



U.T.1

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

DESARROLLO WEB EN ENTORNO CLIENTE

CONTENIDO

- 01** DESARROLLO DE SOFTWARE
- 02** MODELO CLIENTE – SERVIDOR
- 03** ARQUITECTURA DE UNA APLICACIÓN WEB
- 04** TECNOLOGÍAS DEL LADO CLIENTE
- 05** HERRAMIENTAS PARA EL DESARROLLO EN EL LADO CLIENTE
- 06** INTEGRACIÓN DEL CÓDIGO CON LAS ETIQUETAS HTML



01

DESARROLLO DE SOFTWARE

DESARROLLO DE SOFTWARE

Puede definirse el desarrollo de aplicaciones como el proceso de creación de un programa informático (software) para realizar las diferentes tareas que cubren una necesidad



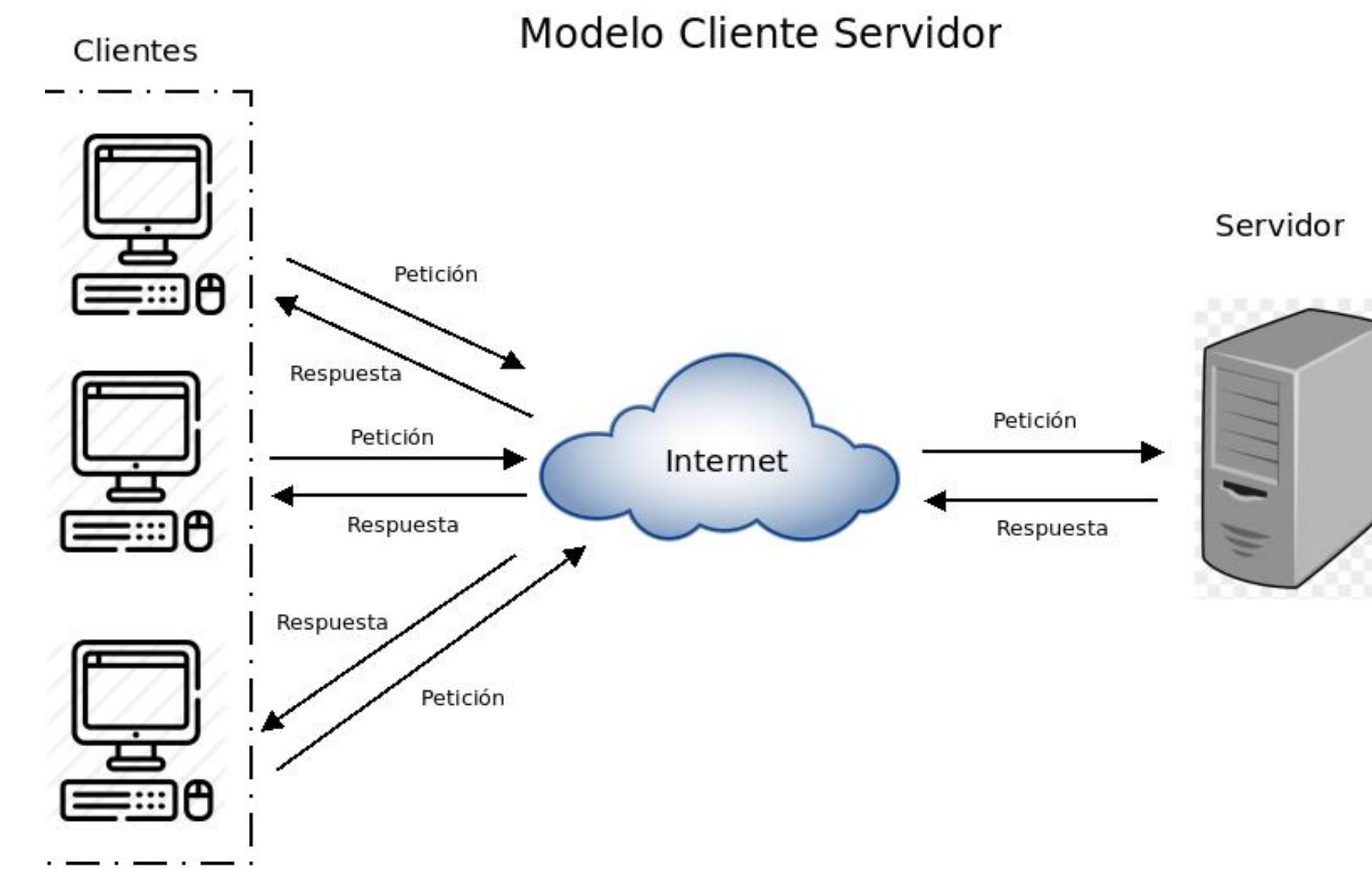


02

**MODO CLIENTE
SERVIDOR**

MODELO 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.



LADO CLIENTE Y LADO SERVIDOR

LADO CLIENTE

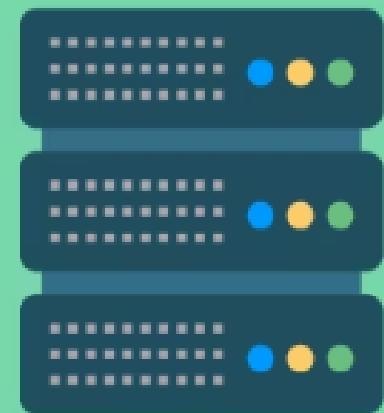


FRONT END

- Se trata de todo aquello que el usuario puede ver, interactuar y experimentar
- El objetivo de estos desarrolladores es programar las partes del sitio web que son visibles para el usuario y que se ejecutan en el navegador
- Desarrollan los diseños de la interfaz de usuario y su experiencia

LADO SERVIDOR

- Es donde está programada la lógica de negocio de la aplicación
- Las acciones realizadas por el usuario son analizadas, recuperadas y devueltas por el back-end a través de los programas almacenados y ejecutados en el servidor.
- El trabajo de estos desarrolladores es vincular el front-end con las bases de datos



BACK END

DIFERENCIAS ENTRE EL DESARROLLO DE LOS LADOS CLIENTE Y SERVIDOR

