

Glosario full stack

Flujo básico del desarrollo de una aplicación

Inicio → **Interfaz de usuario** → **Conexión a BD** → **Fin (listo para producción)**

Inicio: Proyecto desde cero: planear la idea, estructuras carpetas, instalar herramientas, etc.

Interfaz de usuario (UI): lo que el usuario ve e interactúa, botones, formularios, menús, colores, fuentes (HTML, CSS, JavaScript, React, etc).

Conexión a Base de Datos: conecta la lógica y seguridad de backend a una base de datos para el flujo de datos.

Relacionales: MySQL, PostgreSQL (usan tablas como Excel).

No relacionales: MongoDB (usan estructuras tipo JSON)

Guardar tareas nuevas

Consultar tareas existentes

Eliminar o editar datos

Fin (listo para producción): después de probar y ajustar todo, la app está lista para subirla a internet (publicar) "desplegar a producción".

a

Accesibilidad: principio de diseñar sitios y aplicaciones para que todas las personas puedan usarlos, independientemente de sus capacidades físicas, cognitivas o tecnológicas. Mejora la experiencia de usuario y beneficia el SEO.

Principio	Definición	Ejemplo
Perceptible	El contenido debe poder verse, oírse o percibirse de algún modo.	Usar alt="Descripción" en imágenes para lectores de pantalla, Contraste alto entre texto y fondo para usuarios con baja visión
Operable	Navegar y usar la web con teclado, mouse u otras ayudas.	Navegación completa con teclado (sin mouse)
Comprensible	Contenido claro, lógico y fácil de entender.	Formularios con etiquetas (<label>) claras
Robusto	Funciona en dispositivos, navegadores, tecnologías de asistencia (como lectores de pantalla) y soporte futuro.	Responsive

Elementos accesibilidad

Etiquetas semánticas	<code><header></code> , <code><nav></code> , <code><main></code> , <code><section></code> , <code><article></code> , <code><footer></code> , <code><button></code> ...
Atributos utiles	Alt Descripción de imágenes.
	Label Describe entradas de formularios.
	Tabindex Controla el orden de navegación con teclado.
	Title Da información extra en tooltip (no se debe abusar).

Atributos HTML: `<tag atributo="valor">` se escriben dentro de la etiqueta de apertura

Clave → el nombre del atributo (src, href, alt, class, id)

Valor → lo que ese atributo contiene o define (URL, clase CSS, texto alternativo)

Se unen con =

El valor siempre va entre comillas " " (dobles preferidas)

ej ``

Etiqueta: ``



Clave: `src`

Valor: `"foto.png"`

Relación: `src="foto.png"`

ARIA: Accessible Rich Internet Applications atributos en las etiquetas HTML para mejorar accesibilidad, para elementos no semánticos o necesitan más contexto para tecnologías asistivas.

Atributos ARIA

Atributos comunes javascript	Uso	Ejemplo	Valor
role	Define qué "tipo de elemento" es.	<code>role="button"</code>	texto
aria-label	Proporciona una etiqueta legible para lectores de pantalla	<code><button aria-label="Buscar"></code>  <code></button></code>	texto
aria-hidden	Oculto el contenido a tecnologías asistivas	<code></code>	true / false
aria-pressed	Indica si un botón está activado (usado en botones tipo toggle)	<code>role="button" tabindex="0"</code> <code>aria-pressed="false"</code> <code>onclick="toggleLike(this)"></code>  Me gusta	true / false

Glosario full stack

aria-expanded	Indica si un panel desplegable está abierto o cerrado	<code><button aria-expanded="false" aria-controls="detalle">Ver más </code>	true / false
aria-live	Anuncia dinámicamente cambios de contenido (útil en chats, alertas, etc.)	<code>aria-live="polite"</code>	polite / assertive / off
aria-checked	Si un checkbox está marcado	<code>role="checkbox" tabindex="0" aria-checked="false" onclick="toggleCheck(this)">  Acepto los términos </div></code>	true / false

aria-live

Opciones	Función
off	No lee cambios
polite	Lee el cambio cuando pueda, sin interrumpir
assertive	Lee el cambio inmediatamente, incluso interrumpiendo

Cuando usar

Atributos comunes	Uso
label	siempre que se pueda → más claro para todos
placeholder	como ejemplo, no como única explicación
aria-label	cundo no hay texto visible, como en botones con solo íconos

b

Bootstrap: framework responsivo, se puede integrar mediante CDN, contiene un sistema de grillas, reduce uso de CSS, gran documentación (utilities) y componentes.

