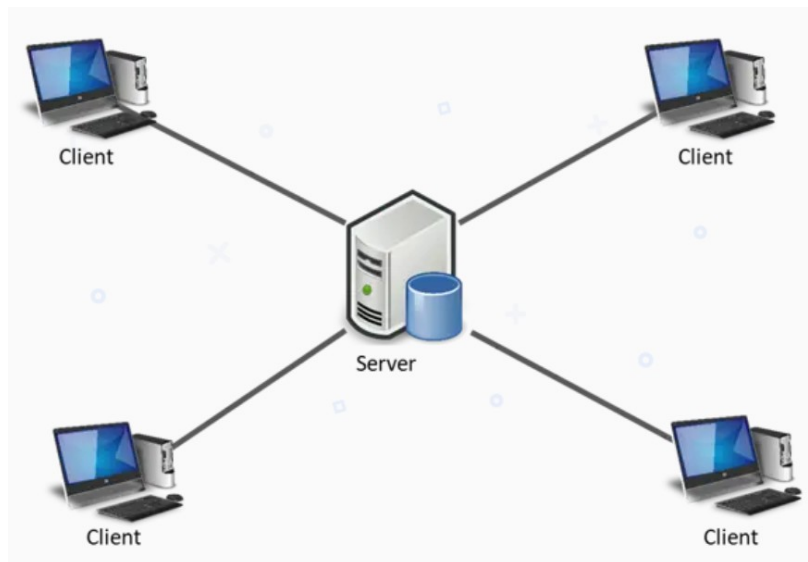


## Arquitectura cliente/servidor

La arquitectura **cliente-servidor** tradicional es considerada una arquitectura distribuida en la que participan dos componentes: uno es el cliente que utiliza unos servicios, y otro es el servidor que proporciona esos servicios. Entre ambos se tiene que efectuar una comunicación de red, habitualmente mediante Internet. Las aplicaciones cliente/servidor normalmente distribuyen los componentes de la aplicación de forma que la base de datos reside en el servidor, el interfaz de usuario reside en el cliente, y la lógica de negocio puede residir tanto en el cliente (en forma de código), como en el servidor (en forma de procedimientos almacenados) o en ambos.

En una arquitectura Cliente-Servidor existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar, en este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado. Una de las complicaciones es gestionar las actualizaciones en los clientes, pues puede haber muchos terminales con el cliente instalado y tenemos que asegurar que todas sean actualizadas cuando salga una nueva versión.



Ejemplos: Outlook, WhatsApp.

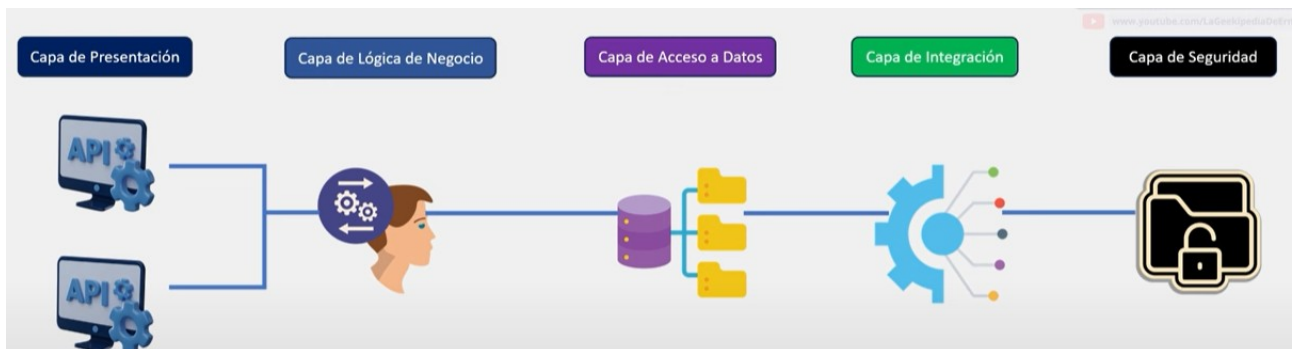
Bajo este modelo, podemos ver variaciones en la construcción de la arquitectura de los sistemas. Este tipo de arquitecturas son conocidas como *arquitectura por capas*. La arquitectura en capas es una de las técnicas más comunes que los desarrolladores de software utilizan para separar un sistema software. Consiste en dividir la aplicación en capas, con la intención de que cada capa tenga un rol muy definido, como podría ser, una capa de presentación (UI), una capa de reglas de negocio (servicios) y una capa de acceso a datos (DAO), sin embargo, este estilo arquitectónico no define cuantas capas debe de tener la aplicación, sino más bien, se centra en la separación de la aplicación en capas.

La **arquitectura cliente-servidor en tres capas** dividen la arquitectura del sistema en tres capas lógicas:

- Capa de presentación: Interfaz de usuario. Recibe peticiones de los usuarios y muestra información al usuario llamando a la capa lógica para lo que necesiten.
- Capa de lógica de negocio: contiene los servicios que necesita la capa de presentación. Utiliza la capa de acceso a datos para llamar a la BD. Nunca llama a la capa de presentación.
- Capa de acceso a datos: Almacena y accede a los datos. No llama a la capa lógica.

En una arquitectura en capas, todas las capas se colocan de forma horizontal, de tal forma que cada capa solo puede comunicarse con la capa que está inmediatamente por debajo, por lo que, si una capa quiere comunicarse con otras que están mucho más abajo, tendrán que hacerlo mediante la capa que está inmediatamente por debajo. Por ejemplo, si la capa de presentación requiere consultar la base de datos, tendrá que solicitar la información a la capa de negocio, la que a su vez, la consultará a la base de datos, finalmente, la respuesta retornará en el sentido inverso hasta llegar la capa de presentación.

Adicionalmente están las capas de integración y de seguridad.



Este modelo de arquitectura en 3 capas hace más robustas las aplicaciones, ya que el cliente se mantiene aislado de los cambios en el resto de la aplicación. La capa de interfaz se comunica solo con la capa de negocio, nunca directamente con la capa de base de datos. La capa de lógica de negocio se comunica por un lado con la capa de interfaz y por otro lado con la capa de acceso a la base de datos. De esta forma, un cambio en la base de datos no afecta a la parte cliente.

Esta arquitectura nos permite **reutilizar todo el código** si se cambia de interfaz, ya que sólo habría que tocar esa capa. Las demás, si se ha diseñado todo correctamente, no tendrían que modificarse. Ya que el resto del sistema nunca hace uso de la capa de presentación.

Igualmente, si deseamos cambiar el acceso a datos o la forma en la que se persisten, habría que modificar únicamente dicha capa, y puede que algunas llamadas desde la capa de la lógica de negocio. Pero no sería necesario modificar nada de la interfaz o el modelo.

**MVC: Modelo – Vista - Controlador**

Normalmente, la capa de presentación suele implementarse utilizando el modelo MVC (modelo-vista-controlador) para esa capa. Propone el uso de 3 componentes o módulos:

- **Vista:** la interfaz de usuario pura: HTML, o FXML, o JPanel de Swing, o la consola de usuario
- **Controlador:** Responde a las peticiones de la vista, y las atiende llamando a la capa de servicios. Ejemplo: cuando se hace click en algún botón o cuando por ejemplo se accede a alguna página web.
- **Modelo:** Contiene la lógica de negocio, accesos a BD, validaciones,...

