

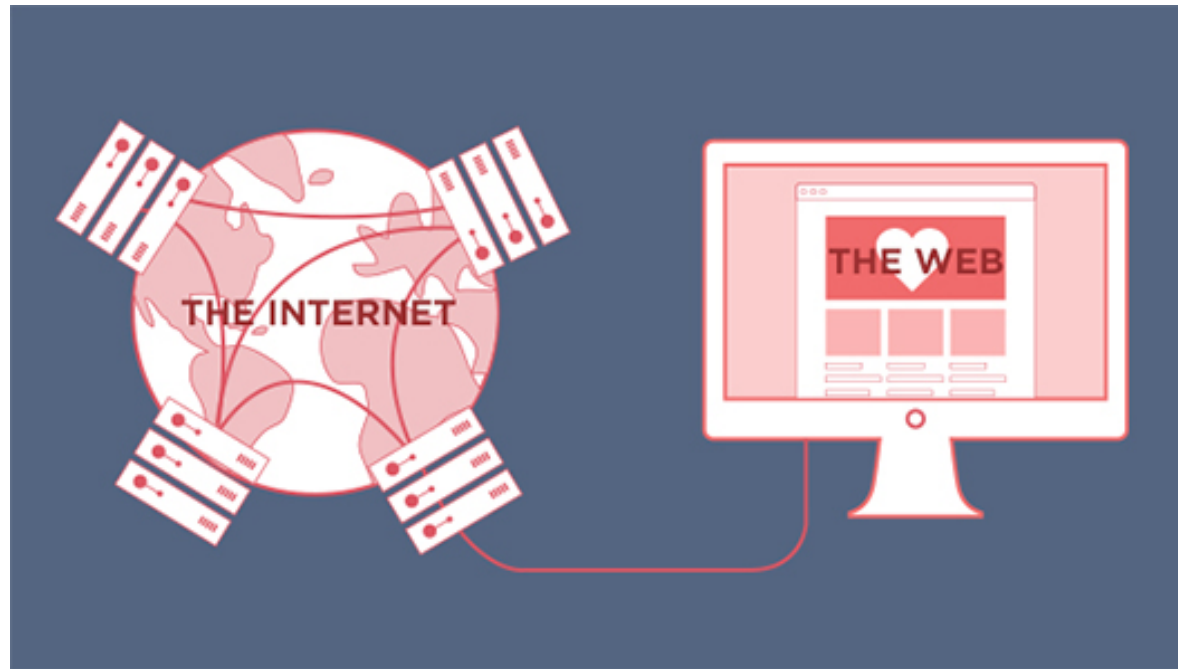
Front-end

UT1. HERRAMIENTAS DE PROGRAMACIÓN WEB FRONTEND

Desarrollo Web en Entorno
Cliente
2ºDAW

1. INTRODUCCIÓN

¿¿ Internet === WWW ??



1. INTRODUCCIÓN

Internet:

- El conjunto de ordenadores interconectados a lo largo de todo el planeta a través de una red global denominada la “red de redes”.
- Tiene sus inicios en Arpanet en el año 1969.

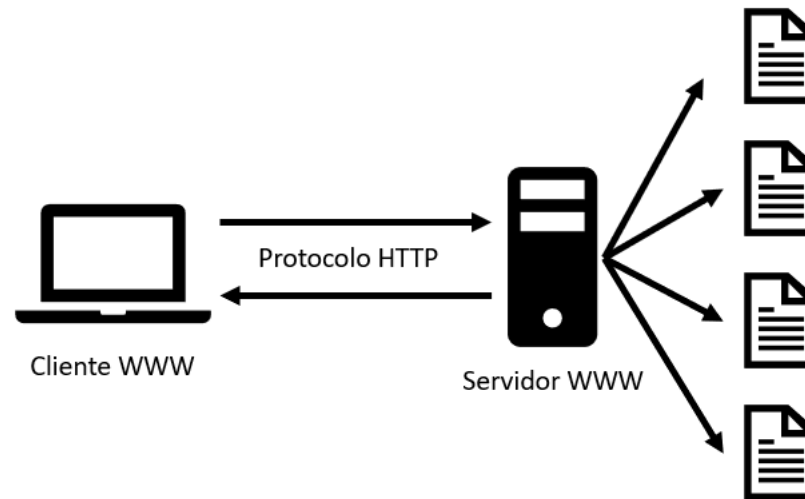
WWW:

- Abreviatura de **World Wide Web** o simplemente “**La Web**”.
- Es el conjunto de todas las páginas de hipertexto interrelacionadas y conectadas.
- Es el servicio más popular de Internet. Se encarga de ofrecer páginas de hipertexto HTML mediante el protocolo HTTP (o su versión segura HTTPS).
- Tiene sus inicios en 1991, año en que aparece HTML y HTTP.