	Lado cliente	Lado servidor
Definición	Implica la implementación efectiva de los componentes visuales de una aplicación web.	Implica la implementación efectiva de funcionalidades de una aplicación web, donde se incluyen el acceso a datos, la administración de servidores, etcétera.
Habilidades requeridas	HTML, CSS, SASS, JavaScript...	Python, Ruby, Java, PHP...
Independencia	No puede funcionar de forma independiente excepto en el caso de sitios estáticos.	Funciona de forma independiente respecto del <i>front-end</i> .
Objetivo	Garantizar que todos los usuarios puedan acceder a la aplicación y que siga respondiendo en todos los dispositivos.	Garantizar que la aplicación se ejecute en todos los casos, sea escalable y funcione de manera eficiente con baja latencia y sin fallos.
Equipo de desarrollo	Su trabajo es diseñar y desarrollar la apariencia e interactividad de la aplicación en función de la entrada de usuario.	Su trabajo es proporcionar datos al <i>front-end</i> , vincular páginas, brindar seguridad y soporte a los usuarios.
Frameworks utilizados	AngularJS, React, vue.js...	Django, Flask, CakePHP, Laravel, Ruby on Rails...
Habilidades adicionales	Una buena comprensión del diseño de UI y UX.	Razonamiento lógico y resolución de problemas.



03

ARQUITECTURA DE UNA APLICACIÓN WEB

ARQUITECTURA DE UNA APLICACIÓN WEB

Arquitectura de tres niveles o Three-tier architecture

Nivel de presentación

Define la interfaz de usuario y la comunicación con la aplicación.

