Capas

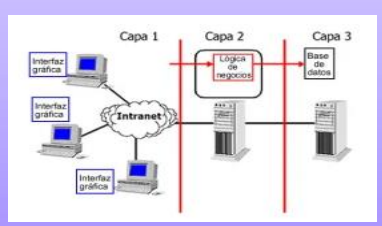
Es pensar nuestro sistema en capas y cada capa debe exponer en forma clara las operaciones que puede realizar. Estas operaciones se deben exponer mediante un API que nos digan qué servicio me ofrece esa capa y cuál será su retorno sin importar como este implementado

2 Capas

La información atraviesa dos capas entre la interfaz y la administración de los datos.

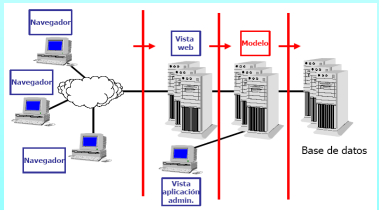
3 Capas

Introduce una capa intermedia (la capa de proceso) entre presentación y los datos.



N Capas

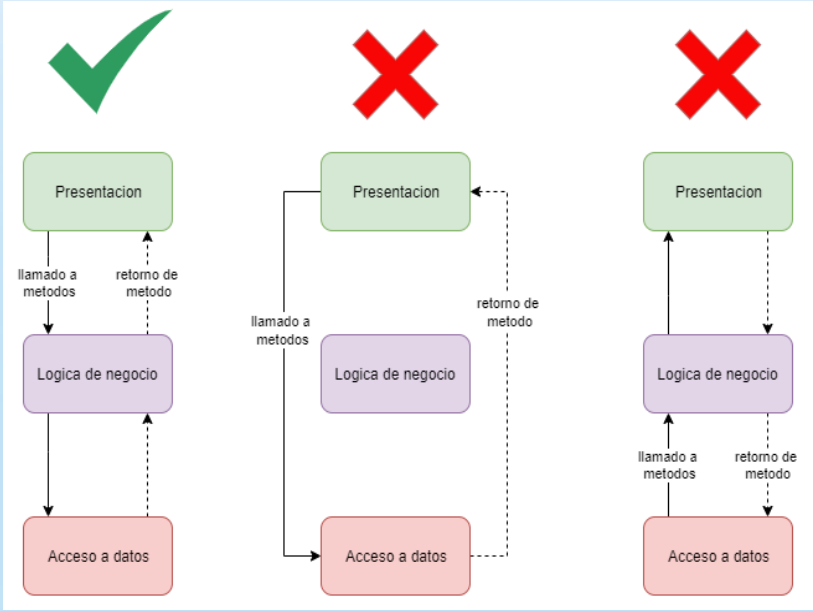
A emergido como la principal arquitectura para la construcción de aplicaciones multiplataforma.



Capas

La arquitectura establece reglas de cómo se deben comunicar las capas.

- Cada capa debe tener una responsabilidad única. Es decir que las capas deben estar perfectamente delimitadas de que se ocupa cada una de ellas.
- Las capas deben respetar una estructura jerárquica estricta. Quiere decir que cada capa pude comunicarse solo con la que está debajo suyo, pero NO al revés.



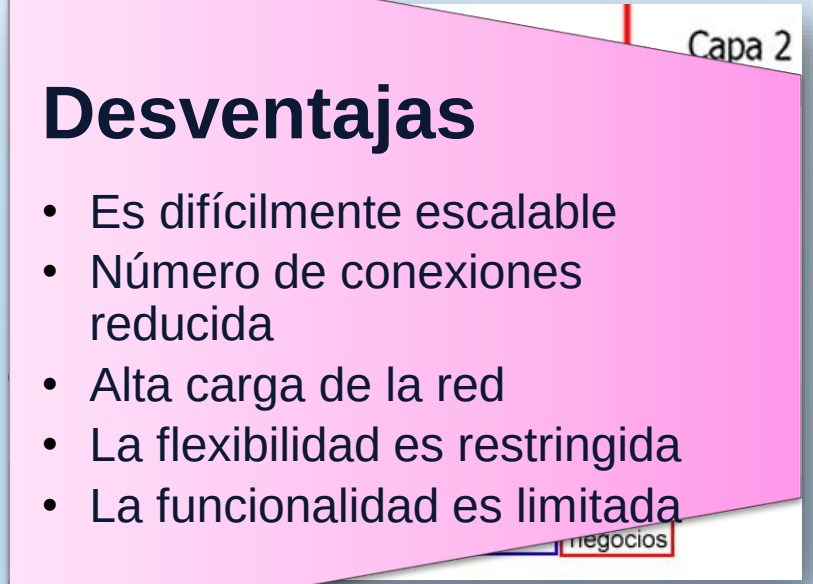
2 Capas

Se encarga de manejar la interacción entre un cliente y la aplicación.

- **Capa aplicación:** en este nivel se encuentra toda la interfaz del sistema y permite al usuario realizar su actividad con el sistema.
- **Capa de base de datos:** este nivel de la base de datos, es la capa en donde se almacena toda la información ingresada en el sistema y que se deposita en forma permanente.

