# Investigación – Cliente Servidor

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

En resumen, es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes, resultan en un trabajo realizado por otros computadores llamados servidores".

## Arquitectura Cliente Servidor

Es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

### Arquitecturas multicapas

#### Arquitectura de 2 capas

La arquitectura en 2 niveles o 2 capas se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

##### Características

**El cliente se caracteriza por:**

* Es el que inicia solicitudes o peticiones. Tiene, por tanto, un papel activo en la comunicación (dispositivo maestro).
* Espera y recibe las respuestas del servidor.
* Por lo general, puede conectarse a varios servidores a la vez.
* Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

**El servidor se caracteriza por:**

* Al iniciarse espera a que le lleguen las solicitudes de los clientes. Desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
* Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
* No es frecuente que interactúen directamente con los usuarios finales.

##### Ventajas

* **Centralización del control:** Los accesos, recursos y la integridad de los datos son controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
* **Tecnologías:** Existen algunas suficientemente desarrolladas, diseñadas para el paradigma de C/S, que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz y la facilidad de empleo.

##### Desventajas

* Es difícilmente escalable
* Número de conexiones reducida
* Alta carga de la red.
* La flexibilidad es restringida
* La funcionalidad es limitada.

##### Ejemplo

La página web de la facultad

#### Arquitectura de 3 capas

Es un tipo de arquitectura usada en la gran mayoría de sistemas. Se suele usar en sistemas que implementan un modelo de negocio como podría ser una tienda online, una aplicación para gestionar ciertos datos, etc. En la arquitectura en tres niveles existe un nivel intermedio. Esto significa que la arquitectura generalmente está compartida.

##### Componentes

* **Un cliente:** Es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
* **El servidor de aplicaciones** **(también denominado software intermedio)**: Cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
* **El servidor de datos:** que proporciona al servidor de aplicaciones los datos que éste le solicitó.

##### Características

* Centraliza la gestión de la regla del negocio en un único lugar (no se duplica en cada aplicación)
* Los clientes pidan o envíen información a esta aplicación centralizada, no al gestor de base de datos en el servidor
* Esta aplicación centralizada que conforma una nueva capa dentro de un sistema Cliente-Servidor, se conoce como capa intermedia o middle-tier.

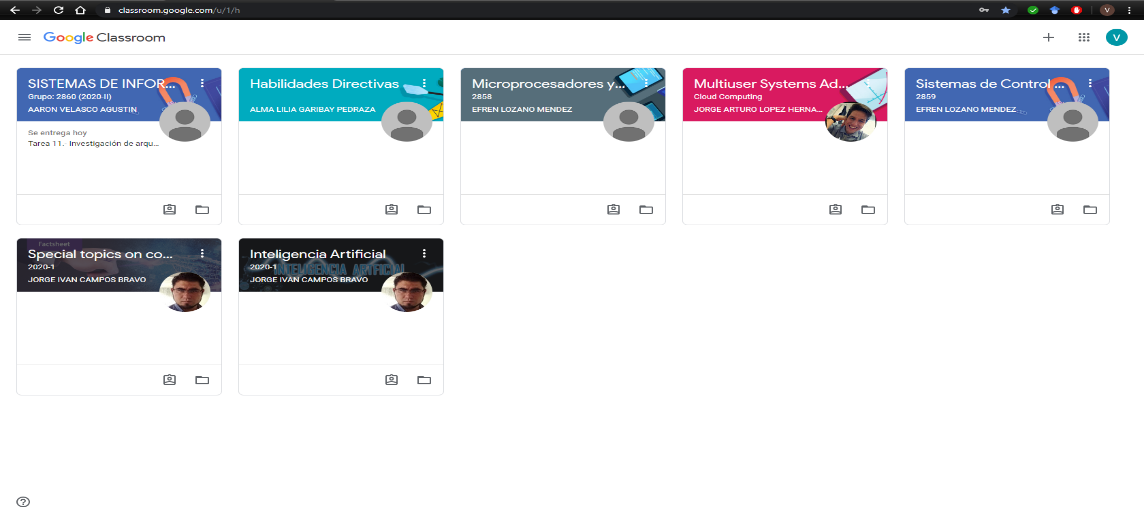
##### Ventajas

* Un mayor grado de flexibilidad.
* Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
* Mejor rendimiento, ya que las tareas se comparten entre servidores.

##### Desventajas

* Pueden incrementar el tráfico en la red cuando muchos clientes envían peticiones a un solo servidor.
* Requiere más balance de carga y tolerancia a las fallas.
* Los exploradores actuales no son todos iguales.

##### Ejemplo



Classroom

##### Monitores Transaccionales

Son sistemas especializados en la creación, ejecución y manejo de aplicaciones de procesamiento de transacciones, los monitores son tecnologías que se especializan en la administración de transacciones desde un punto de origen, y a través de uno o mas servidores, para luego volver al origen.

###### Funciones y Operaciones

* **Control de procesos**
* **Manejo de transacciones**
* **Comunicación cliente/servidor**

###### Tipos de Monitores

1. **OTM (Object Transaccion Monitor)**
   1. Maneja contenedores que corren los componentes que brindan servicio
   2. Manejo óptimo de objetos
   3. Distribuye la carga y coordina transacciones multicomponentes
2. **TP**

###### Ejemplo

Pagina de reservaciones de vuelos aeromexico