2. ARQUITECTURA WEB

Arquitectura más habitual cliente/servidor.

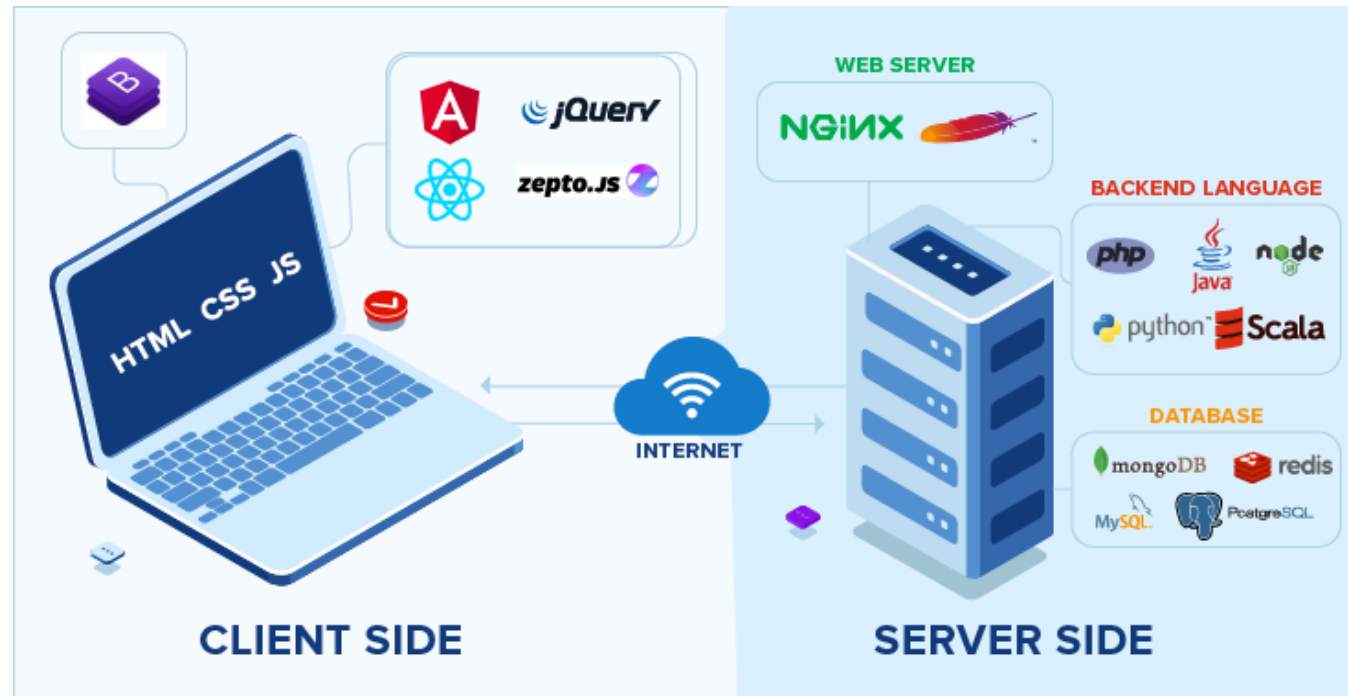
- **Cliente** es un consumidor de servicios. El navegador web de nuestro dispositivo hace las veces de cliente.
- **Servidor** es uno o varios procesos alojados en máquinas que proveen dichos servicios.



3. DESARROLLO FRONTEND Y BACKEND

Backend: Parte de la aplicación web que se ejecuta en el servidor.

Frontend: Parte de la aplicación web que se ejecuta en el cliente.



3.1 DESARROLLO BACKEND

Es el desarrollo que se lleva a cabo en la parte del servidor.

Se encarga de la lógica de negocio y la persistencia de datos (almacenamiento en la base de datos)

Lenguajes de programación:

- Java
- Python
- Node.js
- Ruby
- PHP
- ASP.NET



3.2 DESARROLLO FRONTEND

Es la parte desarrollada para ejecutarse en el cliente (navegador web)

El navegador Web solo sabe interpretar tres lenguajes

- **HTML:** Para la estructura y contenido de la página
- **CSS:** Define el aspecto de la página web.
- **Javascript:** Lenguaje que define el comportamiento dinámico.



