

Guía de ejercicios 1 - HTML



¡Hola! Te damos la bienvenida a tu primera guía de ejercicios.

¿En qué consiste esta guía?

Esta guía te ayudará a profundizar los conocimientos vistos en la clase y te entregará las herramientas necesarias para resolver el desafío, si bien no todos los aprendizajes entregados en esta guía son necesarios para resolver el desafío, los contenidos cubiertos son muy útiles en el día a día trabajando como desarrollador web.

Además, en esta guía repasaremos algunos conceptos básicos vistos en clases sobre HTML, aprenderemos cómo estructurar un proyecto web cuando tenemos múltiples archivos y realizaremos una actividad estructurando un proyecto que nos servirá como base para el desafío, luego profundizaremos en las etiquetas que van en el head de un documento HTML y aprenderemos sobre codificación, finalmente profundizaremos en las etiqueta `` `<a>` y aprenderemos a crear tablas ocupando html.



¡Vamos con todo!

Tabla de contenidos

| | |
|--|----------|
| Guía de ejercicios 1 - HTML | 1 |
| ¿En qué consiste esta guía? | 1 |
| Tabla de contenidos | 2 |
| ¿Qué es HTML? | 3 |
| Estructura base de un documento HTML | 3 |
| Estructura de un proyecto web | 4 |
| Actividad 1: Estructurando un proyecto web | 6 |
| Profundizando en las etiquetas dentro HEAD | 7 |
| Título | 7 |
| Codificación | 7 |
| Favicon | 8 |
| Actividad 2: El head de nuestro nuevo curriculum | 9 |
| Profundizando en etiquetas dentro del body | 9 |
| Imágenes | 9 |
| Atributo alt | 10 |
| <a> Links | 10 |
| Links a otros sitios | 11 |
| Links a páginas dentro de nuestro sitio | 11 |
| Imágenes con links | 12 |
| <table> Tabla | 13 |
| Actividad 3: Tablas | 15 |
| Actividad 4: Investigando sobre thead y tbody | 15 |
| Actividad 5: Revisando los aprendizajes | 16 |



¡Comencemos!

¿Qué es HTML?



Sigla que proviene del inglés **HyperText Markup Language**, y que podemos traducir como un **lenguaje de marcas de hipertexto**, que sirve para la elaboración de páginas web.

HTML es un lenguaje de marcas que se utiliza para estructurar y escribir contenido, el cual es interpretado por un **navegador web**.

Conocer este lenguaje y su alcance, es el primer paso para empezar a comprender cómo crear sitios que se adapten a las necesidades del mercado, con **buenas prácticas de programación** y una visión completa del proceso de desarrollo.

Estructura base de un documento HTML

HTML se organiza en base a **etiquetas**, que son los elementos con los que puedes **dar formato y estructura a un archivo HTML**. Cada una de ellas apunta a diferentes tipos de elementos, además, cada una tiene un formato determinado por defecto (que revisaremos más adelante). A continuación, te mostramos un ejemplo:

```
<p> Hola </p> <!-- esto es una etiqueta! -->
```

Por lo general, la estructura o **sintaxis** de una etiqueta es la siguiente:

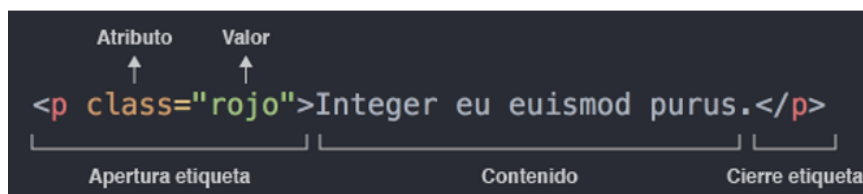


Imagen 1. Sintaxis de etiquetas.

Fuente: Desafío Latam.

Como puedes observar en la imagen, una etiqueta cuenta con una apertura, contenido y cierre.

Cada etiqueta puede contener ciertos **atributos** junto a su valor, por ejemplo, a la etiqueta **<p>** creada anteriormente, se le da un **atributo class** al que le asignaremos su valor. Este puede ser un nombre que definamos o alguna palabra específica que definirá el comportamiento de la etiqueta, en este caso el valor es "rojo".



En la próxima unidad usaremos este tipo de atributos y valores para dar estilo a nuestras páginas web.

