ARQUITECTURAS Y TECNOLOGÍAS DE PROGRAMACIÓN CLIENTE

UT1. JAVASCRIPT

2º DAW IES MARTÍNEZ MONTAÑÉS

## 

ISABEL PASTOR LÓPEZ

Monday, 3 de October de 2022

Contenido

[ARQUITECTURAS Y TECNOLOGÍAS DE PROGRAMACIÓN CLIENTE 3](#_Toc115640483)

[Modelos de programación en entornos cliente/servidor 3](#_Toc115640484)

[Mecanismos de ejecución de código en un navegador Web. Capacidades y 4](#_Toc115640485)

[limitaciones de ejecución. Compatibilidad con navegadores Web 4](#_Toc115640486)

[Lenguajes de programación en entorno cliente. Tecnologías asociadas 4](#_Toc115640487)

[Características de los lenguajes de script. Ventajas y desventajas 4](#_Toc115640488)

[Integración del Javascript con las etiquetas HTML y CSS 4](#_Toc115640489)

[Herramientas de programación para Javascript 4](#_Toc115640490)

ARQUITECTURAS Y TECNOLOGÍAS DE PROGRAMACIÓN CLIENTE

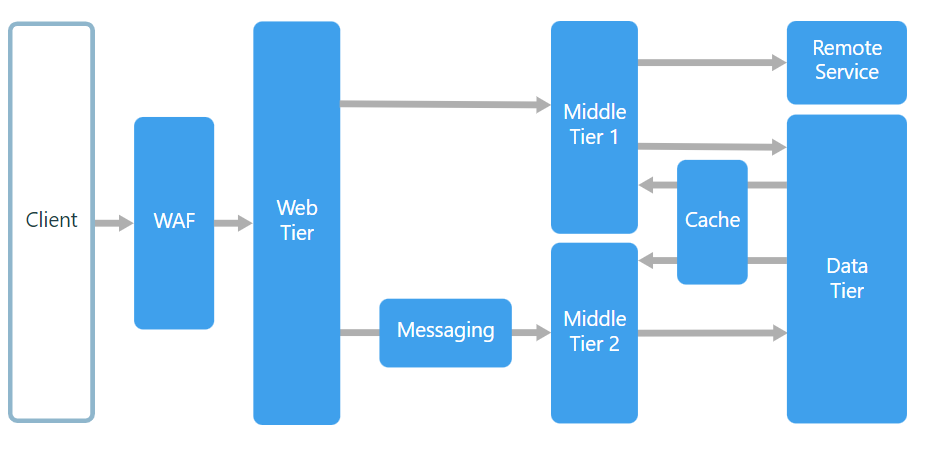
## Modelos de programación en entornos cliente/servidor

La arquitectura cliente/servidor es un modelo en el que se ejecutan tareas separadas de clientes y servidores que se comunican en la misma red, distribuyéndose la carga de trabajo.

El cliente hace peticiones y el servidor se encarga de procesar y mostrar, o no, respuesta a dichas peticiones

En la programación por capas se incluyen todas las configuraciones, la interfaz gráfica, la lógica de negocio, la lógica de los datos… y todas están abstraídas unas de otras.

Los modelos de programación por capas se clasifican por número de capas, de 1 a N capas



Una arquitectura de n niveles divide una aplicación en capas lógicas y niveles físicos. **Fuente:** <https://learn.microsoft.com>

Capa de presentación:

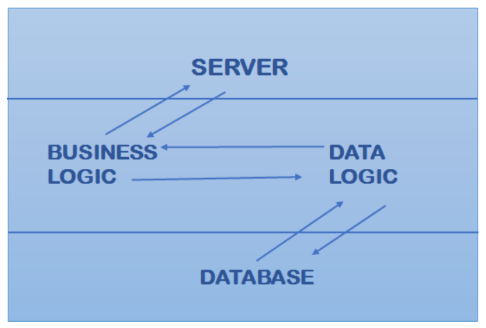
La capa de presentación se ejecuta en el lado cliente, atañe a la interfaz, aquí entra la usabilidad, accesibilidad, adaptabilidad… Esta capa sólo se comunica con la capa de negocio

Capa de negocio:

Son los programas que “hacen de intermediarios”, recogen peticiones del lado del cliente y reciben respuestas del servidor y se ocupan de establecer las normas en dicha comunicación

Capa de datos:

La capa de datos se encarga de, una vez recibidas las peticiones, devolver los datos solicitados