Desventajas

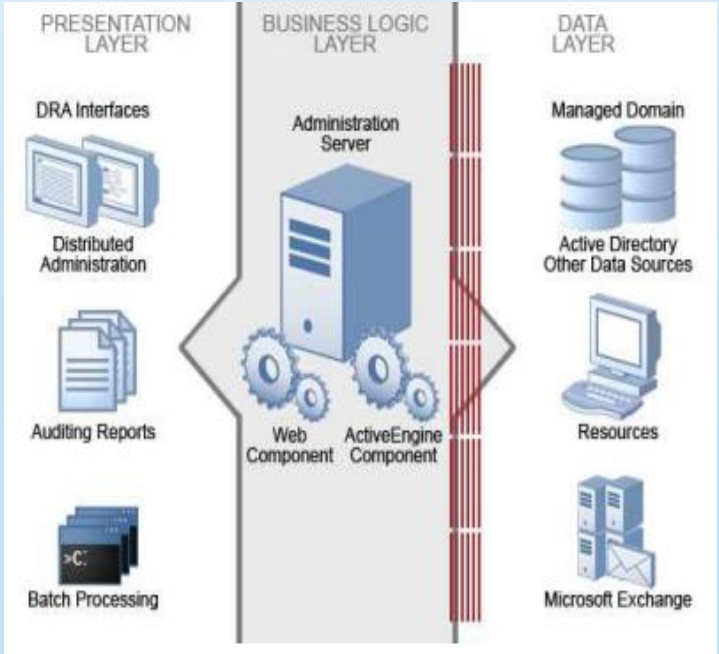
- Es difícilmente escalable
- Número de conexiones reducida
- Alta carga de la red
- La flexibilidad es restringida
- La funcionalidad es limitada



3 Capas

Permite separar la presentación de la lógica del negocio y esta de la de datos.

- **Presentación:** la presentación del programa ante el usuario, debe manejar interfaces que cumplan con el objetivo principal de este componente, el cual es facilitar al usuario la interacción con la aplicación.
- **Lógica de Negocio:** es aquí donde se encuentra toda la lógica del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos existentes. También se procesa la información ingresada o solicitada por el usuario en la capa de Presentación.
- **Datos:** capa en donde se almacenan los datos y toda la información



Ventajas – 3 capas

Ventajas

- Se reduce la complejidad
- Facilidad para distribuir el desarrollo del software
- Mayor encapsulamiento
- Alta escalabilidad
- Facilidad para desarrollar en múltiples plataformas (web, escritorio, móvil)

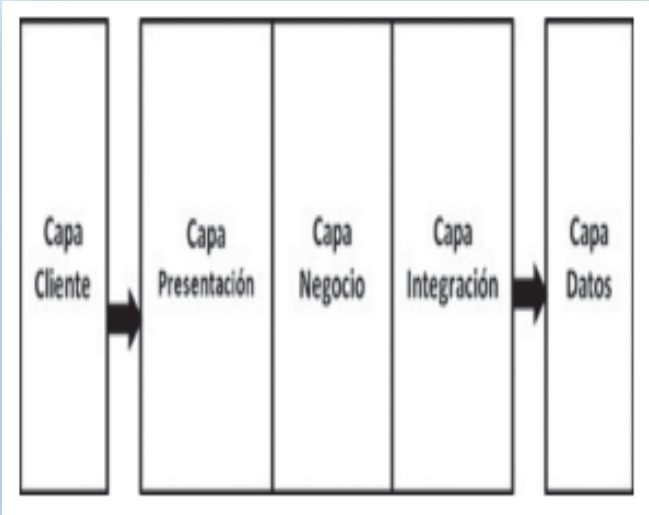
Desventajas

- Debe existir una balance entre las capas y subcapas del sistema, necesario para conservar el grado de sencillez para una posible migración.
- De existir demasiadas capas, es posible que exista perdida de eficiencia, trabajo redundante, gasto de espacio de la aplicación en disco.

N capas

La arquitectura multicapas introduce muchas mejoras importantes dentro del diseño de la aplicación, incluyendo la flexibilidad a través de una adecuada separación entre la capa de presentación y la lógica de negocio.

- La presentación puede cambiar sin impactar la lógica existente. Además, la misma lógica de negocio puede ser reutilizada en diferentes tipos de presentación.
- Con la introducción de una capa de negocio separada, se simplifica la integración de múltiples fuentes de datos dentro de una aplicación.
- Permite configurar las opciones de despliegue.