La estructura básica de un archivo HTML consiste principalmente en una cabeza (<head>) y un cuerpo (<body>), la cual, podemos observar en la siguiente imagen:

```
<!DOCTYPE html>
<html>
<head>
  <!-- Aquí va la información para el navegador -->
</head>
<body>
  <!-- Aquí va el contenido para el usuario -->
</body>
</html>
```

La primera etiqueta `<!DOCTYPE html>` es la que le indica al navegador cómo debe interpretar el resto del documento (el doctype especificado es de HTML5, el cual es el estándar de hoy).

Luego, el `<head>` contiene toda la información que es para el navegador y el `<body>` contiene todo el contenido que es para el usuario.

Finalmente, la etiqueta `<html>` especifica que todo lo que está dentro de ella deberá ser interpretado como HTML.

Estructura de un proyecto web

A medida que un proyecto web crece, la cantidad de archivos también lo hace, y eso puede dificultar la mantención si no se mantiene una estructura adecuada.

La estructura depende del alcance del proyecto, pero existe una convención frecuentemente utilizada que propone un modelo basado en la organización por tipos de archivos. En este modelo, los archivos HTML se guardarán en el directorio principal, mientras que los recursos adicionales como imágenes, vídeos, hojas de estilo, etc, se almacenarán en subcarpetas dentro de una carpeta común llamada “assets”, como se grafica en la imagen. Por ahora utilizaremos esta estructura:

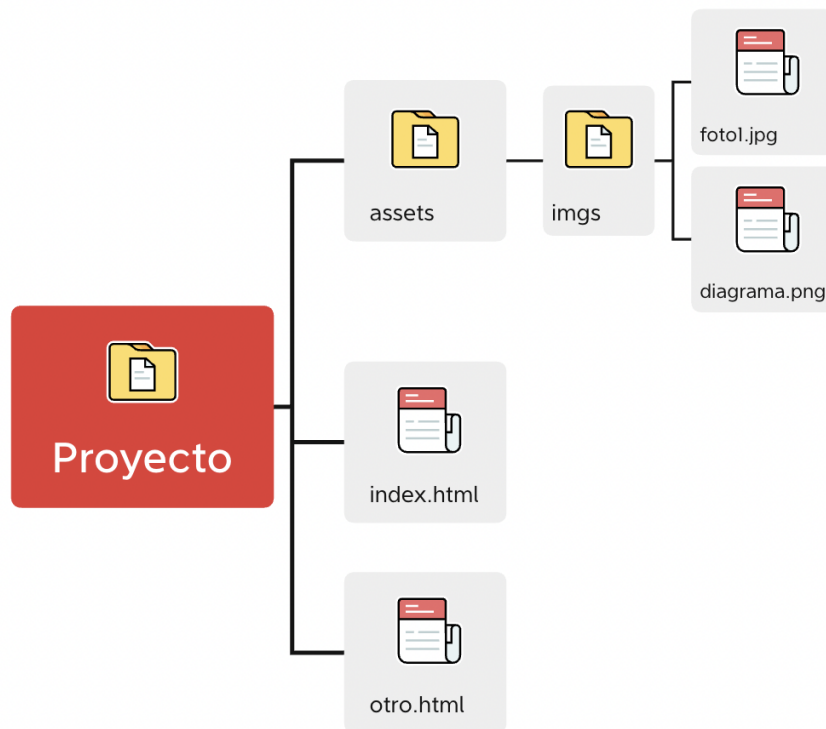


Imagen 2. Estructura de carpetas.
Fuente: Desafío Latam.

La carpeta "assets" corresponde entonces a una convención que indica dónde se deben almacenar los archivos adicionales de nuestro proyecto, como imágenes u otros archivos como hojas de estilo (css) que estudiaremos más adelante. Ahora te toca a ti.



Actividad 1: Estructurando un proyecto web

- Crea un proyecto nuevo llamado cv2 (carpeta).
- Carga el proyecto en vscode.
 - Esto lo puedes lograr arrastrando la carpeta a VSCode cuando está abierta.
- Crea el archivo index.html con la estructura base de un documento html.
- Crea el archivo tutoriales.html con la estructura base de un documento html.
- Crea el archivo tabla.html con la estructura base de un documento html.
- Crea la carpeta assets.
 - Utiliza minúsculas para los directorios.
- Dentro de la carpeta assets crea la carpeta imgs.
 - Utiliza minúsculas para los directorios.
- Guarda alguna foto que tengas en tu computador en esa carpeta, si no tienes alguna puedes descargar una de internet, pero tienes que descargarla.
- Agrega la imagen al documento .html utilizando:
 - ``
- Abre el archivo index.html con el navegador y prueba que puedes ver la foto.



