

# Investigación de arquitecturas de Cliente Servidor

## Arquitecturas de dos capas

Consiste en la capa Cliente-Aplicación y la capa Base de datos. El servidor Cliente-Aplicación se comunica directamente con el servidor de la Base de datos. La transferencia de datos o información entre los dos componentes es rápida debido a la ausencia de un *middleware*.

La aplicación del cliente contiene los códigos para interactuar con el usuario y también para guardar datos en el servidor de la base de datos. La aplicación cliente envía la solicitud al servidor y procesa la solicitud y la devuelve con datos. Esto significa que la Aplicación cliente maneja tanto la capa de Presentación (interfaz de la aplicación) como la capa de la Aplicación (operaciones lógicas).

Por ejemplo, las aplicaciones que se encuentran instaladas en las computadoras de algunas bibliotecas para realizar la búsqueda de libros utilizan este tipo de arquitectura; ya que en la aplicación del cliente realizan todo el procesamiento y operaciones (como ordenar los libros alfabéticamente, por año, etcétera) y obtienen los datos directamente del servidor con la base de datos.

## Arquitecturas de tres capas

Es una arquitectura modular cliente-servidor que consta de un nivel de presentación (cliente), un nivel de aplicación y un nivel de datos.

La capa de presentación es la capa superior de una aplicación: Esta capa de interfaz traduce tareas e información a algo que el usuario puede entender. También toma la solicitud del usuario, la envía a la capa de aplicación donde se procesa y el resultado se envía nuevamente al usuario.

La capa de aplicación es la capa intermedia: Contiene la lógica de negocio y operaciones lógicas para realizar tareas como procesar comandos, validación, evaluaciones lógicas, toma de decisiones, interacciones de bases de datos y estructuración de aplicaciones web.

La capa de datos es la capa de fondo: Incluye los mecanismos de persistencia de datos (servidores de bases de datos, archivos compartidos, etc.) y la capa de acceso a datos que encapsula los mecanismos de persistencia y expone los datos.

Por ejemplo, páginas y aplicaciones web modernas como Reddit en el que se tiene una capa de presentación que es la interfaz con la que interactúa el usuario y manda solicitudes de la acción que quiere ser realizada por este. Dicha solicitud es recibida por la capa de aplicación que se encarga de realizar todas las interacciones con la base de datos y procesamiento, para después mostrarle al usuario el contenido que solicito ver en la capa de presentación.

### **Arquitecturas de tres capas con tecnología de monitoreo en el procesamiento de transacciones**

Se utiliza la misma arquitectura de tres capas y se agrega un monitor de procesamiento de transacciones el cual se es un programa que monitorea las transacciones de una etapa a la siguiente, asegurando que cada una se complete con éxito; de lo contrario, o si ocurre un error, el monitor toma la acción apropiada. El objetivo / objetivo principal de un monitor de procesamiento de transacciones es permitir el intercambio de recursos y garantizar el uso óptimo de los recursos por parte de las aplicaciones.

Un monitor de procesamiento de transiciones es crítico en arquitecturas de niveles múltiples. Con los procesos que se ejecutan en diferentes plataformas, una transacción determinada se puede reenviar a cualquiera de varios servidores. En general, el monitor maneja todo el equilibrio de la carga. Después de completar cada transacción, puede procesar otra transacción sin ser influenciado por la transacción anterior.

En general, agregar un monitor de procesamiento de transiciones proporciona la siguiente funcionalidad:

- Coordinar recursos
- Crear nuevos procesos cuando sea necesario
- Proporcionar acceso seguro a los servicios.
- Servicios de enrutamiento
- Empaquetado mensajes de datos en mensajes
- Desempaquetado de mensajes en paquetes / estructuras de datos
- Monitoreo de operaciones / transacciones
- Administrar colas
- Manejo de errores a través de acciones como el reinicio del proceso
- Ocultar detalles de comunicaciones entre procesos

Por ejemplo, servicios de streaming como GeForce Now de Nvidia en el que se tiene un monitoreo del proceso de transacciones para administrar colas para el uso de sus servicios, así como para coordinar sus recursos, asegurar su uso óptimo y disponibilidad.