Servidor Web

Nivel de aplicación

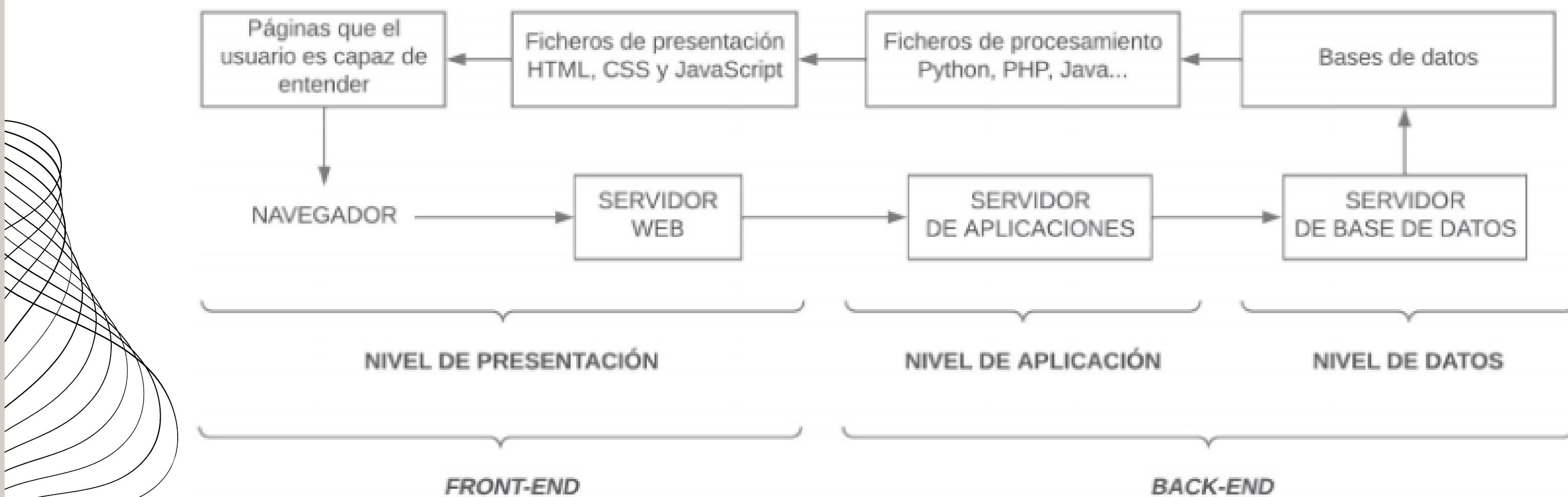
La información recopilada en el nivel de presentación se procesa, a veces con la colaboración del nivel de datos, utilizando la lógica de negocio

