

HTML

Desarrollo de Software - UTN FRSFCO

2024

¿Con qué se hacen las páginas/aplicaciones web?

- ¿Qué es? (contenido) → **HTML**
- ¿Cómo se ve? (formato) → **CSS**
- ¿Cómo se comporta? (lógica) → **Javascript**

HTML

HyperText MarkUp Languaje

- Es un lenguaje de **marcado** (no es un lenguaje de programación)

Kids, violence is
never the answer



HTML is a Programming
Language



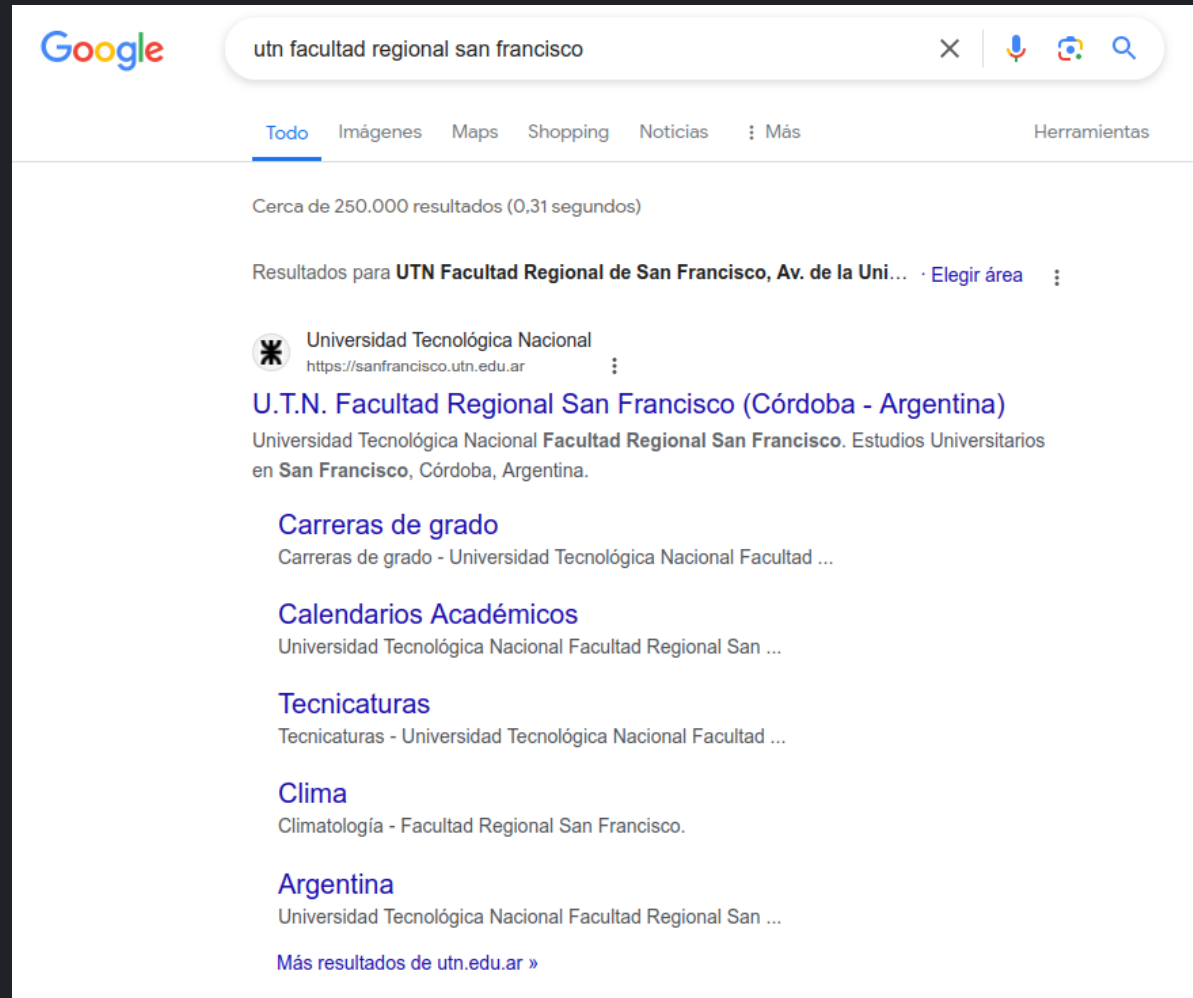
HTML

HyperText MarkUp Languaje

- Es un lenguaje de **marcado** (no es un lenguaje de programación)
- Toda la web está armada con ese lenguaje
