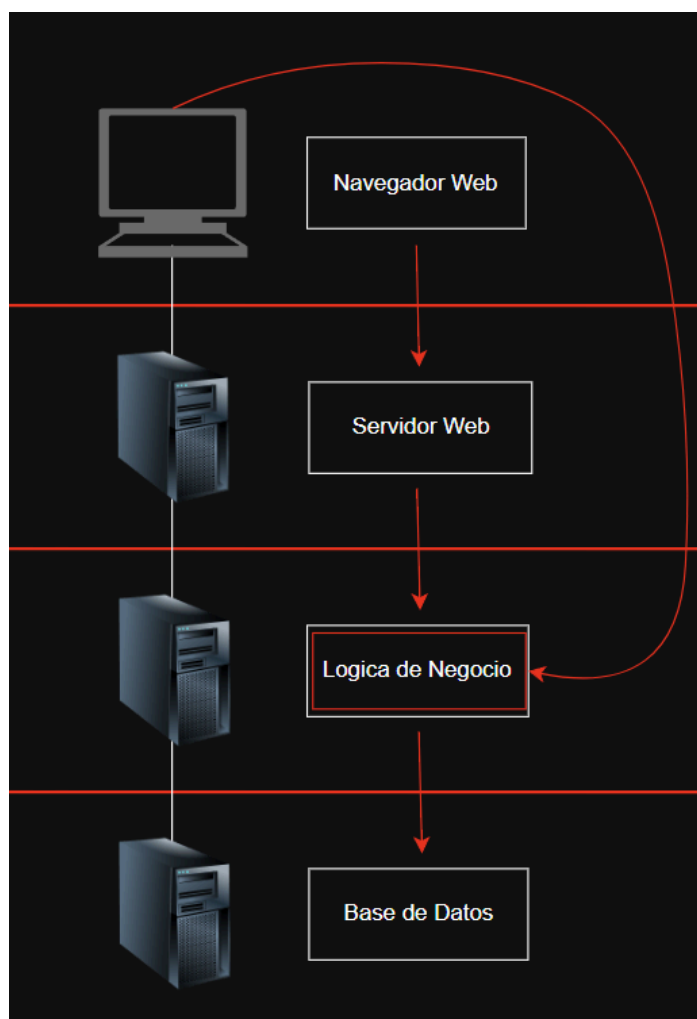


- Arquitectura de 3 capas:





- Arquitectura de n capas:





6. Comparación general

Característica	Dos capas	Tres capas	N capas
Componentes principales	Cliente y base de datos	Cliente, servidor de aplicaciones, base de datos	Cliente, múltiples capas intermedias y base de datos
Flujo de información	Directo cliente = base de datos	Cliente = lógica = base de datos	Cliente = varias capas = base de datos
Ventajas	Simplicidad, bajo coste	Seguridad, mantenimiento más fácil	Escalabilidad, flexibilidad, modularidad
Desventajas	Baja seguridad, poca escalabilidad	Mayor complejidad, más recursos necesarios	Complejidad elevada, latencia, alto consumo de recursos
Ejemplo web	App de reservas local	Amazon, eBay	Netflix, Google



7. Reflexión final

El diseño de una arquitectura cliente-servidor determina la capacidad de una aplicación web para crecer, mantenerse y garantizar la seguridad de los datos.

La **arquitectura de dos capas** puede ser útil en proyectos pequeños pero se queda corta cuando aumenta el número de usuarios.

La **de tres capas** se adapta mejor a aplicaciones web medianas y grandes.

La **arquitectura de n capas** representa la evolución en entornos modernos, con microservicios y APIs que permiten a grandes compañías como Netflix o Google procesar millones de peticiones simultáneas y evolucionar sin detener sus servicios.

Bibliografía

<https://tec755.wordpress.com/infografia/>

<https://es.scribd.com/doc/279764551/Arquitecturas-de-2-y-3-Capas>

Temario de clase