Viewports y breakpoints para ser responsive

Container (ancho fijo cambia según el dispositivo) y container-fluid (ancho 100%)

Sistema de grillas: 12 columnas; ordenadas por row (filas), col (columnas) (según el dispositivo ej. col-xs-12 col-sm-6 col-md-4), gutters (espacios/márgenes horizontales y verticales entre columnas en el sistema de grid)

btn: se usa para crear botones con estilo ya listo (bootstrap)

c

CDN: Content Delivery Network, enlace externo para cargar **bootstrap** desde internet, sin tener que descargar archivos localmente. Se coloca en el `<head>` (link) o antes del `</body>` (script)

Glosario full stack

CSS: Cascading Style Sheets, hojas de estilos en cascada, lenguaje de diseño gráfico, qué elementos seleccionar y describe los estilos que se van a aplicar a esos elementos. (documentos HTML o XML).

d

Declaración: lo que esta dentro de llaves { }

div: `<div>` `</div>` división, **contenedor en bloque** para organizar y estructurar contenido en HTML, suelen tener contenido similar relacionado entre sí.

e

Etiquetas: rodean el texto que debe mostrarse de forma específica o transformarse, suelen tener apertura `<etiqueta>` y cierre `</etiqueta>` algunas son auto-cerradas, como ``

Bloque: Ocupan todo el ancho disponible de su contenedor, siempre comienzan en una **nueva línea**, estructura el contenido.

Etiquetas comunes:

`<div>`, `<p>`, `<h1>` a `<h6>`, ``, ``, ``, `<section>`, `<article>`, `<main>`, `<footer>`, `<header>`, `<nav>`, `<form>`, etc.

Linea (inline): Ocupan solo el ancho necesario del contenido que encierran, no cortan la línea, pueden estar dentro de un bloque o junto a otras etiquetas inline.

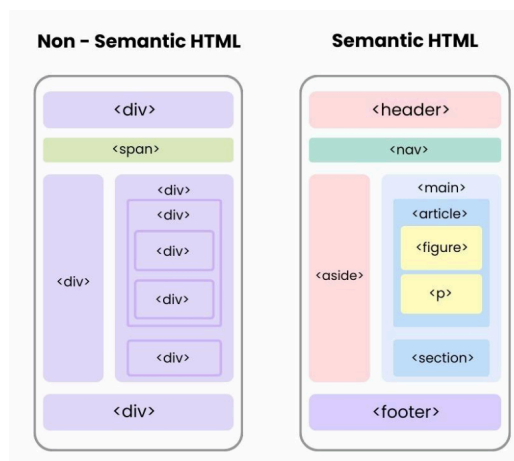
Etiquetas comunes:

``, `<a>`, ``, ``, ``, `<label>`, `<input>`, `<small>`, `<abbr>`, `
`, etc.

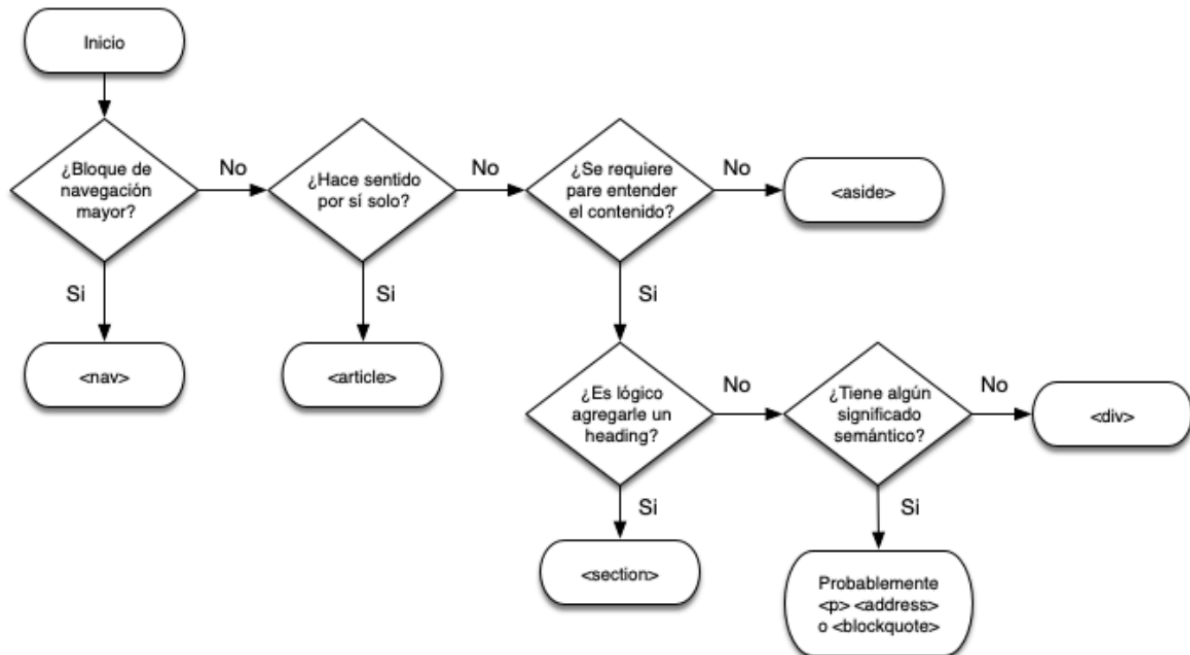
Elemento: Una etiqueta y cualquier contenido entre la apertura y cierre.