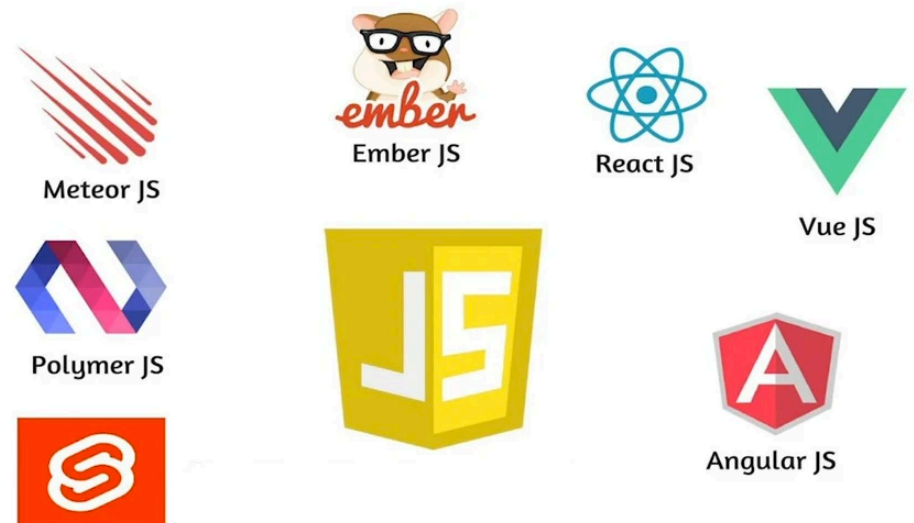
3.2 DESARROLLO FRONTEND

Actualmente está tomando mucho protagonismo por el mayor peso de las aplicaciones web en la parte del cliente → Web Apps SPA

Podemos escribir la lógica de nuestra página en **Vanilla JS**.

Pero La complejidad de los proyectos ha hecho necesaria la aparición de diversos **frameworks** y **librerías** de JS:

- Angular
- React
- Vue
- ...



4. EL LENGUAJE JAVASCRIPT



JS es un lenguaje interpretado

Se define como

- Orientado a objetos
- Imperativo
- Basado en prototipos
- Débilmente tipado
- Dinámicamente tipado

Se usa principalmente en el lado del cliente, implementado como parte del navegador web para dar mejoras a la interfaz web y dar comportamiento dinámico.

4.1 VERSIONES DE JS



JS es un dialecto de ECMAScript y esta definido por dicho estándar.

- Creado por Brendan Eich para Netscape en 1995. En principio llamado LiveScript → JavaScript
- En 1997 ECMA crea el primer estándar ECMAScript
- **ECMAScript 5, ES5**, lanzado en 2009 fue una versión que introdujo muchas mejoras y fue un estándar durante muchos años.
- **ECMAScript 6**, También denominada **ES6**, de 2015. introdujo grandes mejoras como el uso de clases y módulos. Es la versión más extendida soportada por todos los navegadores actuales excepto IE.
- A partir de dicha versión se ha ido liberando una versión por año. Por ejemplo la versión de 2020 es ECMAScript 11.
- Podemos comprobar en <https://html5test.com/> qué versiones de ECMAScript soporta nuestro navegador.

5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND



Un buen **editor de texto**.

- Notepad++
- Sublime text
- Atom
- VS Code



Nosotros usaremos VS Code de Microsoft.

- Soporte para depuración
- Control integrado de Git e integración con GitHub
- Resaltado de sintaxis
- “snippets” o trozos de código reutilizables
- Multitud de extensiones instalables.



5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND

Editores de código online. Tienen la ventaja de no tener que instalar y configurar herramientas en tu equipo. Son muy útiles como “sandbox” (entorno controlado).

- <https://stackblitz.com/>
- <https://codesandbox.io/>
- <https://codepen.io/>

Herramientas de control de versiones

- [Git](#) Herramienta opensource instalable en nuestro equipo.
- <https://github.com/> herramienta de control de versiones online.



5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND

AJAX → Java Script asíncrono y XML. Es una técnica para obtener datos de manera asíncrona mediante JS.

XML → Es un tipo de fichero de marcado el cual es ampliamente utilizado para transferir datos desde una API Web.