En este ejercicio logramos dos aprendizajes importantes:

1. Que podemos cargar imágenes locales de nuestro computador, incluso cuando están en subcarpetas.
2. Que podemos ordenar nuestro proyecto utilizando una estructura adecuada de carpetas como la que presentamos en este ejercicio y utilizaremos a lo largo del curso.

Profundizando en las etiquetas dentro HEAD

Dentro de la etiqueta HEAD, podemos encontrar varias etiquetas importantes que estudiaremos un poco más adelante, dentro de ellas: el título (<title>), la codificación (<meta charset="utf-8">), el favicon (<link rel="shortcut icon">) y otros metadatos (<meta>).

También podemos encontrar los estilos (<style> o <link rel="stylesheet">), los cuales aprenderemos en la próxima unidad.

Título

La etiqueta <title></title> muestra el título de la página en el tab, además, tiene atribuciones semánticas para el navegador, y posee la siguiente estructura:

```
<title> Academia Desafío Latam - Desafío Latam </title>
```

Y se traduce en el navegador de esta manera:



Imagen 3. Título Web Desafío Latam.
Fuente: Desafío Latam.

Codificación

La etiqueta <meta> con el atributo charset especifica la codificación de los caracteres del documento. En este caso usaremos utf-8 para poder codificar símbolos y caracteres latinos. Posee la siguiente estructura:

```
<meta charset="UTF-8">
```

Esta etiqueta es muy importante, ya que si no la usamos, letras como la "Ñ" o con tilde no se codificarían bien. Algo tan simple como este texto: Ñ á é í ó ú se vería así:

Ã‘ Ã¡ Ã© Ã³ Ãº

Imagen 4. Tildes.
Fuente: Desafío Latam.

Favicon

Es una pequeña imagen asociada al sitio web que se muestra en la pestaña correspondiente. Para poner el pequeño ícono que aparece en la pestaña del navegador se tiene que tener una imagen pequeña (formato jpg, png, gif o bmp) y cuadrada. Posee la siguiente estructura:

```
<link rel="icon" type="image/png" href="/assets/favicon/icono.png">
```

Y se traduce en el navegador de esta manera:

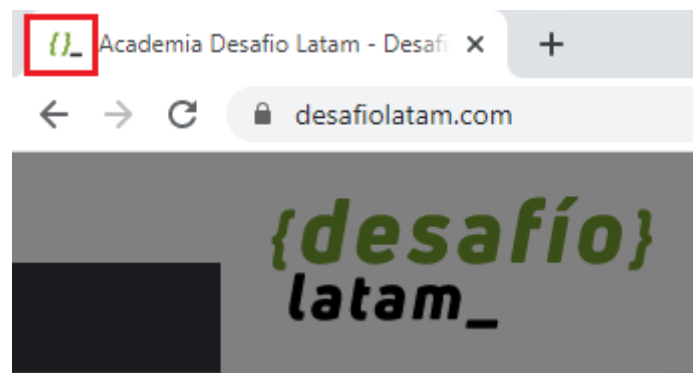


Imagen 5. Favicon Web Desafío Latam.
Fuente: Desafío Latam.



En la página <https://www.favicon-generator.org> puedes generar un favicon a partir de una imagen para tu sitio, o, en la página <https://favicon.io> puedes generar un favicon a partir de un texto. Si buscas en Google por generadores de favicon encontrarás muchas más opciones.



Actividad 2: El head de nuestro nuevo curriculum

Dentro del proyecto ya creado “cv.2”, del archivo index.html y del archivo tutoriales.html, deberás:

- Asegurar que esté el meta charset utf-8.
- Agregar como título Curriculum de {tu nombre}
- Crear un favicon y agregarlo al sitio, asegúrate de que puedas verlo en el navegador.
- Procurar que todas estas etiquetas tienen que estar dentro del head de la página.

Profundizando en etiquetas dentro del body

 Imágenes

La etiqueta img permite agregar una imagen en una página web, esta puede ser un archivo que esté dentro de la carpeta del proyecto (o en una subcarpeta) o puede ser un archivo que se encuentre en otra página web.

Diremos que los archivos serán **locales** cuando estén dentro de la carpeta del proyecto o dentro de una subcarpeta de este.

Cuando los archivos sean locales, utilizaremos alguna de estas dos alternativas:

```

```

```

```



Extensiones ocultas

Considera que los archivos además del nombre tienen una extensión, por ejemplo hola.jpg el nombre es hola y la extensión es .jpg.

