Desarrollo web

Sitios web

- Un sitio web es:
  - un **conjunto de recursos digitales** (*páginas, estilos, scripts, imágenes, datos, etc*.)
  - **organizados bajo un mismo dominio** y que
  - se almacenan en un **servidor web** y
  - están **accesibles a través de Internet mediante un navegador**.

Sitios web tradicionales vs modernos

- **Sitio web tradicional**
  - ✓ **varias páginas web relacionadas entre sí.**
  - ✓ Cada vez que se hace clic en un enlace, el navegador pide al servidor una nueva página *(archivo) (midominio.com/contacto.html)*.
  - ✓ Cada recurso HTML es un archivo físico **distinto en el servidor web** (*index.html, contacto.html, productos.html*)
  - ✓ **Cada página implica RECARGAR TODO EL DOCUMENTO**
- **Sitio web SPA** (*Single Page Applications*)
  - ✓ **Una sola página web principal** *(un único index.html)*.
  - ✓ El navegador descarga **un único archivo HTML** del servidor (el index.html)
  - ✓ A partir de ahí, el framework (*Angular, React, Vue, etc)* usa **JavaScript para manejar las rutas** *(por ej. /contacto, /productos)* sin que el navegador tenga que pedir otro HTML distinto. Así, dentro de esa página se **cargan dinámicamente los contenidos** que el **enrutador de la SPA** decide que hay que mostrar.
  - ✓ En **enrutamiento lo maneja JavaScript en el cliente.**
  - ✓ *Ejemplo:*

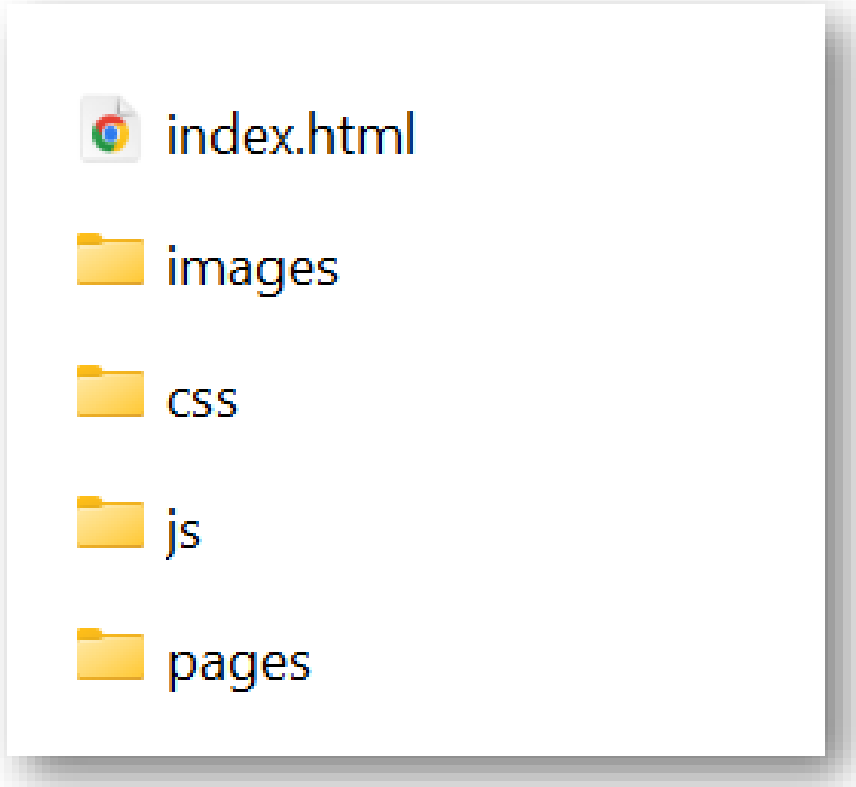
    entro en /midominio.com/contacto → el servidor devuelve index.html → Angular renderiza la vista de contacto.
  - ✓ ***Nunca se recarga la página entera, sólo se actualizan las partes necesarias EN LA VISTA.***
    - ✓ ***El usuario cree que sí que está navegando por varias páginas.***

Sitios web tradicionales vs modernos

- **Sitio web PWA** (*Progressive Web Apps*)
  - ✓ Una **PWA** puede ser una **SPA o un sitio web tradicional.**
  - ✓ La diferencia con las anteriores es que **añade:**
    - **service workers** : controla la caché, notificaciones push, offline.
    - **manifest.json** : define icono, nombre, instalación como app.
    - **caché inteligente**
  - ✓ Se comporta como **una app nativa,** pero sigue siendo web.
  - ✓ Si es **SPA**, sigue la lógica de **un solo HTML base + recursos dinámicos.**
  - ✓ Si es **multipágina**, puede seguir sirviendo distintos HTML.