Semánticos: Indica la finalidad del contenido que se encuentra dentro de una etiqueta. Ordena contenido para facilitar accesibilidad.

No semánticos: No especifica el contenido dentro de una etiqueta, si no la estructura de los contenidos dentro de la web y/o para uso de estilos.



Glosario full stack



Entidades HTML: siempre empiezan con & y terminan con ;

Para indicar	Escribir..
< (menor que)	<
> (mayor que)	>
& (ampersand)	&
" (comillas dobles)	"
' (comillas simples)	'

f

Favicon: icono de página

se agrega en el <head>

<link rel="icon" href="favicon.ico" type="image/x-icon">

Formatos: tipos de archivo comunes usados en desarrollo web

Imagen: .jpg, .png, .svg, .webp, .gif

Documento: .html, .css, .js, .pdf

Iconos (gráficos): .ico, .svg, .png

Videos: .mp4, .webm, .ogg

Audio: .mp3, .wav, .ogg

Imagen

Formato	Características
JPEG	Joint Photographic Experts Group, lossy.
PNG	Portable Network Graphic, sin fondo, + calidad sin comprensión de pérdida.
WebP	Previene pérdida de calidad, no es amplia su compatibilidad
SVG	Scalable Vector Graphics, basado en XML para gráficos bidimensionales, estáticos y animados. Es escalable sin perder calidad, a diferencia de los formatos basados en píxeles.

Formularios `<form>`: recoge datos del usuario a través de controles de entrada (inputs, selects, checkboxes, etc.) y enviarlos a un servidor o procesarlos con JavaScript.

```
<form action="/ruta" method="post">
```

<!-- Controles de entrada -->

```
</form>
```

action: Define a dónde se enviarán los datos (URL del servidor).

method: Define cómo se enviarán los datos (GET, POST, etc.).

Atributos de entrada: propiedades que se pueden aplicar a cualquier tipo de entrada para controlar su comportamiento o identificación:

Atributo	Descripción	Ejemplo
name	Nombre del dato (clave al enviarse)	name="usuario"
id	Identificador único (para JS o label)	id="campo1"
type	Tipo de entrada	type="email"
placeholder	Texto guía dentro del campo	placeholder="Tu correo"
required	Hace obligatorio el campo	required
value	Valor por defecto	value="Ejemplo"
min / max	Valor mínimo y máximo (para number, date, time)	min="1" max="10"
checked	Marca seleccionada por defecto (radio/checkbox)	checked
readonly	Solo lectura	readonly
disabled	Campo deshabilitado (no editable)	disabled

id debe coincidir con **for**

```
<form action="">
```

```
  <label for="newName">Name</label>
```

```
  <input type="text" name="Name" placeholder="write ur name" id="newName">
```

```
</form>
```

Glosario full stack

Métodos HTTP en formularios (requieren JavaScript o frameworks como fetch o Axios).

Método	Función	Uso común
GET	Envía los datos en la URL	Formularios simples, búsqueda
POST	Portable Network Graphic, sin fondo, + calidad sin comprensión de pérdida.	Formularios con info sensible (login, registro)
PUT	Actualiza un recurso existente (más usado con JS/API)	Edición de datos
DELETE	Elimina un recurso (más usado con JS/API)	Eliminar cuentas, etc.