- **Describe la estructura del contenido** (no el formato, no la lógica)
- ¿Por qué es importante hacerlo bien?
 - SEO (Search Engine Optimization)
 - Accesibilidad

HTML

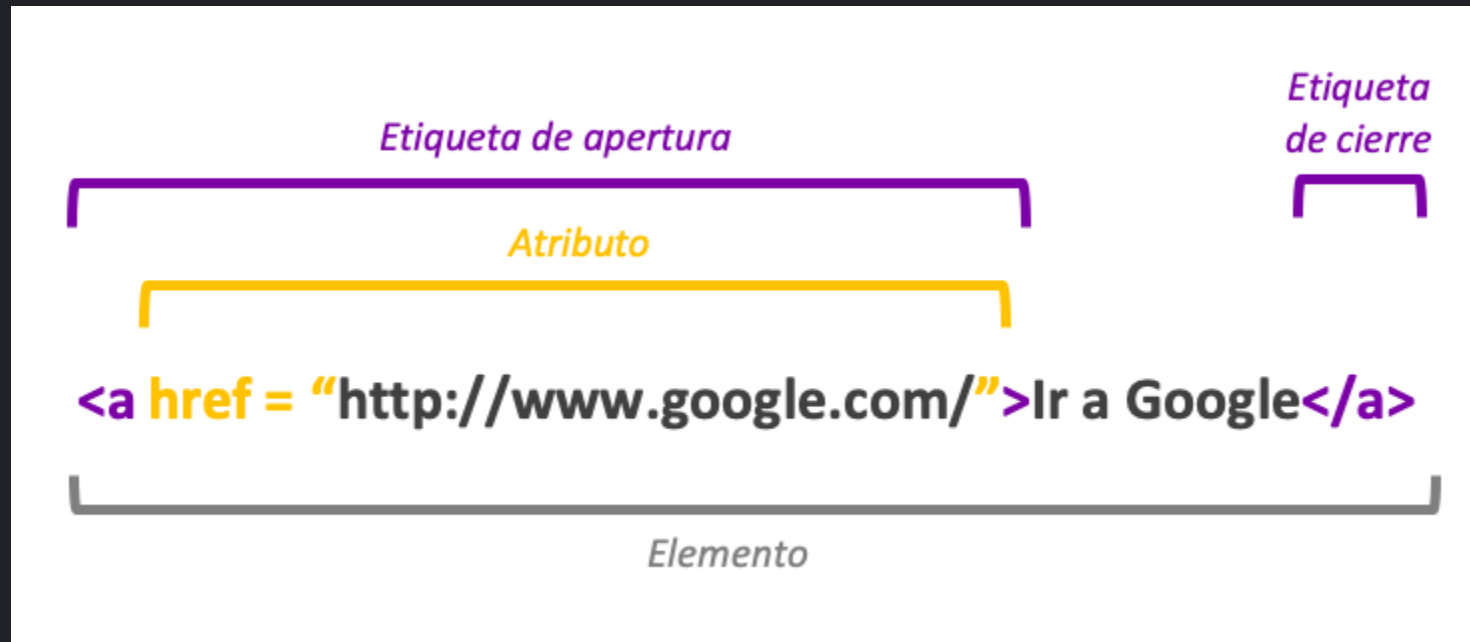
Ejemplo: ¿cómo sabe Google sobre las secciones?



Etiquetas (tags)

- El lenguaje está compuesto por **etiquetas**.
- Cada etiqueta describe un elemento del documento HTML.
- Cada etiqueta tiene:
 - apertura
 - atributos
 - valores para los atributos
 - contenido
 - cierre (no todas lo tienen)

Etiquetas (tags)



Títulos

h1, h2, h3, h4, h5, h6

Título de primer nivel, de segundo, de tercer nivel,
etc.