Estructura de un sitio web básico

- Archivo **index.html**
- Carpeta **images**
- Carpeta **css**
- Carpeta **js**
- Carpeta **pages:** *resto de archivos .html*
- Carpeta **assets:** *otros recursos como iconos, vídeo, audio, etc.*



Nombres de archivos

- Evitar usar espacios
- En **minúsculas** **SIEMPRE**
- **Usar guión (-) para separar palabras o guión bajo “_”.**
- **Sin caracteres especiales.**
- **Extensiones correctas:**
    - *HTML → .html*
    - *CSS → .css*
    - *JavaScript → .js*
    - *Imágenes → .jpg, .png, .svg, .gif, .webp*

Dominios y URL

- **IP** (Internet Protocol) =dirección única ordenador
- **DOMINIO** = nombre que identifica a cada servidor conectado a Internet. La **extensión indica el propósito del sitio web**.

```
www.ejemplo.com.
|   |       |   └── Dominio raíz (.)
|   |       └────── Dominio de nivel superior (TLD) → .com
|   └──────────────── Dominio de segundo nivel (SLD) → ejemplo
└──────────────────── Subdominio → www
```

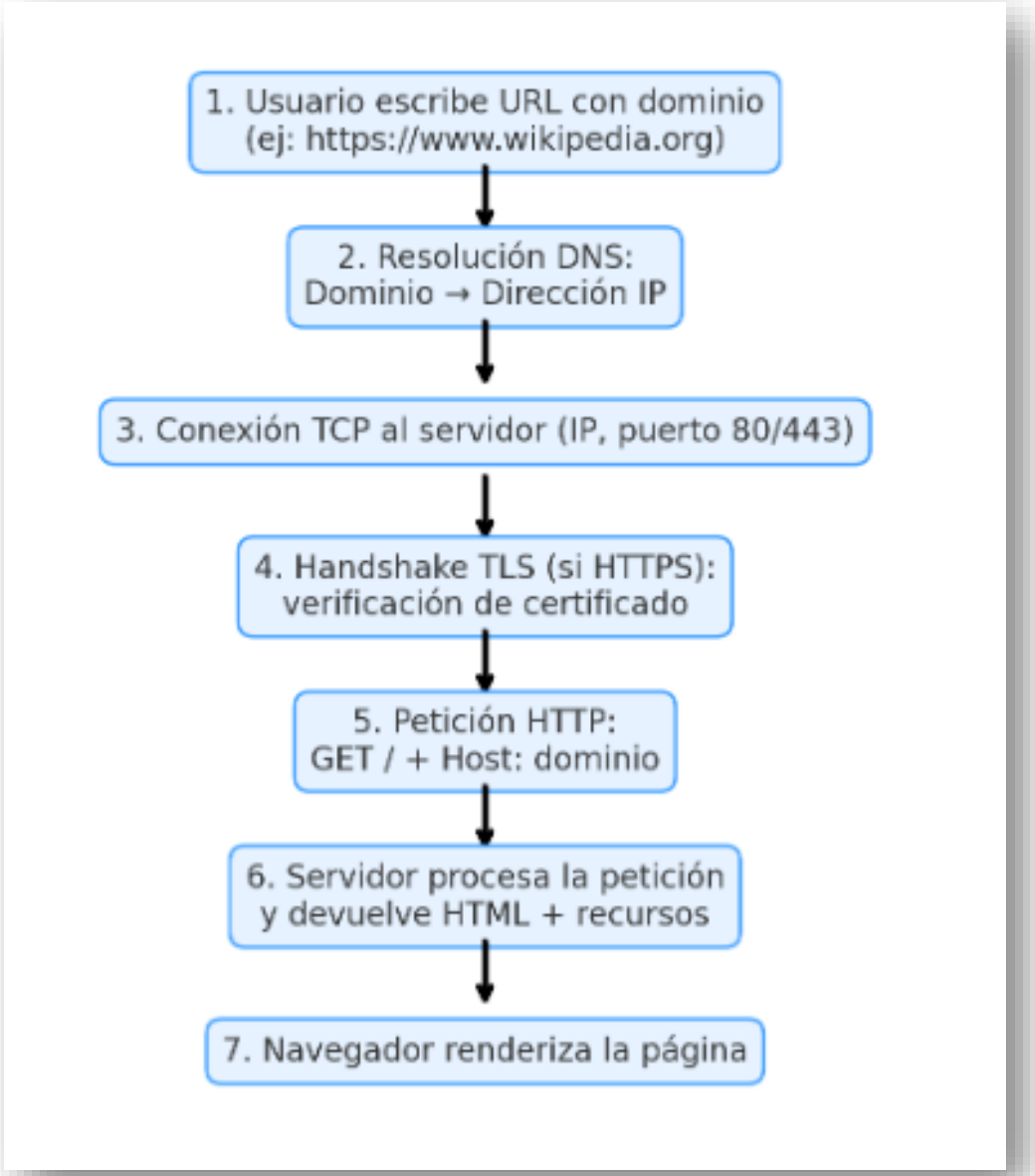
- **Servidor DNS** = contiene una lista de dominios con sus correspondientes IPs y realiza la traducción correspondiente.
- **URL** = dirección completa que se escribe en el navegador para acceder a un recurso en Internet (*una página, imagen, etc)*.