*JavaScript:

Valida datos antes de enviar (`onsubmit`)

Captura los datos del formulario (`event.target`, `FormData`)

Envía datos de forma dinámica (AJAX) sin recargar la página

Floats: casi obsoletos, alinea imágenes a un lado (`float: left/right`) reemplazados por flex y grid (`clearfix`, `overflow: hidden`, `flow-root..`)

h

<Head>: Contenedor de encabezado, incluye referencias a otros archivos que necesita la página o el sitio, puede incluir datos que los motores de búsqueda y los sitios de redes sociales usan para encontrar mejor su sitio web.

<Header>: Etiqueta semántica, representa la cabecera de una página web o de una sección, contiene elementos como logotipo, título, navegación principal, o datos introductorios. Puede usarse más de una vez, no usar dentro de `<footer>`, `<address>` ni dentro de otro `<header>`.

Hipertexto: es texto que contiene enlaces (hipervínculos) enlaza diferentes recursos (documentos, imágenes, correos, secciones internas).

Se implementa con la etiqueta `<a>` (de *anchor/ancla*) y el atributo `href` define a dónde lleva el enlace.

Todo **hipervínculo es un tipo de hipertexto**, pero no todo hipertexto tiene que ser solo texto (puede ser imagen, correo, conexión).

Hipervínculos: enlaces que permiten al usuario navegar a otras cosas.

`<a>` `` ancla, atributo llamado **href** (referencia de hipertexto)

HTML: HyperText Markup Language, lenguaje de marcado de hipertexto, lenguaje principal se usa para crear una página web, describir la información (enlaces o hipervínculos) que se va a mostrar al usuario, indica a un explorador cómo mostrar texto.

Glosario full stack

Estructura

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Comments <!-- Esto es un comentario en HTML -->

Etiqueta	Descripción
<!DOCTYPE html>	Indica la navegador como interpretar el documento. HTML5
<html></html>	Etiqueta raíz, indica el inicio y fin del documento en código HTML. Se puede agregar el atributo lang para indicar el idioma del contenido
<head></head>	Información para el navegador
<meta />	Metadatos (descripción, palabras clave y codificación de caracteres). Dato EXIF- Exchangeable Image File Format recopila información técnica. Charset: codificación de caracteres utf-8 para latinos.
<title></title>	Título del documento, tiene atribuciones semánticas, se visualiza en pestaña.
<link />	Relaciones entre el documento actual y otros recursos, como CSS, favicon...
<body></body>	Información para el usuario
<style></style>	Reglas de estilo.
<p></p>	Párrafo
<h1></h1> a <h6></h6>	Encabezados, siendo h1 el más grande, mas importante y solo uno por temas de accesibilidad
, , 	Un order list, order list, elemento lista
<a>	Enlace o vinculo a una pagina o archivo (url)
	Imagen
<div></div>	Contenedor genérico, agrupa contenido y aplica estilos o scripts a un bloque completo.
<table></table>, <thead></thead>, <tbody></tbody>, <tfoot></tfoot>	Table Table head encabezado de columna Table body datos Table foot pie de tabla

Glosario full stack

<code><tr></tr></code> , <code><th></th></code> , <code><td></td></code>	Table row fila de tabla Table head Table data
<code><script></script></code>	Documento JavaScript
<code><nav> </nav></code>	Contiene enlaces de navegación del sitio. (Menús, barras de navegación, enlaces a secciones internas)
<code><aside> </aside></code>	Información secundaria o complementaria, como barras laterales o datos relacionados. (Barras laterales, widgets, publicidad, biografías de autor, etc.)
<code><main> </main></code>	Contenido principal de la página. Solo debe aparecer una vez por documento. (+importante)
<code><article> </article></code>	Contenido que puede existir por sí solo y tiene sentido fuera del sitio. (Noticias, publicaciones de blog, comentarios, productos, publicaciones individuales)
<code><section> </section></code>	Secciones independientes, agrupa contenido relacionado temáticamente, (capítulos, bloques de contenido, formularios, etc.)
<code><figure> </figure></code>	Agrupar contenido independiente, como una imagen, gráfico, video, código o ilustración, se usa cuando el contenido está relacionado con el texto, pero también tiene sentido por sí mismo.
<code><figcaption> </figcaption></code>	Va dentro de <code><figure></code> , da una descripción o pie de figura (caption). Agrupa la imagen + explicación, mejora accesibilidad y SEO.
<code><footer></footer></code>	Define el pie de página de una sección o documento. Suele incluir información de contacto, créditos, enlaces legales, redes sociales, etc.
<code><address></address></code>	Define información de contacto relacionada con el autor o propietario del contenido.
<code><div></div></code>	Contenedor genérico en bloque. Se usa para agrupar elementos y aplicarles estilos o scripts.
<code></code>	Contenedor en línea. Se usa para aplicar estilo o manipular una parte específica del texto sin romper la línea.

