

## Guía de ejercicios 1 - HTML



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



¡Recuerda! La guía es un **complemento** de la sesión sincrónica. Si no fuiste a clases, te recomendamos revisar la grabación antes de revisar la guía.

### ¿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, 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 etiquetas <img> <a> y aprenderemos a crear tablas ocupando HTML.

### ¿Cómo sacarle el máximo provecho a esta guía?

- Los conceptos importantes aparecen **resaltados**, pone atención en estos, anótalos en un cuaderno o documento de apuntes junto a su definición, y también agrega cualquier otra palabra que no conozcas a tu glosario.
- Realiza los ejercicios junto con la guía.
- Realiza todas las actividades.



**¡Vamos con todo!**

## Tabla de contenidos

¿Qué es HTML?	3
Herramientas para crear un sitio web	3
Creando una página web desde cero	4
Introducción a las Etiquetas	6
Anatomía de una etiqueta	8
Etiquetas básicas	8
Comentarios	9
Etiquetas mal cerradas	9
<img> Imágenes	10
Texto alternativo (Atributo alt)	12
<a> Links	12
Agregando un link a una página interna	13
Rutas relativas y rutas absolutas	13
Imágenes con links	15
Estructura base de un documento HTML	16
Estructura de un proyecto web	17
Actividad 1: Estructurando un proyecto web	18
Profundizando en las etiquetas dentro de HEAD	20
Título	20
Codificación	21
Favicon	21
Actividad 2: El head de nuestro nuevo curriculum	22
<table> Tabla	22
Actividad 3: Tablas	24
Actividad 4: Investigando sobre thead y tbody	25
Actividad 5: Revisando los aprendizajes	25



**¡Comencemos!**

## ¿Qué es HTML?

**HTML** es un lenguaje diseñado para crear páginas web, en el cual escribiremos dentro de un archivo que nosotros mismos crearemos utilizando un **editor de código**.

Una vez creado el archivo y agregado un poco de contenido, lo abriremos con el navegador, el cual lo transformará en una página web que podremos leer.



La sigla HTML proviene del inglés **HyperText Markup Language**, y que podemos traducir como un **lenguaje de marcas de hipertexto**. Al final de esta guía estudiaremos qué significa exactamente esto.

## Herramientas para crear un sitio web

Utilizaremos principalmente 2 herramientas:

1. El editor de código VSCode.
2. El navegador Firefox.

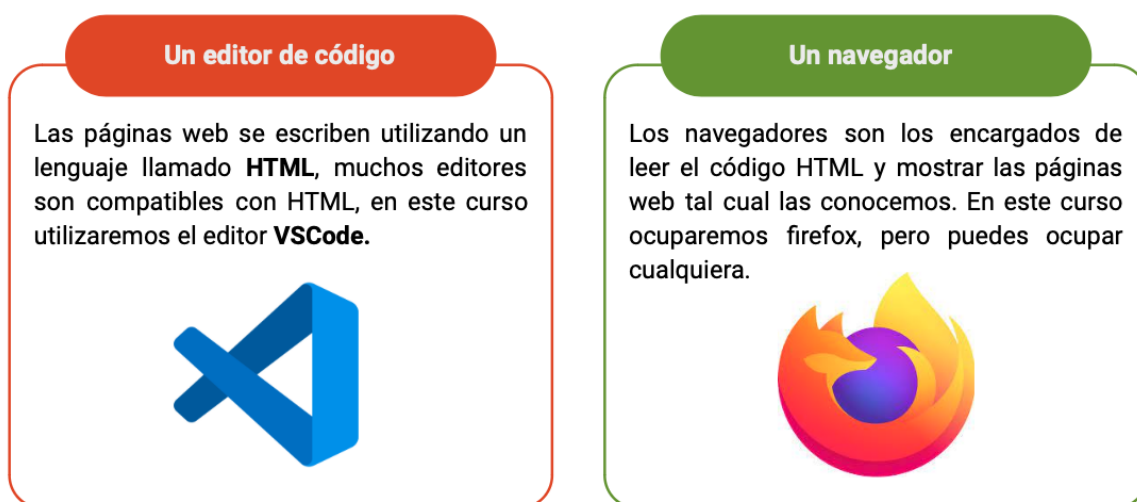


Imagen 1. Herramientas básicas para crear una página web  
Fuente: Desafío Latam

Muy probablemente ya tengas un navegador instalado en tu computador, de todas formas te recomendamos instalar firefox para que puedas seguir el curso con las mismas herramientas que fue creado.

En el caso de Visual Studio Code, te recomendamos instalarlo desde la página oficial <https://code.visualstudio.com/>

## Creando una página web desde cero

Para crear nuestra primera página web, haremos un directorio (nueva carpeta), y luego, abriremos el editor de código arrastrando la carpeta creada a este. Al realizar esta acción, veremos nuestro directorio dentro de la columna de explorador.