Ejemplo:

```
<h1>Soy un título</h1>
```

- Solo debería haber un único h1 por documento (ej: un libro no puede tener 2 títulos)
- El formato no tiene que ver con el elemento

Párrafos

```
<h1>Soy un título</h1>
```

```
<p>Y yo un párrafo</p>
```

Soy un título

Y yo un párrafo

Elementos anidados

- Las etiquetas HTML pueden anidarse (una dentro de otra). Esto da la estructura del documento.

```
<p>  
  +15 años de experiencia como </strong>Ingeniero de Software</strong>  
</p>
```

- `` indica que eso va **resaltado** NO EN NEGRITA!!!

Lista desordenada

(el orden no es importante)

```
<ul>  
  <li>un item</li>  
  <li>otro item</li>  
</ul>
```

- un item
- otro item

ul: unordered list

Lista ordenada

```
<ol>  
  <li>un item</li>  
  <li>otro item</li>  
</ol>
```

1. un item
2. otro item

ol: ordered list

Imágenes

```

```

- La etiqueta no lleva cierre.
- El atributo **src** se usa para indicar la ruta de la imagen.
- El atributo **alt** es opcional, es un texto que se muestra si no se pudo cargar la imagen.
 - Sirve además para accesibilidad (por ejemplo lectores de pantalla).
 - Sirve a los robots para saber de qué es la imagen.

Imágenes

Rutas: relativas vs absolutas

1. `imagen.png` (R)
2. `./imagen.png` (R)
3. `img/imagen.png` (R)
4. `../img/imagen.png` (R)
5. `C:/Usuarios/pagina/img/imagen.png` (A)
6. `https://picsum.photos/200` (A)

Atributos

Proporcionan información adicional sobre el elemento

- **id**: identificador **único** para referirse a ese elemento.
- **class**: el elemento pertenece a un conjunto (una clase). Comparte características en común con otros.

```
<ol id="lista_items">  
  <li class="item">un item</li>  
  <li class="item">otro item</li>  
</ol>
```


Anclas (anchor)

Permite navegar interna y externamente por el documento.

```
<a href="#titulo3">Ir a titulo 3</a>
```

Navegación interna: conduce (scroll) al elemento con id `titulo3`.

Anclas (anchor)

```
<a href="https://www.google.com">Ir a titulo 3</a>
```

Navegación externa: conduce al enlace indicado.

```
<a href="https://www.google.com" target="_blank">Ir a titulo 3</a>
```

Tip: abrir enlace en una pestaña nueva

Anclas (anchor)

Otras posibilidades para el atributo `href` :

- `mailto:pabratte@gmail.com`
- `tel:+5493456123456`
- `https://wa.me/15551234567?text=Hello!`

Etiquetas obsoletas

Antiguamente se usaban algunas etiquetas que daban formato! No usar mas!

- `b` (bold)
- `center`

Comentarios

```
<!-- esto es un comentario -->
```

Otras observaciones

Al navegador no le importan los saltos de línea en el código, por ejemplo:

```
<p>Esto es un párrafo</p>
```

da el mismo resultado que

```
<p>Esto  
    es un  
    párrafo  
</p>
```

Estructura básica de un documento HTML

```
<!DOCTYPE html>
<html>
  <head>

  </head>

  <body>
    <h1>titulo</h1>
    <p>esto es un párrafo</p>
  </body>
</html>
```

doctype

- La primer línea de un documento HTML.
- Informa al navegador qué versión de HTML (o XML) se usó para escribir el documento.
- Es una declaración no una etiqueta.

head

- Metadatos (datos sobre los datos), contiene información sobre la página.
- Nada de lo que pongamos aquí se renderiza dentro de la ventana del navegador.

head

Algunas cosas que pueden ir dentro del head:

```
<meta charset="utf-8">  
<title>Título de la pestaña</title>  
<link rel="icon" type="image/png" href="path-to-favicon">
```

body

- El contenido del documento, lo que renderiza el navegador.
- Las etiquetas que estuvimos viendo hasta ahora.