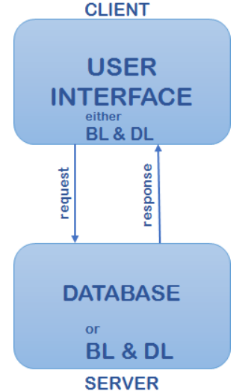
Arquitectura de 1 nivel:

En este tipo de entorno cliente servidor, la interfaz de usuario, la lógica comercial y la lógica de datos están presentes en el mismo sistema. Este tipo de servicio es más barato, pero es difícil de manejar debido a la inconsistencia de los datos que permite la repetición del trabajo

Arquitectura de 2 niveles:

En este tipo de entorno cliente servidor, la capa de presentación se almacena en la máquina del cliente y la capa de datos se almacena en el servidor. La lógica de la base de datos y la lógica de negocio se almacenan en el cliente o en el servidor.

Si la lógica de negocios y la lógica de datos se pueden almacenar tanto en el lado del cliente, como en el lado del servidor.

La arquitectura de 2 niveles es útil cuando un cliente se comunica directamente con un servidor. No hay servidor intermedio. Suele utilizarse en entornos pequeños. Aquí, la interfaz de usuario se ubica en el entorno de escritorio del usuario, mientras que los servicios del sistema gestor de base de datos generalmente se ubican en un servidor. El procesamiento de la información se divide entre el entorno de la interfaz del sistema del usuario y el entorno del servidor de administración de la base de datos

Arquitectura de 3 niveles:

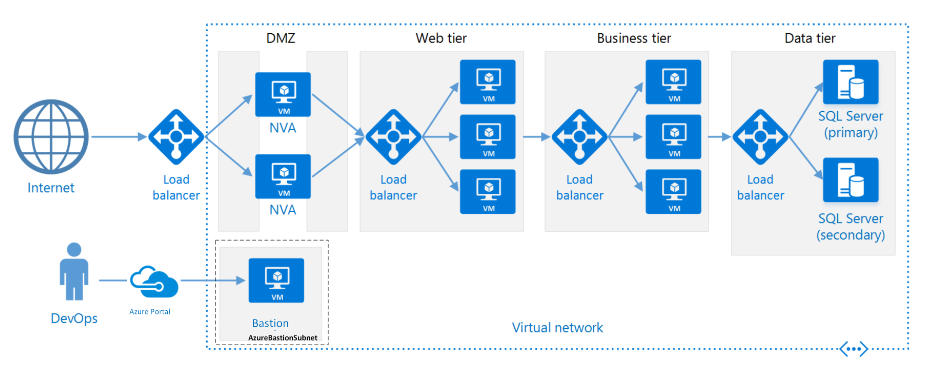
Este tipo de arquitectura implica una capa intermedia (middleware) que recibe las peticiones del cliente y procesa la respuesta del servidor. O sea, se ocupa de la capa de lógica de negocio y de la capa de datos. Su coste es mayor, es más fácil de manejar. La finalidad del middleware es la preparación de bases de datos, la puesta en cola, la ejecución de aplicaciones, la programación, etc. Ejemplos de middleware pueden ser: un servidor de archivos, un servidor de mensajes, un servidor de aplicaciones, etc.



Fuente: [TechBlog](https://msatechnosoft.in/blog/types-of-client-server-architecture/)

Arquitectura de n niveles:

Este tipo de arquitectura implica el uso de máquinas virtuales y se usan para una alta disponibilidad y mayor seguridad, cada capa dispone de al menos dos máquinas virtuales



**Fuente:** <https://learn.microsoft.com>

## Mecanismos de ejecución de código en un navegador Web. Capacidades y limitaciones de ejecución. Compatibilidad con navegadores Web

Cuando el código se ejecuta en el entorno cliente (navegador web) como por ejemplo con JavaScript, puede modificar el contenido de la página, así como su aspecto, puede crear animaciones con texto, imágenes, desplegables, puede revisar formularios antes de ser enviados, etc.

Hoy en día casi todos los lenguajes de programación que se ejecutan del lado cliente, son compatibles con la mayoría de navegadores, con muy escasas excepciones.

En [MDN](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/String#compatibilidad_del_navegador) podemos comprobar la compatibilidad con el navegador en cada una de las funcionalidades:



También la web [Can I use](https://caniuse.com/ciu/comparison) nos facilita esta tarea para programar en JS y otros lenguajes de marcas y estilos, previniendo a programadores de las incompatibilidades

## Lenguajes de programación en entorno cliente. Tecnologías asociadas

PISTA para tecnologías: Frameworks Javascript

## Características de los lenguajes de script. Ventajas y desventajas

HRDRH

## Integración del Javascript con las etiquetas HTML y CSS

DDFG

## Herramientas de programación para Javascript