Servidor de aplicaciones

Nivel de datos

Se almacena y administra la información procesada por la aplicación

Servidor de Bases de datos





04

**TECNOLOGÍAS DEL
LADO CLIENTE**

TECNOLOGÍAS DEL LADO CLIENTE



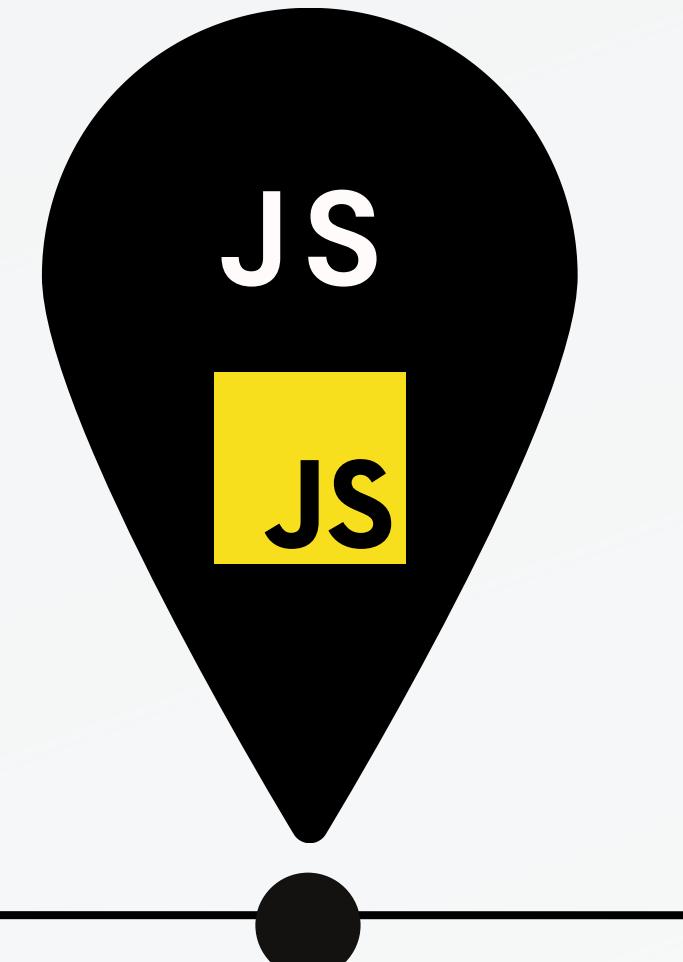
HyperText Markup Language

El Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos.



Cascading Style Sheets

CSS se utiliza para ordenar las instrucciones referentes a la apariencia de un sitio y presentar los contenidos de una página de forma atractiva.



Javascript

JavaScript se utiliza para hacer páginas web interactivas. Proporciona flexibilidad y capacidad de respuesta rápida al sitio web.



JAVASCRIPT

- Es muy fácil de implementar: tan solo es necesario colocar el código en un documento HTML y decirle al navegador que es JavaScript.
- Funciona en todos los navegadores que usan los usuarios de la web, incluso cuando están desconectados.
- Permite crear interfaces altamente amigables que mejoran la experiencia del usuario y brindan mucho dinamismo, sin tener que esperar a que el servidor reaccione y muestre otra página.
- Puede cargar contenido de forma asíncrona en el documento si el usuario lo necesita y cuando lo necesite, sin recargar toda la página.
- Puede comprobar lo que es posible hacer en un navegador y reaccionar en consecuencia.
- Puede ayudar a solucionar problemas de incompatibilidades entre navegadores y corregir problemas de diseño con CSS.