N capas

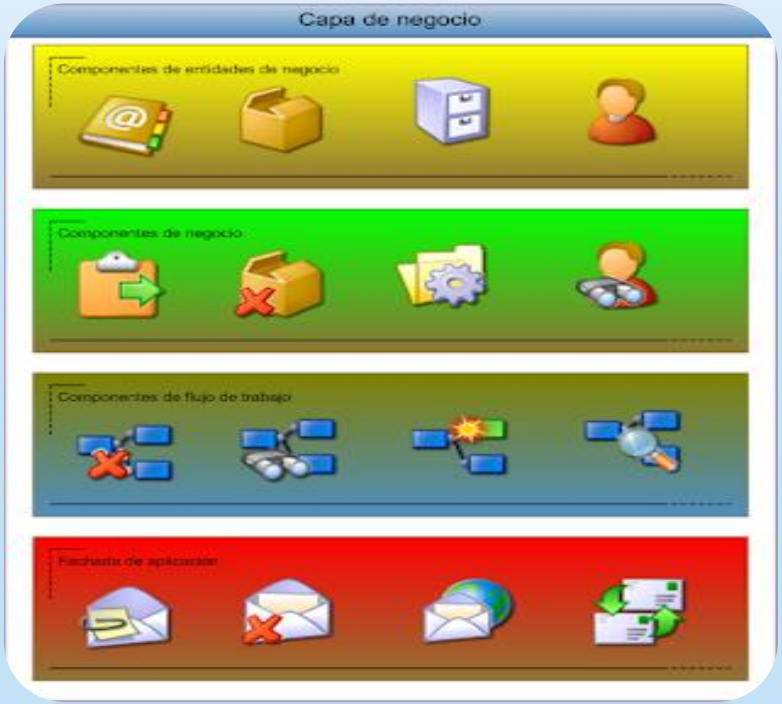
La arquitectura n-capas es una generalización de la arquitectura 3-capas, en la que se incluye una capa adicional, responsable de la interacción con el usuario, y capas adicionales para manejar la conexión a la fuente de datos.

- **Capa Cliente.-** Es la responsable de la interacción entre el sistema y el usuario a través de una presentación específica. La separación con la capa de negocio permite manejar diferentes tipos de interfaz de usuario, como un browser web o un dispositivo móvil.

N capas

Capa de presentación.- Permite al sistema soportar múltiples interacciones con un único usuario. Coordina y mantiene una sesión de usuario, manipulando los datos asociados con la sesión y con la interacción con la capa de negocio; coordina y mantiene la integridad con múltiples actividades para el mismo usuario; ejecuta procesos que no requieren acceso a los recursos empresariales. Es de aclarar que no hay una instancia de los recursos de negocio para cada usuario, sino que hay sólo una instancia de la capa de negocio y de datos que es compartida por todos los usuarios del sistema.