En algunos sistemas operativos, la configuración de no mostrar extensión de archivos se encuentra activada, y para verlas, puedes intentar buscar y cambiar la configuración o utilizar el explorador del proyecto del editor de código para ver el nombre completo (nombre + extensión del archivo).

Ahora bien, si una imagen local no carga probablemente es por uno de los siguientes motivos:

- El nombre es incorrecto.
 - Ejemplo: Agregaste como nombre hola23.png en lugar holaaa23.png
- La extensión es incorrecta.
 - Ejemplo: La dirección que agregaste dice "foto.jpg" en lugar de foto.png
- La ruta al archivo es incorrecta.
 - Ejemplo: Agregaste como dirección "carpeta/archivo.png" y el archivo .png está en la carpeta pruebas, en ese caso la dirección será "prueba/archivo.png"
- En algunos sistemas los nombres de los archivos son sensibles a las mayúsculas (Hola.jpg es distinto a hola.jpg") mientras que en otros no, por lo mismo, se recomienda utilizar siempre minúsculas.
- Revisa que no existan espacios en blanco en el nombre del archivo, puedes cambiarlos por un guión.
 - Ejemplo: En lugar de "hola mundo.jpg" utiliza "hola-mundo.jpg"

Atributo alt

El atributo alt de la imagen nos permite mostrar un texto alternativo en caso de que por algún motivo la imagen no pueda ser mostrada. Los desarrolladores junior tienden a ignorar este atributo por no encontrarlo importante, pero es crítico tanto por temas de accesibilidad como por la importancia que tiene para el SEO del sitio.

Ejemplo de una imagen con alt:

```

```

<a> Links

Los links permiten enlazar nuestra página a:

- Otros sitios, diferentes al nuestro;
- Otras páginas de nuestro sitio;
- Archivos;
- Secciones dentro de la misma página;
- Entre otros.

La etiqueta para los enlaces es: `<a>`, y se escribe de la siguiente manera:

```
<a href="link">Texto a mostrar</a>
```

Al interior de `href=""` debe ir la ruta a la cual seremos dirigidos. Y entre los símbolos `><` debería ir el contenido mostrado para hacer clic.

Links a otros sitios

Escribe lo siguiente:

```
<a href="https://www.w3schools.com/">Aprende más sobre desarrollo web en W3Schools</a>
```

Esto creará un link que te redirigirá al sitio enlazado, pero algunas veces vamos a querer que los links se abran en una nueva pestaña, sobre todo cuando la página a la cual redirige es de un sitio distinto al nuestro. Esto se consigue añadiendo otro atributo a la etiqueta `<a>` llamado `target`, y para que siempre se abra en una nueva pestaña, debe llevar el valor `_blank` como se observa a continuación:

```
<a href="https://www.w3schools.com/" target="_blank">Aprende más sobre desarrollo web en W3Schools</a>
```

Cuando utilizamos `target="_blank"` adicionalmente deberíamos incluir `rel="noopener noreferrer"` Para evitar problemas que por ahora no especificaremos

Para abrir un link en un tab nuevo debería quedar así:

```
<a href="https://www.w3schools.com/" target="_blank" rel="noopener noreferrer">Aprende más sobre desarrollo web en W3Schools</a>
```

Links a páginas dentro de nuestro sitio

Para dirigir páginas dentro de un mismo sitio, basta con poner el nombre del archivo HTML al que hacemos referencia. Por ejemplo, si quisieras hacer un enlace desde otra página a tu inicio, escribirías lo siguiente:

```
<a href="index.html">Link a la página principal de nuestro proyecto</a>
```

Recuerda que el valor de `href` es la ruta al archivo.



Existen 2 tipos de rutas: **relativas** y **absolutas**, y su diferencia radica en que las absolutas incluyen la dirección completa desde la raíz o disco duro del sistema operativo, mientras que las relativas apuntan a un archivo a partir de la ubicación del documento en el que se está especificando la dirección. Para los archivos locales (de cualquier tipo) se utilizan rutas relativas.

Imágenes con links

No siempre el contenido mostrado para hacer clic es un texto, a veces puede ser una imagen. A continuación, haremos un link a Google con su logo clickeable.