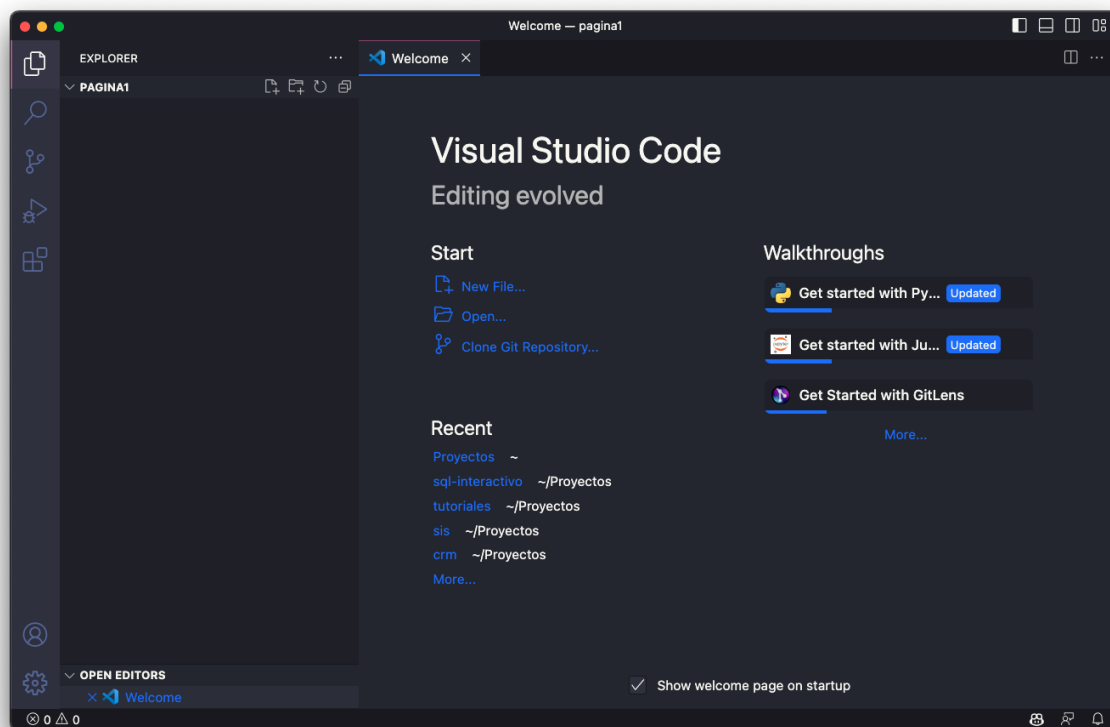


Imagen 2. Creando un proyecto en vscode  
Fuente: Desafío Latam



Una forma común de organizar los espacios de trabajo es mediante la creación de una carpeta por proyecto, donde a la hora de trabajar arrastraremos la carpeta completa del proyecto a VSCode.

A continuación, haciendo clic derecho en el ícono “nuevo archivo” (new file), generaremos un archivo al que llamaremos index.html. Este ícono se encuentra a la derecha de tu proyecto, por lo que debes buscar el símbolo del archivo con un símbolo +.

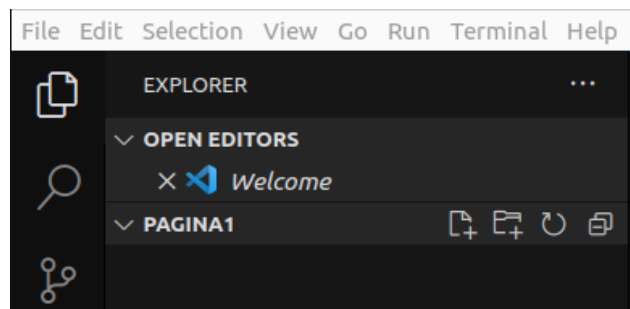


Imagen 3. Creando un archivo dentro del proyecto.  
Fuente: Desafío Latam.

Dentro de este archivo creado, escribiremos el texto “hola mundo” y luego lo guardaremos.



**Importante: ¡Guarda los cambios!**

Un error frecuente consiste en no guardar los cambios. Cada vez que agregamos información o hacemos cambios en la página web utilizando el editor, debemos guardarlos. Puedes ocupar el atajo del teclado `ctrl + s` en windows y linux o `cmd + s` en OS X.