95%

De todas las aplicaciones web se crean con Javascript

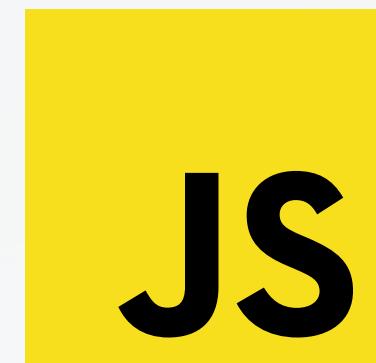




05

HERRAMIENTAS PARA EL DESARROLLO EN EL LADO CLIENTE

HERRAMIENTAS PARA EL DESARROLLO EN EL LADO CLIENTE



NAVEGADOR

EDITOR DE
CÓDIGO

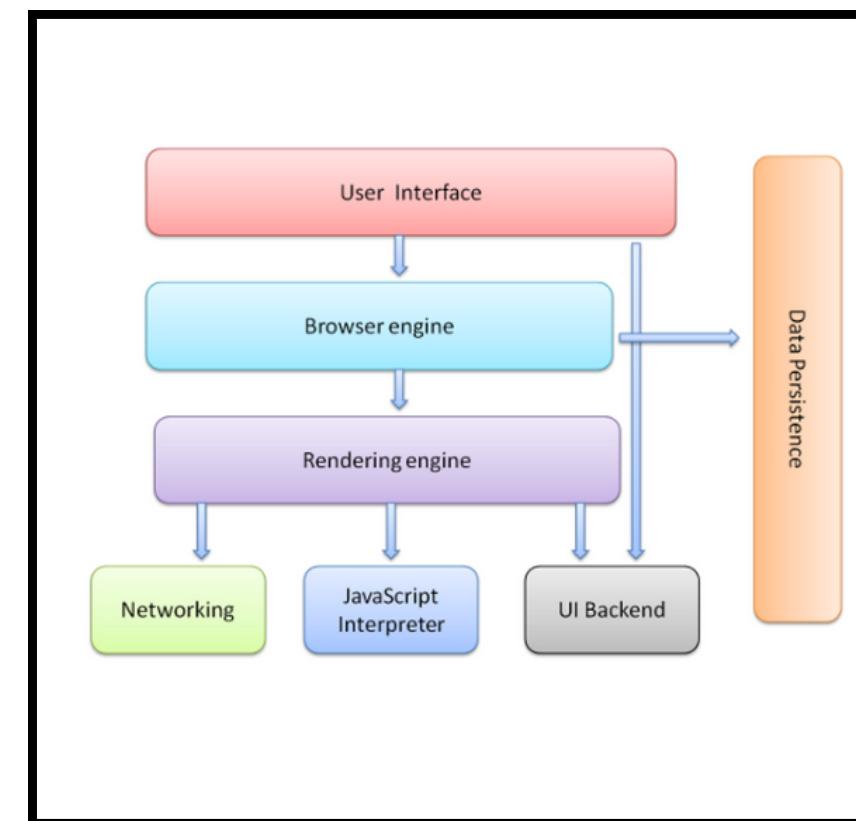
INTÉPRETE DE
JAVASCRIPT

NAVEGADOR

- La función principal de un navegador es solicitar al servidor los recursos web que elija el usuario y mostrarlos en una ventana. El recurso suele ser un documento HTML, pero también puede ser un archivo PDF, una imagen, otro objeto...
- La forma en la que el navegador interpreta y muestra los archivos HTML se determina en las especificaciones de CSS y HTML. Estas especificaciones las establece el consorcio W3C (World Wide Web Consortium).

COMPONENTES PRINCIPALES DEL NAVEGADOR

1. Interfaz de usuario: incluye la barra de direcciones, el botón de avance/retroceso, el menú de marcadores, etc. (en general, todas las partes visibles del navegador, excepto la ventana principal donde se muestra la página solicitada).
2. Motor de búsqueda: coordina las acciones entre la interfaz y el motor de renderización.
3. Motor de renderización: es responsable de mostrar el contenido solicitado. Por ejemplo, si el contenido solicitado es HTML, será el responsable de analizar el código HTML y CSS y de mostrar el contenido analizado en la pantalla.
4. Red: es responsable de las llamadas de red, como las solicitudes HTTP. Tiene una interfaz independiente de la plataforma y realiza implementaciones en segundo plano para cada plataforma.
5. Intérprete de JavaScript: permite analizar y ejecutar el código JavaScript.
6. Servidor de la interfaz: permite presentar widgets básicos, como ventanas y cuadros combinados.
7. Almacenamiento de datos: es una capa de persistencia. El navegador necesita guardar todo tipo de datos en el disco duro (por ejemplo, las cookies).



EDITOR DE CÓDIGO

- La función principal de un editor de código es editar código fuente en diversos lenguajes de programación y ofrecer múltiples herramientas para facilitar el trabajo y aumentar la productividad.

CARACTERÍSTICAS

- **Comprobación de errores** conforme se va escribiendo, lo que proporciona un primer nivel de corrección.
- **Sugerencias de autocompletado**, que agiliza la escritura y acorta los tiempos de desarrollo.
- **Fragmentos de código** previamente configurados con aquellas porciones utilizadas con mucha frecuencia y que evitan reescribirlos constantemente.
- **Resaltado de sintaxis**, que ayuda mucho a identificar cada uno de los elementos de los programas con un simple golpe de vista.
- **Navegación entre archivos y recursos**, que ayuda a tener localizados todos los ficheros en sus ubicaciones correctas.
- **Funcionalidad extendida** a través de complementos específicos de las tecnologías con las que se trabaja en un proyecto en concreto.