- Escribe la etiqueta `<a>`, con su atributo `href` dirigiendo a [Google](https://www.google.cl), como es un enlace a un sitio externo le daremos el atributo `target="_blank"`.
- Luego, en el mismo Google busca el logo de Google y se genera la etiqueta `` con sus atributos `src` y `alt`, dentro del contenido de la etiqueta `<a>` como se observa a continuación:

```
<a href="https://www.google.cl" target="_blank"></a>
```

Si das clic en la imagen, te llevará a la que agregaste en el link:



Imagen 6. Enlaces.
Fuente: Desafío Latam.

<table> Tabla

En html podemos agregar tablas, aquí hay varias etiquetas involucradas similar a las listas donde tenemos elementos con . En un tabla tenemos la siguiente estructura:

<table> Comienza tabla

<tr> Comienza una fila

<td> Un elemento </td>

</tr> Termina fila

</table> Termina tabla

Por ejemplo, supongamos que tenemos una tabla con estudiantes y notas.

```
1 <table>
2   <tr>
3     <td>Mai</td>
4     <td>10</td>
5   </tr>
6   <tr>
7     <td>Camila</td>
8     <td>10</td>
9   </tr>
10  <tr>
11    <td>Gonzalo</td>
12    <td>10</td>
13  </tr>
14  <tr>
15    <td>Alex</td>
16    <td>10</td>
17  </tr>
18 </table>
```

| | |
|---------|----|
| Mai | 10 |
| Camila | 10 |
| Gonzalo | 10 |
| Alex | 10 |

Imagen 7. Tabla de estudiantes y notas.

Fuente: Desafío Latam.

En el código anterior cada <tr> marca el inicio de una fila nueva y cada <td> un elemento nuevo dentro de esa fila. También es posible dejar un elemento vacío.

Podemos hacer una tabla un poco más compleja agregando un par de etiquetas nuevas.

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Nota 1</th>
    <th>Nota 2</th>
  </tr>
  <tr>
    <td>Camila</td>
    <td>10</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Gonzalo</td>
    <td>10</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Felipe</td>
    <td></td>
    <td>9</td>
  </tr>
  <tr>
    <td>Javiera</td>
    <td>10</td>
    <td>10</td>
  </tr>
</table>
```

En la tabla anterior, utilizamos **th** en lugar de **td** al principio de la tabla para indicar que estos son elementos de la cabecera de la tabla, lo otro interesante a observar es el caso de felipe, donde hay un elemento vacío.



Actividad 3: Tablas

En el archivo tabla.html realizado en la Actividad 1, crea una tabla igual a la de la siguiente imagen:

| A | B | C |
|---|---|---|
| 1 | 2 | 3 |
| 4 | | 6 |
| 7 | 8 | 9 |

Imagen 8. Actividad 3 tablas.
Fuente: Desafío Latam.



Actividad 4: Investigando sobre thead y tbody

Investiga sobre las etiquetas thead y tbody y rehace la tabla ocupando estas etiquetas. Si todavía te cuesta identificar etiquetas mal cerradas, te recomendamos hacer la actividad desde cero y sin utilizar el autocompletado del editor.



Actividad 5: Revisando los aprendizajes

En esta actividad, vamos a revisar los aprendizajes adquiridos hasta ahora y para ello, realizaremos las siguientes acciones:

- **Paso 1:** Abrir un archivo html con el navegador en tu computador.
- **Paso 2:** Abrir un archivo .html en el editor de código.
- **Paso 3:** Abrir un proyecto (carpeta) en el editor de código.
- **Paso 4:** Crear un archivo index.html con la estructura base de un documento html (head y, body).
- **Paso 5:** Agregar un párrafo.
- **Paso 6:** Agregar cabeceras (h1 .. h5).
- **Paso 7:** Encontrar una etiqueta mal cerrada.
- **Paso 8:** Agregar una lista.
- **Paso 9:** Agregar un link a un archivo html en la misma carpeta.
- **Paso 10:** Agregar una imagen que esté en la misma carpeta que el archivo html.
- **Paso 11:** Agregar una imagen que esté en una subcarpeta o sub-sub-carpeta (como la del proyecto de la actividad).
- **Paso 12:** Agregar un link a una página externa.
- **Paso 13:** Agregar un link a una página externa y que esta se abra en un nuevo tab.
- **Paso 14:** Crear una tabla con varias filas y columnas.

Ahora que terminaste la guía, resuelve el desafío que se encuentra en la plataforma LMS y cuando termines todo recuerda leer sobre SEO y accesibilidad, dos temas muy importantes fuertemente relacionados con el desarrollo web.

¡Continúa aprendiendo y practicando!