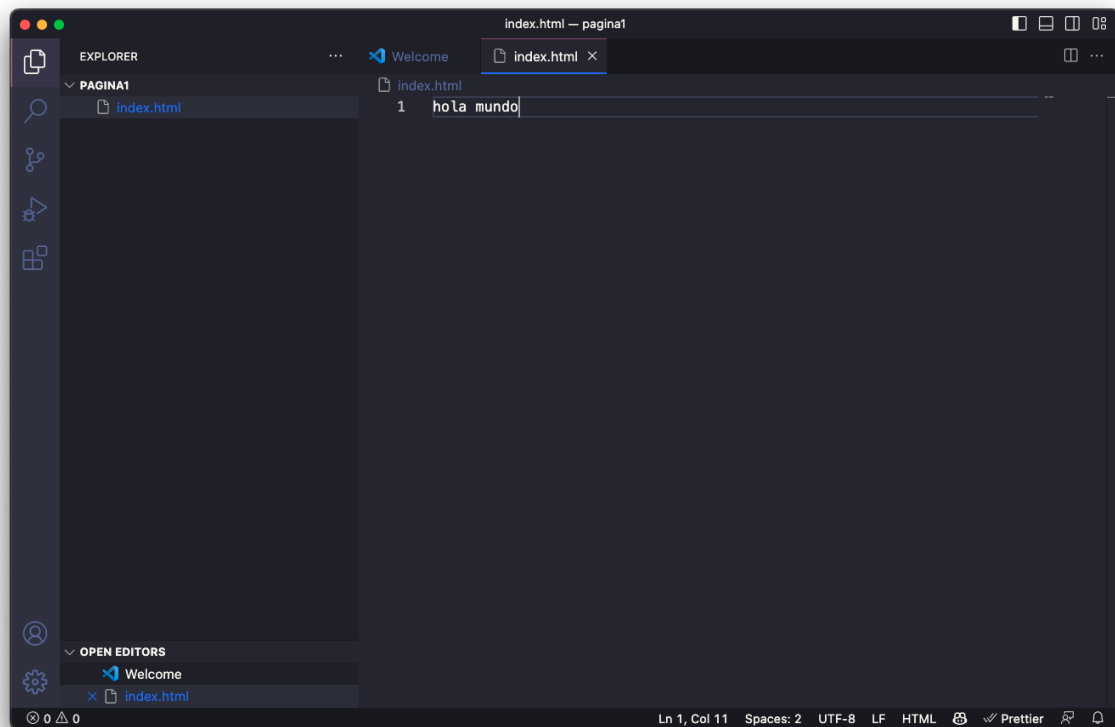


Imagen 4. Creación del archivo index.html  
Fuente: Desafío Latam

Utilizando el explorador de archivos de nuestro sistema operativo, abriremos la carpeta recién creada y haremos doble clic sobre el archivo index, lo que abrirá nuestra página en el navegador. En caso de que esto no funcione, puedes hacer clic derecho y seleccionar la opción “abrir con”, señalando el navegador que prefieras para abrir la página. También es posible abrir el documento arrastrando el archivo creado index.html, sobre el navegador.

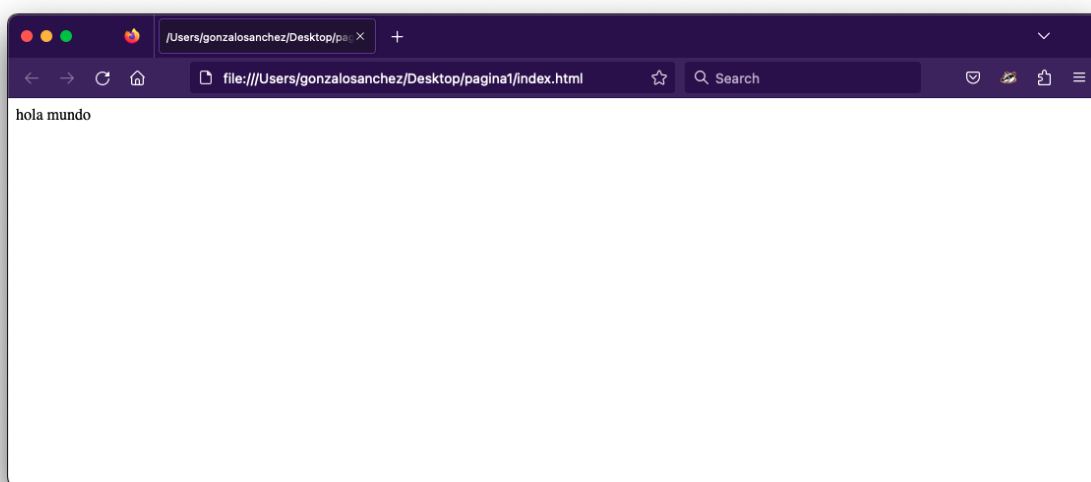


Imagen 5. Nuestra primera página web  
Fuente: Desafío Latam.