Indica:

- **Qué protocolo** usar (HTTP, HTTPS, FTP, ...)
- **Qué dominio o IP** buscar
- **Qué recurso específico** dentro del servidor queremos (ruta y archivo).

http://www.ejemplo.com/contacto.html

Protocolo | Dominio | Recurso



Partes de una URL

1. **Protocolo** → https://
   - Indica cómo se comunica el navegador con el servidor.
   - Habituales: http:// , https:// , ftp:// , mailto:
2. **Dominio o IP** → www.ejemplo.com
   - Nombre legible que se traduce a una IP mediante DNS
3. **Puerto** (opcional) → :80
   - Puerto de conexión en el servidor = **qué servicio de ese servidor**
   - Por defecto: 80 (HTTP) y 443 (HTTPS).
   - Solo se escribe si es distinto del estándar.
4. **Ruta** → /carpeta/pagina.html
   - Indica la ubicación del recurso dentro del servidor.
5. **Parámetros (query string)** → ?usuario=ana
   - Información extra que se envía al servidor (normal en formularios o filtros)
6. **Fragmento o ancla o hash** → #seccion2
   - Marca un punto concreto dentro de una página (no viaja al servidor, lo interpreta el navegador)

- Página web:

```arduino
https://es.wikipedia.org/wiki/HTML
```

- Imagen:

```arduino
https://www.ejemplo.com/img/logo.png
```

- Enlace con parámetros:

```arduino
https://tienda.com/productos?categoria=ropa&orden=precio
```

```
http://www.ejemplo.com:8080/pagina.html
```

Hipervínculos o enlaces

- **Son referencias a otros recursos** con un solo clic.
- Se pueden definir con **URLs absolutas o relativas.**
- Pueden conectar a:
  - **Externos**: Otra página web
  - **Internos**: Otro recurso dentro del mismo sitio web
  - Un **archivo descargable**
  - Un **correo electrónico**
  - **Anclas o hash**: saltan a una parte específica de la misma página.

```
<a href="https://www.wikipedia.org">Ir a Wikipedia</a>
```

```
<a href="/contacto.html">Contacto</a>
```

```
<a href="documentos/manual.pdf" download>Descargar manual en PDF</a>
```

```
<a href="mailto:profesor@ejemplo.com">Escríbeme</a>
```

```
<a href="#capitulo3">Ir al Capítulo 3</a>
```

URL absolutas y relativas

- **URL absoluta:**
  - Es la **dirección completa** de un recurso en Internet, que incluye el **protocolo, dominio y ruta completa.**
  - **Es única**
  - Sirve para acceder al recurso **desde cualquier lugar,** porque es única.
  - **Características:**
    - Siempre empieza por `http://` o `https://`
    - Se puede usar desde cualquier web del mundo.
    - **No depende de dónde esté ubicado el archivo actual.**

```
<a href="https://www.wikipedia.org/wiki/HTML">Artículo sobre HTML en Wikipedia</a>
<img src="https://misitio.com/imagenes/logo.png" alt="Logo">
```

URL absolutas y relativas

- **URL relativa:**
  - Es una **dirección parcial,** que indica la ubicación de un recurso en Internet, **respecto al archivo actual.**
  - **No incluye dominio, ni protocolo.**
  - Depende de la **estructura de carpetas** del proyecto.
  - **Características:**
    - Son más cortas y fáciles de manejar en proyectos locales.
    - Si mueves el archivo a otra carpeta, **pueden dejar de funcionar.**
    - **Se usan mucho en desarrollo web (*sobre todo, en proyectos grandes con muchas páginas)***

```html
<!-- Archivo en la misma carpeta -->
<a href="contacto.html">Contacto</a>

<!-- Archivo dentro de una carpeta -->
<img src="imagenes/logo.png" alt="Logo">

<!-- Archivo en la carpeta superior -->
<a href="../index.html">Volver al inicio</a>
```

DOM (Document Object Model)

- Es una **representación estructurada** de una página web que el **navegador crea** a partir del código HTML que recibe.
- El navegador convierte HTML en un **árbol de objetos o NODOS** que **JavaScript puede leer y modificar**.

```html
<html>
  <body>
    <h1>Hola mundo</h1>
    <p>Este es un párrafo</p>
  </body>
</html>
```

navegador

```
document
└── html
    └── body
        ├── h1 ("Hola mundo")
        └── p ("Este es un párrafo")
```

JavaScript

```javascript
document.querySelector("h1").textContent = "Título cambiado";
```