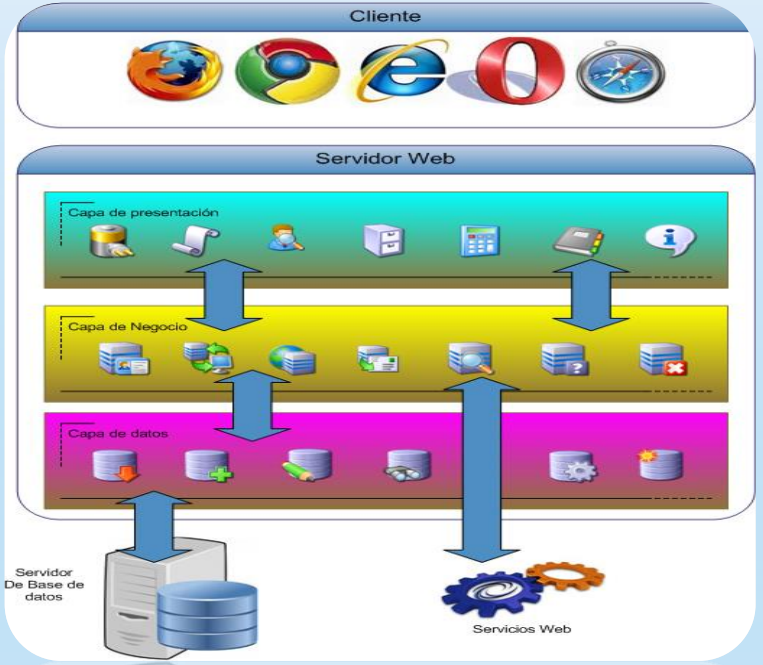
N capas



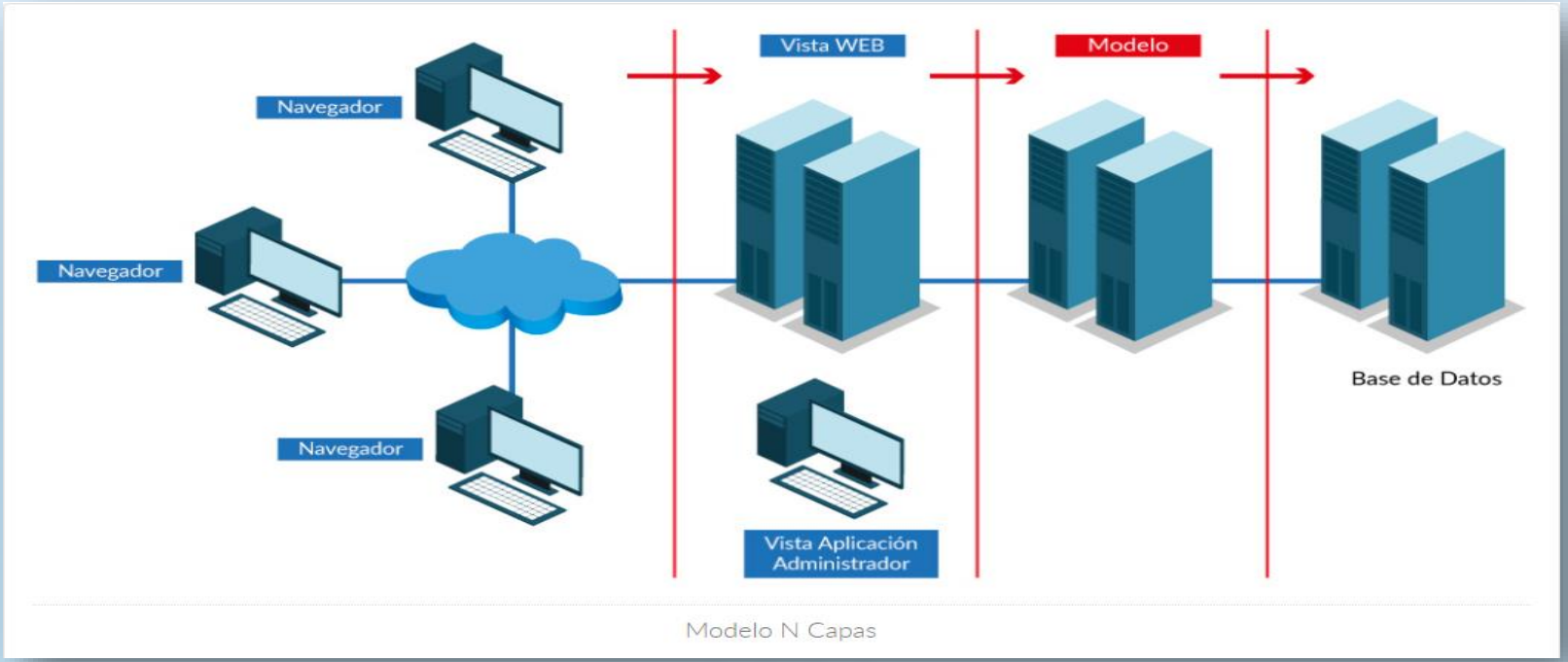
Capa de negocio.- Es responsable de implementar los procesos y las entidades de negocio, y de hacer sus funciones disponibles a través de interfaces. Mantiene la integridad y fuerza el cumplimiento de reglas de negocio; maneja el control de transacciones. Provee un único punto de acceso a los recursos empresariales de manera que estos pueden ser compartidos y reutilizados por múltiples aplicaciones y usuarios.

N capas

- **Capa de integración.-** Es la responsable del manejo y acceso a los recursos empresariales compartidos. Provee acceso a las fuentes de datos y sistema legados.
- **Capa de datos.-** Fuentes de datos con cualquier modelo (sistemas de bases de datos o archivos de datos) y sistemas legados.