JSON → Notación de Objeto de Javascript. Tiene la misma utilidad que XML pero un formato de archivo el cual almacena datos con la notación de objetos de JS.



5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND

Gestores de paquetes. Nos sirven para instalar distintas utilidades, características, frameworks...

- [npm](#) (gestor de paquetes de [node.js](#), se instala con dicho entorno)
- [Yarn](#) una alternativa a npm con algunas mejoras.



5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND

Linters y formateadores de código. Mantienen la consistencia y formateo de nuestro código:

- ESLint
- Prettier

Transpiladores de código. Transforma código JS moderno a versiones más antiguas para mejorar compatibilidad.

- Babel

Bundlers: Empaquetan los componentes necesarios para una aplicación web. Por ejemplo crear un solo JS de múltiples.

- Webpack
- Parcel

FUENTE: [Guía definitiva de tooling para JS y Frontend](#)

5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND

Librerías y frameworks: Permiten extender capacidades y facilitar el uso de JS o CSS.

- [Bootstrap](#) → Framework de CSS. Facilita crear elementos de la interfaz web.
- [jQuery](#) → Librería de JS que facilita su uso y mejora compatibilidad.
- [Angular](#) → Framework JS creado por Google que facilita la creación de aplicaciones web SPA y que siguen el MVC (Modelo Vista Controlador). Está muy extendido y tiene una gran comunidad.
- [Vue.js](#) → Framework JS de código abierto también indicado para crear SPA.
- [React](#) → Biblioteca JS de código abierto creada por Facebook. Es más para definir la Vista aunque puede trabajar con extensiones para definir más partes de la arquitectura. También crear apps SPA.



Ejercicio 1.1: Investiga sobre los diferentes frameworks y selecciona uno que utilizarías para hacer una aplicación SPA. Indica los motivos que te han convencido.

5. HERRAMIENTAS PARA EL DESARROLLO FRONTEND

Navegador web. Importante probar nuestros sitios web en varios de ellos.

Nos permitirán depurar nuestros sitios web con sus herramientas para desarrolladores.

Navegadores:

- Chrome
- Firefox
- Opera
- IE/Edge
- Safari
- ...



6. INCLUIR CÓDIGO JAVASCRIPT

Formas de agregar código javascript.

- Mezclado con el HTML dentro de las propias etiquetas. No se recomienda esta opción excepto en contadas ocasiones
- Dentro de una etiqueta `<script type="text/javascript"></script>`
 - Encerrando el código entre las etiquetas
 - Invocando el archivo mediante `src="ruta-archivo"`

NOTA: Se recomienda que los scripts se carguen al final del documento, por lo que se suelen incluir dentro de `<body>` antes del cierre `</body>`

6. INCLUIR CÓDIGO JAVASCRIPT

Podemos agregar código JS directamente dentro de etiquetas HTML

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hola mundo</title>
8  </head>
9  <body>
10     <a href="#" onclick="console.log('Hola Mundo!!')">Enlace</a>
11 </body>
12 </html>
```

6. INCLUIR CÓDIGO JAVASCRIPT

Podemos incluir código encerrado dentro de la etiqueta `<script>`

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hola Mundo😄</title>
8  </head>
9  <body>
10     <script type="text/javascript">
11         console.log("Hola Mundo!");
12     </script>
13 </body>
14 </html>
```

6. INCLUIR CÓDIGO JAVASCRIPT

Podemos invocar un script con la etiqueta `<script src="ruta-fichero"></script>`

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hola Mundo👁️</title>
8  </head>
9  <body>
10     <script src="./app.js" type="text/javascript"></script>
11 </body>
12 </html>
```

JS app.js

```
1  console.log("Hola mundo!");
```

7. BOOTSTRAP

Haremos uso de esta librería para hacer interfaces “bonitas” sin tener que detenernos mucho en el código CSS.

Agregaremos en el `<head>` el código que viene en la página de [Get Started de Bootstrap](#) para incluir sus ficheros de estilos.

También será necesario incluir sus librerías de JS y las de JQuery para que funcionen algunos elementos.

Podemos instalar en VSCode la extensión Bootstrap 4 para que nos ayude a escribir estructuras de bootstrap gracias a sus “snippets”

7. BOOTSTRAP

El snippet b4-\$ nos creará una estructura básica de html incluyendo los css y js necesarios para que funcione Bootstrap

```
<!doctype html>
<html lang="en">
  <head>
    <title>Title</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-sc

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.c

  </head>
  <body>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3

  </body>
</html>
```