HTML Semántico (HTML5)

Semántico → significado

Etiquetas genéricas:

- `<div>` → división
- ``

Sirven para agrupar elementos pero no aportan significado.

Antes de HTML5...

```
<body>
  <div class="header">
    <div class="nav">
      <div></div>
      <div></div>
      <div></div>
    </div>
  </div>
  <div class="content-container">
    <div id="main-content">

      </div>
    </div>
    <div class="footer">
    </div>
  </body>
```

Antes de HTML5...

- Todas las páginas seguían una estructura general:
 - Encabezado
 - Barra de navegación
 - Contenido
 - Footer
 - Etc...

Se usaban clases o ids para hacer referencia al significado de cada elemento sección, pero no había convenición o reglas fijas.

Antes de HTML5...

- Era todo un lío (para los robots que trataban de parsear el significado de la página).
- **No quiere decir que no se puedan utilizar** `div` o `span`, a veces son necesario para agrupar elementos y estilizarlos.
- Pero se puede hacer mejor: utilizando elementos que aporten semántica.

Nuevas etiquetas agregadas en HTML5

- `header`
- `nav`
- `section`
- `article` → bloque independiente: al sacarlo y ponerlo en otro documento no pierde el sentido.

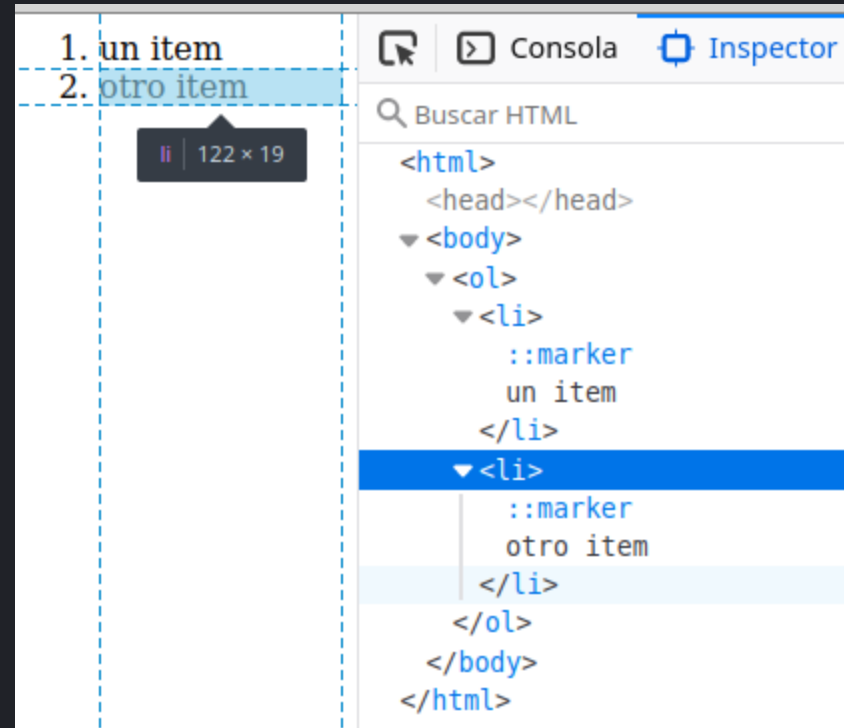
Nuevas etiquetas agregadas en HTML5

- `aside` → lateral en cuanto a contenido
- `main` → solo puede haber una por página, contenido principal, que hace a esa página especial y única.
- `footer`
- `details`
- `summary`

Nuevas etiquetas agregadas en HTML5

- De nuevo: las etiquetas son para **APORTAR SEMÁNTICA** al documento, no para definir cómo se ve.
- Ver listado completo de todas las etiquetas (son muuuchas) en: <https://www.w3schools.com/tags/>

Inspector / herramientas del desarrollador



Formularios

Recolectar información que ingresa el usuario y enviarla a un servidor remoto.

```
<form method="post" action="https://server.com/pagina">
  <label for="first_name">Nombre:</label>
  <input name="first_name" id="first_name" placeholder="Lalo" required>

  <label for="last_name">Apellido:</label>
  <input name="last_name" id="last_name" placeholder="Landa" required>

  <input type="submit" value="Enviar datos">
</form>
```

Nombre: Apellido:

Formularios

Atributos del formulario:

- `action` : la URL donde se enviarán los datos recolectados.
- `method` : método HTTP a utilizar. Generalmente POST. Vemos más adelante de qué se trata esto...