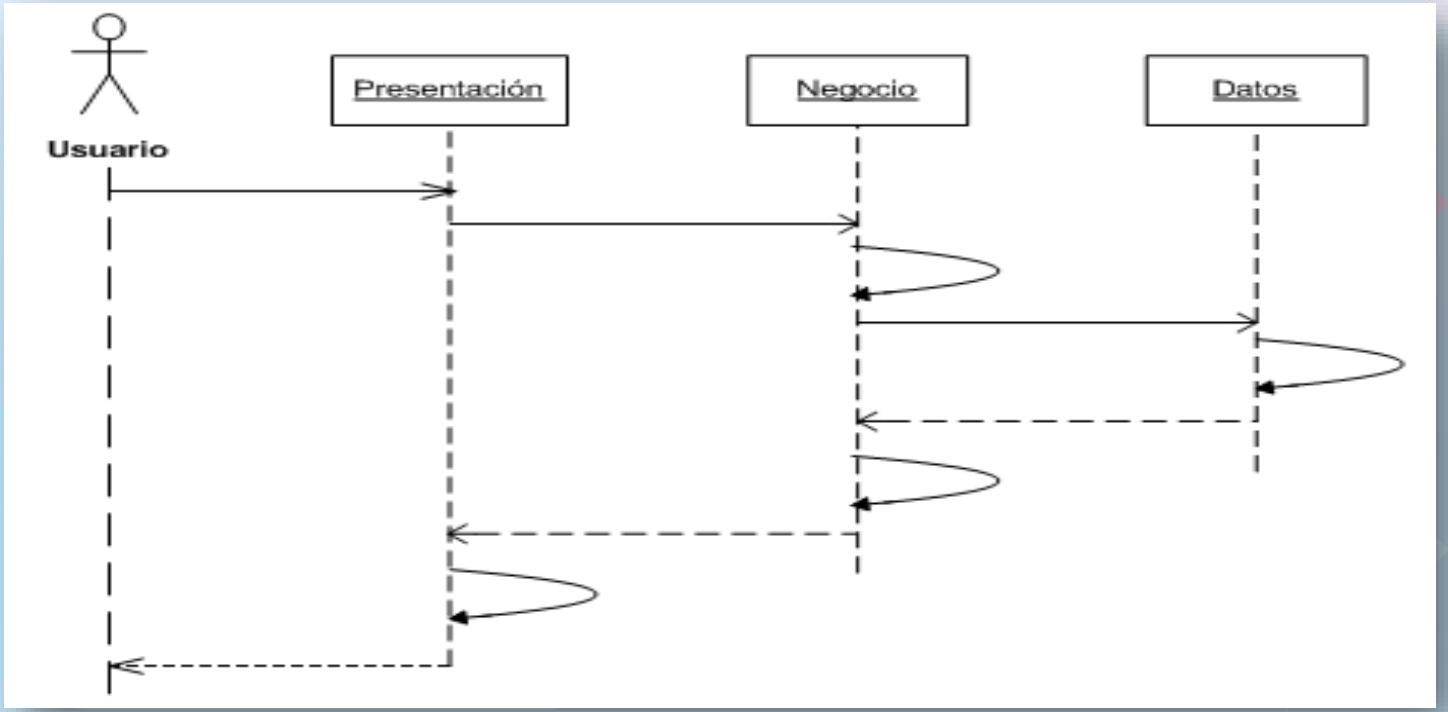


N capas



Modelo N Capas

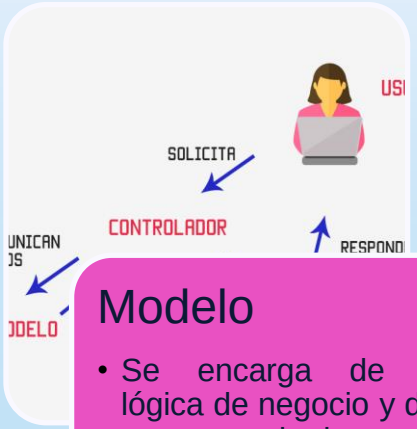
N capas



Patrón MVC

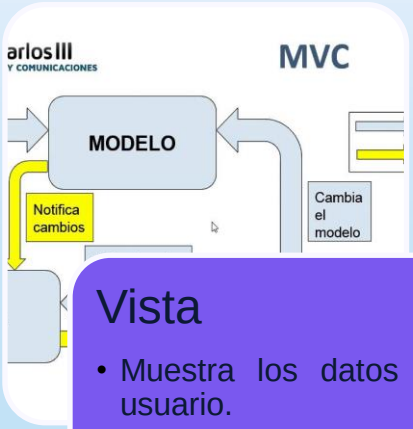
- Es una arquitectura de software que separa el código en tres capas según sus responsabilidades. Surgió para crear software más robusto y mantenible.
- Usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el **desarrollo** esté **estructurado** de una mejor manera, **facilitando** la programación en **diferentes capas** de manera paralela e independiente.

MVC



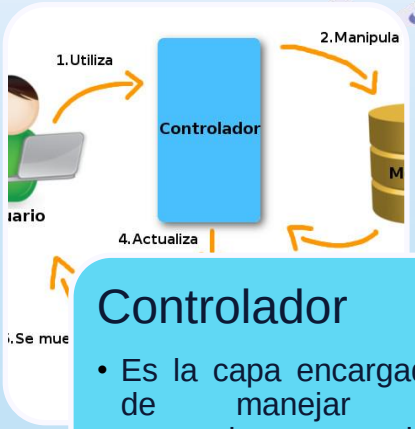
Modelo

- Se encarga de la lógica de negocio y del acceso a la base de datos.
- Representa la información con la que trabaja la app.



Vista

- Muestra los datos al usuario.
- Es la representación del modelo en forma gráfica disponible para la interacción con el usuario.



Controlador

- Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

MVC

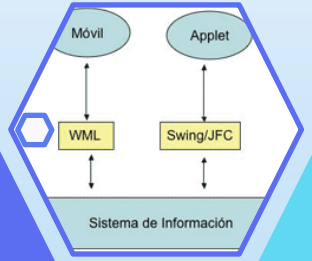
Ventajas

- Separación de responsabilidades
- Mantenibilidad
- Reutilización de código
- Desarrollo en paralelo
- Escalabilidad
- Facilita pruebas

Desventajas

- Curva de aprendizaje
- Mayor complejidad inicial
- Sobrecarga en proyectos simples
- Dependencia de un framework
- Coordinación entre capas

¿Por qué utilizar MVC?