EDITOR DE CÓDIGO - IDE

- Un IDE o Entorno de Desarrollo Integrado es una herramienta robusta para escribir, editar, depurar y ejecutar código.
- Un editor de texto solo permite escribir y editar código.

EDITOR VS IDE

Con ambos puedes escribir código, pero ¿en qué se diferencian?

Editor de Código	IDE (Integrated Development Environment)
Software ligero con ayudas para escribir código (resaltado de sintaxis, autocompletado, etc).	Integra un editor con las herramientas que necesita un desarrollador (debugger, compilador, etc).
Soporta múltiples lenguajes y tecnologías.	Se especializa en un lenguaje o tecnología (Java, Python, Go, Android, etc).
Enfocado en archivos (no tienen el concepto de proyecto).	Enfocado en proyectos completos . Desde la primera línea hasta la salida a producción.
Puedes agregar plugins para darle el poder de un IDE pero te toca configurar cada uno a mano.	Trae herramientas integradas y configuradas (ej. Android Studio trae un emulador de Android).

EJEMPLOS DE EDITORES



EJEMPLOS DE IDES



- Visual Studio Code
- Sublime Text
- Atom

- Visual Studio
- Android Studio
- IntelliJ IDEA

INTÉPRETE JAVASCRIPT

- Un intérprete de JavaScript (también conocido como motor de JavaScript o implementación de JavaScript) es un intérprete informático que interpreta el código fuente en lenguaje JavaScript y ejecuta un script acorde a las instrucciones dadas.
- El intérprete de Javascript es un programa informático que es capaz de analizar y procesar el código fuente en tiempo de ejecución, eliminando por completo el proceso de compilación como en otros lenguajes.
- El intérprete de Javascript procesa el código línea por línea, de modo que lee, analiza y prepara cada secuencia de forma consecutiva para el procesador.

HABILITAR EL INTÉPRETE JAVASCRIPT

- **Chrome:** entrar en la **Configuración**, seleccionar **Seguridad y privacidad**, clicar en **Configuración de sitios** y después en **JavaScript** y finalmente seleccionar **Los sitios pueden usar JavaScript**.
- **Safari:** acceder a **Herramientas**, clicar en **Preferencias**, acceder a **Seguridad** y activar la casilla etiquetada como **Activar JavaScript**.
- **Edge:** acceder al menú de **Opciones** y desplegar **Mostrar opciones avanzadas**. En la sección de **Privacidad** clicar en **Configuración de contenido**. Navegar hasta la sección **JavaScript** y finalmente clicar en **Permitir que todos los sitios ejecuten JavaScript (recomendado)**.
- **Firefox:** hacer clic en **Herramientas**, luego en **Opciones** y en la pestaña **Contenido** hacer clic en la casilla **Activar JavaScript**.



06

INTEGRACIÓN DEL CÓDIGO CON LAS ETIQUETAS HTML

INTEGRACIÓN DEL CÓDIGO CON LAS ETIQUETAS HTML

1. Incluyendo código Javascript entre etiquetas

```
<body>
|   <script>
|       document.write("Módulo Desarrollo Web en Entorno Cliente");
|   </script>
</body>
```

2. Código Javascript en un fichero externo

```
document.write("Módulo Desarrollo Web en Entorno Cliente");
```

```
<body>
|   <p>Ejemplo referencia a fichero javascript externo</p>
|   <script type="text/javascript" src="UT2. Ej2 - Código JS en fichero externo.js"></script>
</body>
```

Atributo	Descripción
src	Especifica la URL del fichero que se desea cargar.
type	Indica el tipo de fichero que se está utilizando. Si se omite, se entiende que se trata de código JavaScript



También se puede incluir la etiqueta script dentro del elemento HEAD, pero es mejor incluir código Javascript una vez que sabemos que la página se ha cargado.

3. Incluyendo código Javascript dentro de elementos html

```
<body>
|   <p onclick="alert('Desarrollo Web en Entorno Cliente');">Párrafo con evento asociado</p>
</body>
```