Ejemplo.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Blog de Tecnología</title>
</head>
<body>

  <header>
    <h1>Blog de Tecnología</h1>
    <nav>
      <ul>
        <li><a href="#inicio">Inicio</a></li>
        <li><a href="#articulos">Artículos</a></li>
        <li><a href="#contacto">Contacto</a></li>
      </ul>
    </nav>
  
```

Glosario full stack

</header>

<aside>

<h2>Sobre mí</h2>

<p>¡Hola! Soy desarrolladora web. Aquí comparto artículos y más.</p>

</aside>

<main>

<section id="articulos">

<article>

<h2>¿Qué es HTML semántico?</h2>

<figure>

<figcaption>Uso correcto de etiquetas HTML5</figcaption>

</figure>

<p>

HTML semántico es usar etiquetas que describen el tipo de contenido que contienen.
Esto mejora la accesibilidad, el SEO y la estructura del código.

</p>

</article>

<article>

<h2>Cómo organizar una página con etiquetas modernas</h2>

<p>

`<header>`, `<main>`, `<section>` y `<footer>` separar las áreas visuales y funcionales del sitio.

Añade `<article>` contenido independiente y `<aside>` para información secundaria.

</p>

</article>

</section>

</main>

<footer>

<section id="contacto">

<h2>Contacto</h2>

<address>

Escrito por Lolita

ejemplo@mail.com

CDMX, México

</address>

</section>

</footer>

</body>

</html>

Glosario full stack

i

input `<input type="...">`

Tipo de entrada	Descripción	Ejemplo
button	Botón personalizado (no envía)	<code><input type="button" value="Haz clic"></code>
submit	Botón para enviar el formulario	<code><input type="submit"></code>
checkbox	Casilla de verificación (varias opciones posibles)	<code><input type="checkbox" name="acepto"></code>
radio	Opción única entre varias	<code><input type="radio" name="color" value="rojo"></code>
image	Imagen que actúa como botón de envío	<code><input type="image" src="boton.png"></code>
number	Campo numérico, puede tener min y max	<code><input type="number" min="1" max="10"></code>
password	Oculto caracteres	<code><input type="password"></code>

img: ``

atributos: **src** señala la ubicación de la imagen. **alt** describe el contenido de la imagen (motores de búsqueda y lectores de pantalla).

l

Label: etiqueta que dice qué significa el campo. Es visible para todos, se asocia al `input` con `for` y `id`.

Si es accesible, los lectores de pantalla pueden decir: "Nombre: campo de texto".

Lighthouse: herramienta creada por Google para analizar sitios web, puede examinar la optimización del motor de búsqueda (SEO) de una página, rendimiento de carga y otros procedimientos recomendados.

Link: `<link rel="stylesheet" href="style.css">`

atributos: **rel** identifica el tipo de recurso al que hace referencia. **href** identifica la ruta de acceso al archivo.

Listas: listas ordenadas `` o desordenadas ``

Atributos para modificarlas:

`type=""` en listas ordenadas (``) para cambiar el tipo de numeración ("1", "A", "a", "I", "i").

`style="list-style-type:"`: Se usa en CSS (`` o ``) personaliza el tipo de viñeta o numeración. (disc, circle, square, decimal, lower-roman, etc.)

p

Glosario full stack

Propiedad: ej. color, separada de un **valor**, orange, con dos puntos (:) y el final de la configuración punto y coma (;).

Placeholder: texto dentro de un input antes de que el usuario escriba algo. Sugerencia visual, **desaparece** al empezar a escribir. **NO** lo leen los lectores de pantalla.