Separación de responsabilidades

Gestión de datos centralizada

Trabajo en equipo

Reutilización de código

Tarea

Investigar niveles en aplicaciones web

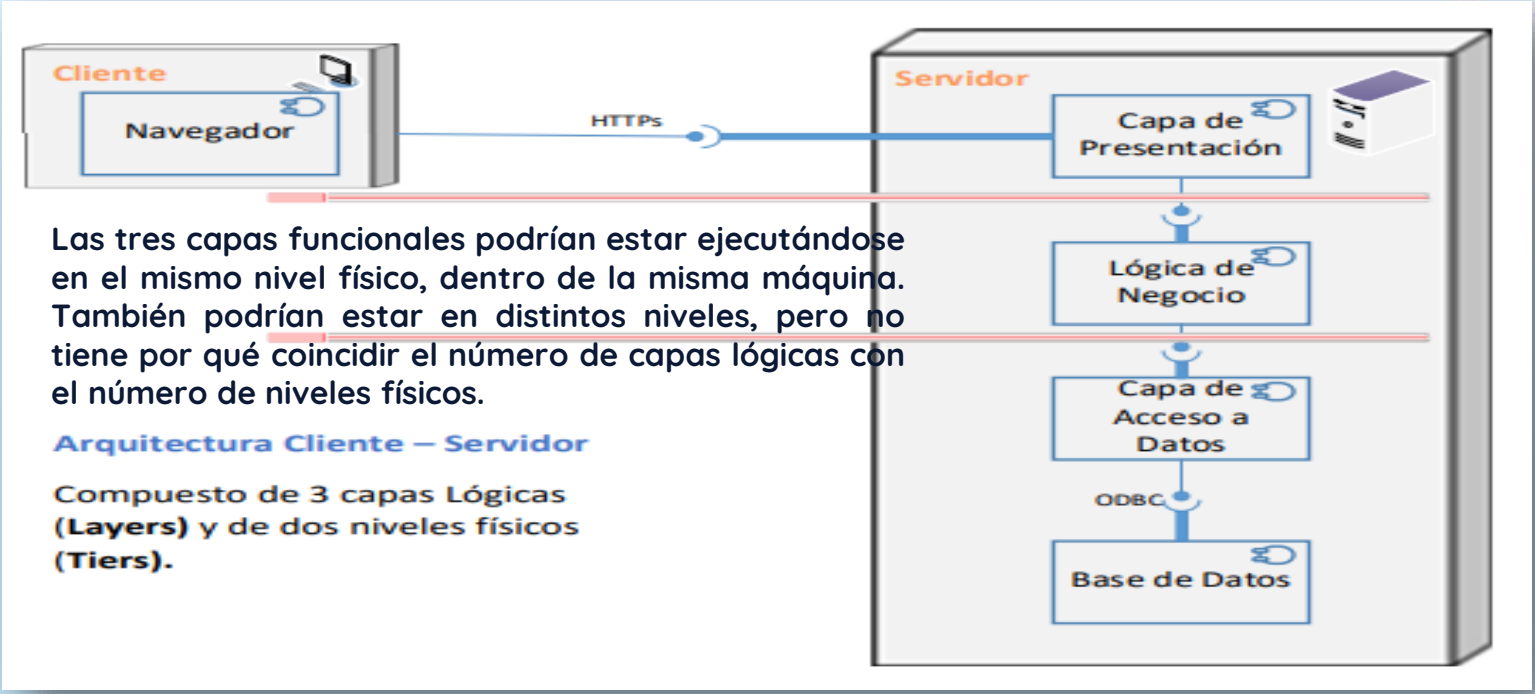
Niveles

Niveles (Tiers) se ocupan de la distribución física de componentes y funcionalidad en servidores separados.

Aunque tanto las Capas (Layers) como los Niveles (Tiers) usan conjuntos similares de nombres (presentación, servicios, negocio y datos), es importante no confundirlos y recordar que solo los Niveles implican una separación física.

Se suele utilizar el termino ”Tier” refiriéndose a patrones de distribución física como “2 Tier”, “3- Tier” y “N - Tier”

2 Niveles



Las tres capas funcionales podrían estar ejecutándose en el mismo nivel físico, dentro de la misma máquina. También podrían estar en distintos niveles, pero no tiene por qué coincidir el número de capas lógicas con el número de niveles físicos.

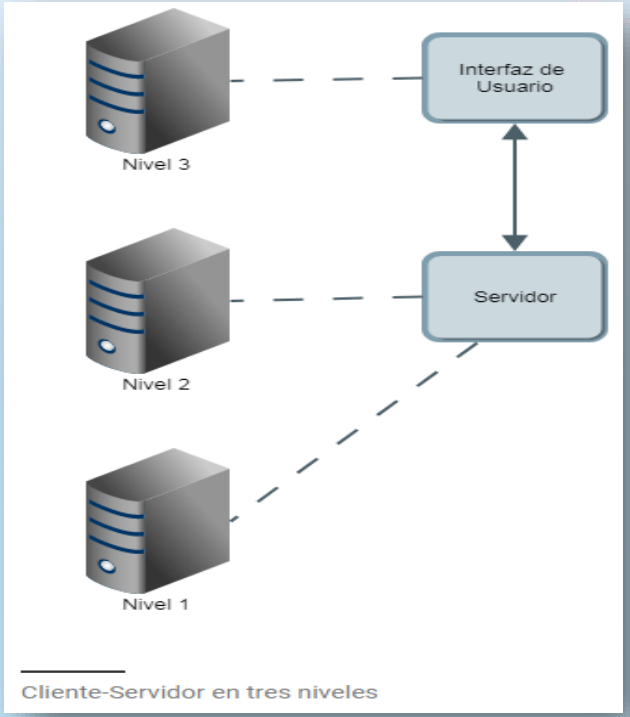
Arquitectura Cliente – Servidor

Compuesto de 3 capas Lógicas **(Layers)** y de dos niveles físicos **(Tiers).**

3 Niveles

Si, entendiendo la diferencia entre capa lógica y nivel físico. Como vemos, no se debe considerar que un SGBD es una capa de Datos en sí. La capa lógica de datos puede contener una lógica de persistencia además de la propia gestión de datos. Y la gestión y almacén de datos puede estar en otro nivel.