Formularios

- `label`: etiqueta para un input. El atributo `for` sirve para asociar la etiqueta a un `input`.
- `input`: el control que registra la entrada de usuario. El atributo `name` es el nombre de la "*variable*" donde se va a guardar lo que el usuario ingreso.

Formularios

Al hacer clic en el `<input type="submit">` se recolectan los datos y se genera un pedido HTTP como el siguiente:

```
POST /pagina HTTP/1.1
Host: server.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 31

first_name=Lalo&last_name=Landa
```

input

- `<input type="password">`
- `<input type="email">`
- `<input type="tel">`

input (radio)

```
<p>Medio de contacto: </p>
<input id="contact_choice_1" type="radio" name="contact_method" value="1">
<label for="contact_choice_1">Email</label>
<input id="contact_choice_2" type="radio" name="contact_method" value="2">
<label for="contact_choice_2">Teléfono</label>
```

Medio de contacto:

☐ Email ☒ Teléfono

- Email es lo que el usuario selecciona
- 1 es el valor que se se envía.

input (checkbox)

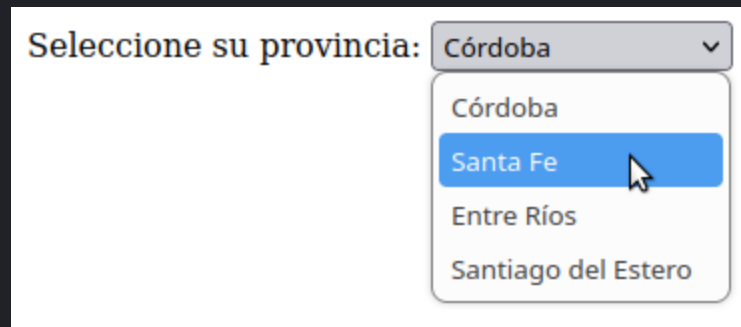
```
<input id="remember_me" type="checkbox" name="remember_me" value="true">  
<label for="remember_me">Recordar mis datos</label>
```

☐ Recordar mis datos

- **IMPORTANTE:** sólo se envía el valor de `value` si el checkbox está activo. No se envía nada de lo contrario.

input (select)

```
<label for="state">Seleccione su provincia: </label>
<select name="state">
  <option value="1">Córdoba</option>
  <option value="2">Santa Fe</option>
  <option value="3">Entre Ríos</option>
  <option value="4">Santiago del Estero</option>
</select>
```



Seleccione su provincia: Córdoba ▼

- Córdoba
- Santa Fe**
- Entre Ríos
- Santiago del Estero

- Santa Fe es lo que el usuario selecciona
- 2 es el valor que se se envía.

button

```
<button>Click me</button>
```

Cuando tenemos un `button` dentro de un formulario, el comportamiento por defecto ante el clic es enviarlo.

video

```
<video width="320" height="240" controls>  
  <source src="http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ElephantsDream.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Your browser does not support the video tag.  
</video>
```

audio

```
<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
  Your browser does not support the audio element.  
</audio>
```

iframe

Se usa para embeber una página web dentro de otra.

Útil, por ejemplo, para embeber videos de YouTube.

```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/3nYLTiY5skU?si=VHjaMyW7yrsMLd5T"  
title="YouTube video player"  
frameborder="0" allow="accelerometer;  
autoplay; clipboard-write;  
encrypted-media; gyroscope;  
picture-in-picture; web-share"  
referrerpolicy="strict-origin-when-cross-origin"  
allowfullscreen></iframe>
```