```
<input type="text" placeholder="Escribe tu nombre">
```

r

README.md: documento en formato Markdown, sirve como presentación principal de un proyecto.

- Explica de qué trata el proyecto.
- Da instrucciones de uso o instalación.
- Muestra ejemplos, dependencias, o enlaces útiles.
- Es lo primero al abrir un repositorio en GitHub (se coloca en la raíz del proyecto).
- Escrito en Markdown (.md), un lenguaje sencillo para dar formato (negritas, títulos, listas, enlaces, imágenes).

<https://www.makeareadme.com/>
Comando Visual Studio Code
⌘ + Shift + V

Markdown

Resultado	Código Markdown
Negritas	**texto** o __texto__
Cursiva	<i>*texto*</i> o <i>_texto_</i>
Tachado	~~texto~~
Código en línea	`código`
Código en bloque	<pre><pre>
código
</pre></pre>
# Título 1	# Título 1
## Título 2	## Título 2
### Título 3	### Título 3
Lista con viñetas	- item o * item
Lista numerada	1. item
Enlace	[texto](https://ejemplo.com)
Imagen	![alt](url-imagen)
Linea horizontal	--- o ***

Role: atributo HTML de accesibilidad para decirle a los lectores de pantalla qué tipo de elemento (botones, menús o pestañas usando `<div>` o ``) Ayuda a personas con discapacidad visual.

Role	Indica..
button	Comportamiento de botón
checkbox	Casilla que se puede marcar o desmarcar
switch	Interruptor on/off
dialog	Ventana emergente o modal
navigation	Menú o sección de navegación

Glosario full stack

tab	Una pestaña
tablist	Grupo de pestañas
alert	Anuncio urgente

Rutas:

Absolutas: ubicación completa del archivo desde el dominio (protocolo http...) hasta el recurso de servidor externo.

``

Relativas: + flexible y portátil, ubicación del archivo actual, se usa para navegar dentro del mismo sitio.

``

s

<section></section>: se puede usar dentro de <main> para dividirlo en partes claras. Usar <section> si el bloque tiene un título propio (<h2>, <h3>..). Si no, probablemente conviene usar <div>.

El **atributo id** sirve para navegar directamente a esa parte desde un enlace (desde el menú como en:), aplica estilos CSS o interacciones JS a esa sección específica y/o identificar de forma única ese bloque dentro del HTML

: etiqueta HTML, genérica, no funciona por si sola. Agrupa texto o elementos en línea y aplicar estilos con CSS o comportamiento con JavaScript. (div selección de bloque y span selección línea).

: negritas

t

Tablas HTML: organiza datos en filas y columnas. Todas son bidimensionales, es decir, se componen de filas (<tr>) y columnas dentro de cada fila (<td> o <th>).

Atributos de tabla

colspan	rowspan
Une columnas en una sola celda.	Une filas en una sola celda.
td colspan="2">Celda que ocupa 2 columnas</td>	<td rowspan="3">Celda que ocupa 3 filas</td>

No semántica:

<table>: Contenedor general de la tabla.

<tr> (table row): Representa una fila.

<th> (table header): Representa celda de encabezado (se muestra en negritas y centrado por defecto).

<td> (table data): Representa celda de datos.

Semántica:

Glosario full stack

`<thead>`: Agrupa el encabezado de la tabla.

`<tbody>`: Agrupa el contenido principal.

`<tfoot>`: Agrupa el pie de tabla (puede aparecer arriba o abajo, es útil para cálculos, resúmenes, etc.).

target=blank: atributo que se usa en enlaces HTML (`<a>`) para abrir el vínculo en una nueva pestaña o ventana del navegador.

Otros valores: **_self** (por defecto en la misma pestaña), **_parent** (en el marco padre (usando frames o iframes anidados). Si no hay padre, actúa como **_self**.), **_top** (en el marco más alto (es decir, sale de cualquier iframe y carga en toda la ventana))

` Visitar sitio externo`

noopener: impide que la nueva pestaña tenga acceso a `window.opener` (evita ataques tipo *phishing* o *tabnabbing*).

noreferrer: evita que el sitio destino vea de dónde vino el usuario (no manda el `referer`).

Toggle button: (botón conmutador) cambia entre dos estados, como on/off, activado/desactivado, mostrar/ocultar. Visualmente es un botón normal, pero cada clic cambia su estado. Para accesibilidad, se usa **aria-pressed**, si está presionado (`true`) o no (`false`).

U

Utilies: (bootstrap) clases pequeñas y específicas que permiten modificar estilo sin escribir CSS manualmente.

Para indicar	Ejemplo	Función
Espaciado	<code>m-3, p-2</code>	Margin y padding
Texto	<code>text-center</code>	Alinear texto
Color	<code>text-danger</code>	Texto rojo
Fondo	<code>bg-dark</code>	Fondo oscuro
Display	<code>d-flex, d-none</code>	Mostrar, ocultar, o usar flex
Ancho/alto	<code>w-50, h-100</code>	Tamaños responsivos
Alineación	<code>align-items-center</code>	Usar Flexbox

```
<div class="bg-primary text-white p-3 text-center">
```

```
  Hola mundo
```

```
</div>
```

crea un bloque azul con texto blanco, centrado y con padding.