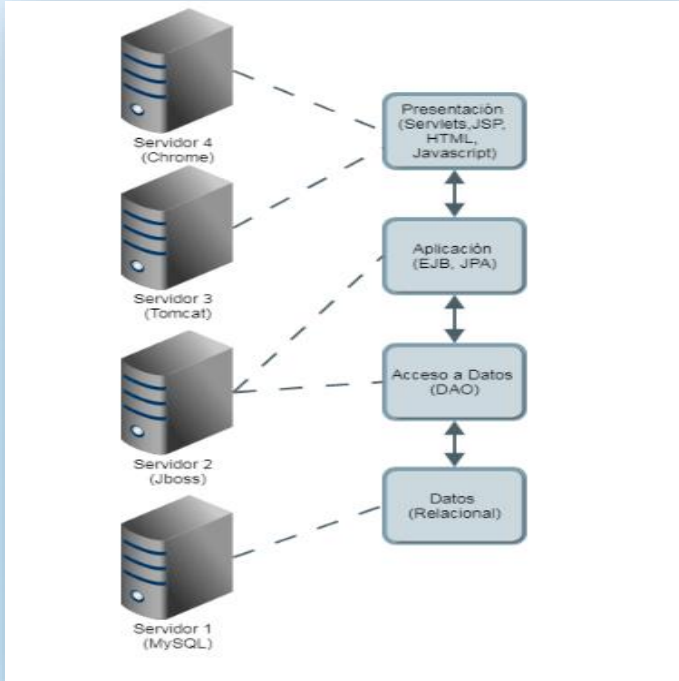
En este caso, el nivel 1 físico tendría la base de datos, y el nivel 2 la lógica y aplicación Servidor. Por ejemplo, un servidor web en Jboss en el nivel 2 y un motor Oracle en el nivel 3, ambos formando parte de la capa lógica de Servidor.



Cliente-Servidor en tres niveles

N niveles

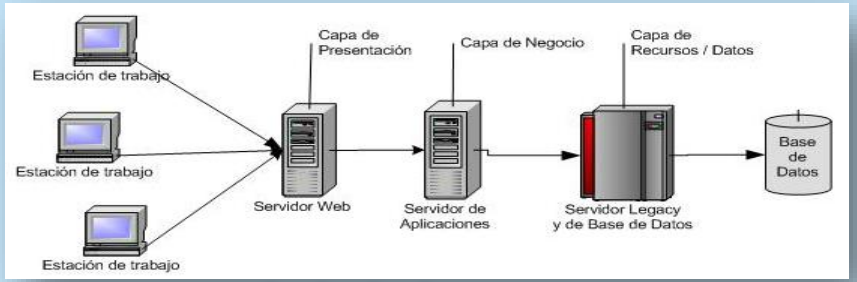
Los diseños de arquitectura multicapa y multinivel aportan gran flexibilidad y una facilidad de mantenimiento, al separar la lógica de cada capa. De esta manera, un cambio en la lógica de una capa no debe afectar al resto de capas. Incluso puede estar separado físicamente, si se encuentran en distintos niveles



N niveles

Los siguientes 3 componentes definen una arquitectura multinivel:

- El componente de **front-end**, responsable de ofrecer portabilidad en la lógica de presentación, como puede ser un servidor web.
- El componente de **back-end**, responsable de ofrecer acceso a servicios dedicados, como un servidor de base de datos.
- Uno o más componentes de **middle-tier**, que permite a los usuarios compartir y controlar la lógica de negocio, aislándola de la aplicación real, como un servidor de aplicaciones.



Ventajas

Los cambios en el interfaz de usuario o en la lógica de la aplicación son independientes unos de otros, lo que permite a la aplicación evolucionar fácilmente para cumplir nuevos requisitos

Se minimizan los cuellos de botellas debido a problemas de red, ya que la capa de la aplicación no transmite información extra al cliente; de hecho, únicamente transmite información que realmente es necesaria para realizar una tarea

Cuando son necesarios cambios en la lógica de negocio, solo debemos actualizar el servidor

Se aísla al cliente de la base de datos y las operaciones de red. El cliente puede acceder a los datos fácilmente sin necesidad de saber donde están los datos o cuantos servidores hay en el sistema

Las conexiones de base de datos se pueden reutilizar (mediante un *pool de conexiones*), de modo que se comparten entre varios usuarios, lo que reduce drásticamente los costes asociados a las licencias por usuarios

Se consigue independencia de la organización respecto a los datos

Desventajas

La arquitectura multicapa aporta mayor complejidad al sistema, y la división en niveles lleva la mayor complejidad al despliegue. Además, hay que observar otros factores de la división en niveles, como la afectación al rendimiento. Al estar separados aumenta la carga de comunicación por red, y esto conlleva latencias en disponibilidad.	Es fácil agrupar las funcionalidades en capas por aquello que tienen en común: • Manejo de datos • Manejo el control • Flujo de la aplicación • Manejo del interfaz de usuario o presentación	Sin embargo la división en niveles no es tan fácil, y requiere buscar un equilibrio entre factores como: • Rendimiento • Escalabilidad • Reusabilidad

Tecnologías y estándares web

Un estándar puede definirse como un conjunto de reglas normalizadas que indican los requisitos a cumplir por todo producto, proceso o servicio, con el fin garantizar la compatibilidad entre los distintos elementos que lo utilicen.

World Wide Web Consortium (W3C) desarrolla estándares web o recomendaciones que tienen por finalidad conseguir que las tecnologías que conforman la Web sean interoperables, eficientes, confiables, accesibles y fáciles de usar, lo que a su vez repercutirá en el desarrollo de aplicaciones cada vez más robustas.

Los estándares web han surgido de la necesidad de evitar la fragmentación de la Web así como de mejorar la organización de la información ofrecida en ella, y muchos de ellos han ido sentando las bases de su desarrollo y fomentando su éxito.

Estándares web

Los estándares Web más conocidos y ampliamente utilizados son:

- El lenguaje de etiquetado para hacer páginas Web **HTML** (HyperText Markup Language).
- El lenguaje de hojas de estilos **CSS** (Cascading Style Sheets).
- **Javascript**, que permite otorgar dinamismo y funcionalidad.
- Permiten controlar la presentación de los documentos (X)HTML.

Estándares web

¿Por que es importante cumplir con los estándares?

La finalidad de los estándares es la creación de una web universal, accesible, fácil de usar y en la que todo el mundo pueda confiar.

Los sitios web son construidos a través de un lenguaje HTML y CSS, estas etiquetas sin el cumplimiento de los estándares o la validación correcta pueden presentar diferencias o errores, si se visualizan desde diferentes navegadores web.

Estándares web

Ventajas

- Mantener la web gratis y accesible para todos.
- Ayudar a simplificar el código fuente.
- Reducción del tiempo de desarrollo y mantenimiento.
- Hacer de la web un lugar más accesible.
- Permitir compatibilidad y validación hacia atrás.
- Ayudar a mantener un mejor seo (optimización de motores de búsqueda).
- Crear un grupo de conocimiento común.

Estándares web

Aspectos que verifica W3C

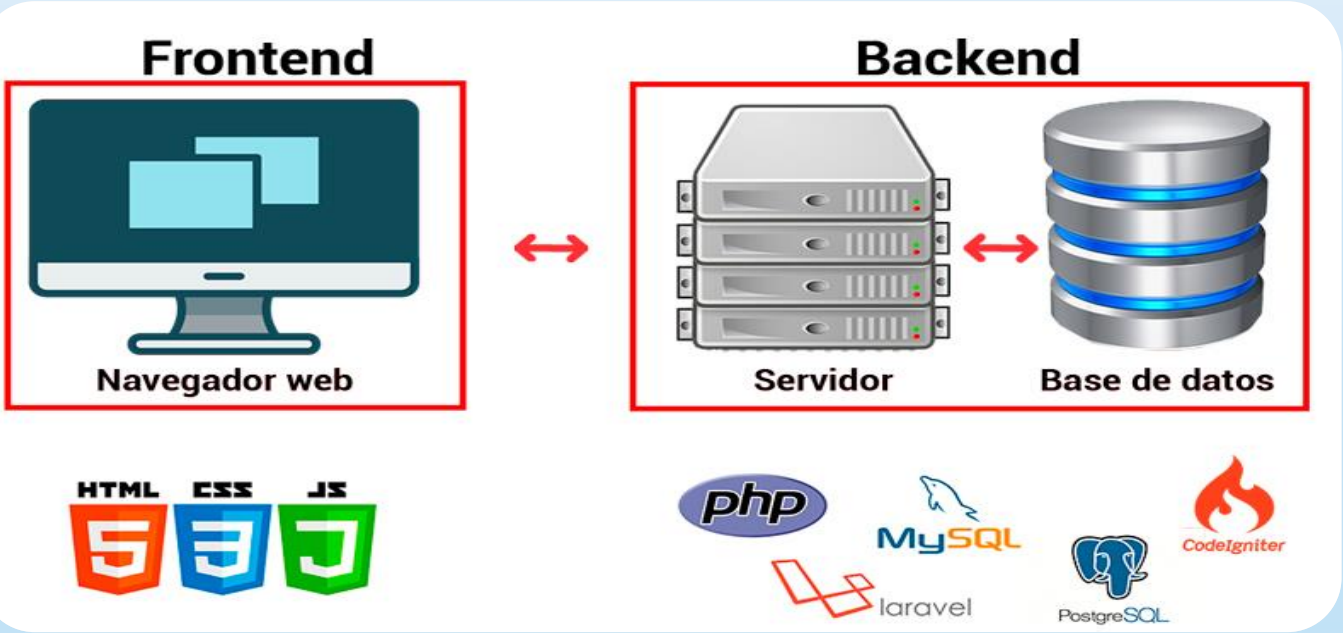
Calidad en el código: usar correctamente las etiquetas, no usar id repetidos, evitar errores de JavaScript.

Accesibilidad para usuarios: usar atributos **alt** para todas las imágenes, usar enlaces descriptivos, contraste en los colores.

Accesibilidad para dispositivos: funciona en la mayoría de navegadores y sus versiones más antiguas, valida si funciona en dispositivos móviles y en diferentes tamaños de pantallas.

Gestión del sitio: no tiene enlaces rotos, usa una pagina 404 personalizad, usa urls amigables.

Estándares web



Tarea

- Realizar una presentación interactiva sobre HTML5