Si intentas agregar más contenido, aunque tenga espacios o líneas en blanco entre medio, veremos que esas no se ven en la página web. Esto se debe a que los documentos HTML no son “what you see is what you get” (lo que ves es lo que obtienes), por lo que debemos darle formato al contenido usando etiquetas HTML.



### Refresca la página

Si haces cambios después de abrir la página en el navegador, estos no serán visibles automáticamente, por lo que para poder verlos deberás recargar la página. Puedes hacerlo rápido con el atajo del teclado ctrl + r en windows y linux o cmd + r en OS X.

## Introducción a las Etiquetas

Las **etiquetas** HTML (o en inglés tags) permiten definir la estructura y el formato de los elementos en la página web. Por ejemplo, podemos usar la etiqueta **<p>** para indicar un párrafo, la etiqueta **<h1>** para un encabezado principal, entre otras.

Lo importante es que al utilizar correctamente las etiquetas, podemos lograr que el contenido se vea organizado y fácil de leer para los usuarios.

Un ejemplo del uso de etiqueta <p> es el siguiente:

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

La etiqueta se abre de esta forma <p> y se cierra de esta forma </p>, considera que todo lo que está dentro de la etiqueta es el contenido, en este caso, el texto “hola” queda como un párrafo.

Será más sencillo visualizar el efecto de las etiquetas si agregamos varias como en el siguiente ejemplo:

```
<h1> Encabezado principal </h1>  
<p> Primer párrafo </p>  
<p> Segundo párrafo </p>  
<p> Tercer párrafo </p>
```

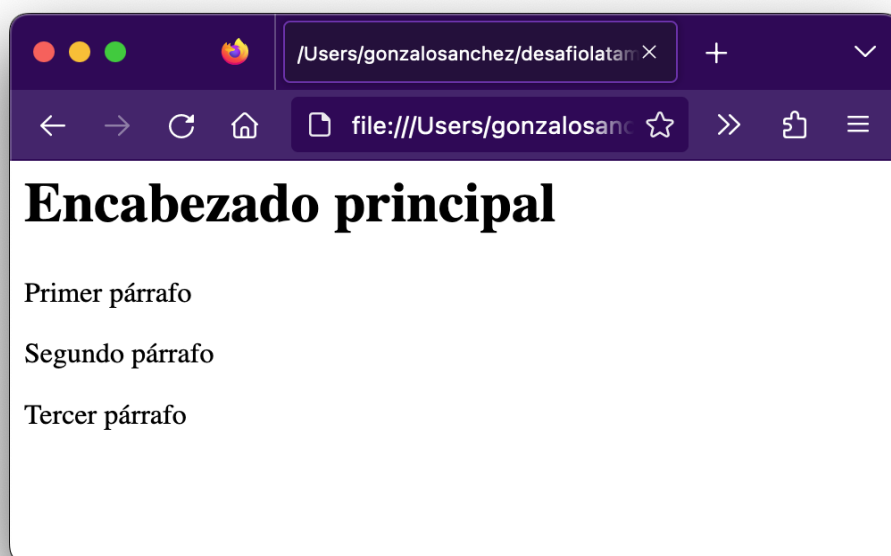


Imagen 6. Etiquetas p y h1  
Fuente: Desafío Latam

Además de las etiquetas para estructurar y formatear el contenido, HTML también nos permite agregar elementos multimedia como imágenes, videos, y audio. Por ejemplo, podemos utilizar la etiqueta <img> para insertar una imagen en la página web, esto lo veremos un poco más adelante en esta misma guía.

## Anatomía de una etiqueta

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".

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

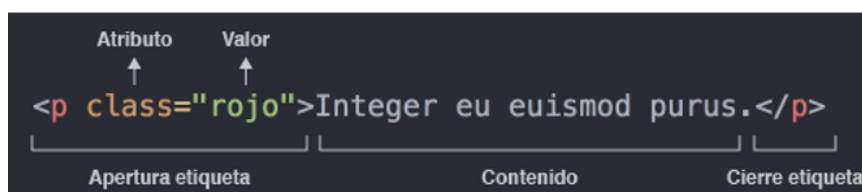


Imagen 7. Sintaxis de etiquetas

Fuente: Desafío Latam.

Utilizando estos atributos y valores podemos insertar imágenes en el caso de la etiqueta `<img>` y podemos hacer links en el caso de la etiqueta `<a>`



En la próxima unidad utilizaremos los atributos `class` y `style` para dar estilo a nuestras páginas web.

## Etiquetas básicas

Etiqueta	Descripción	Ejemplo
<code>&lt;p&gt;</code>	Define un párrafo de texto.	<code>&lt;p&gt;Este es un párrafo de ejemplo.&lt;/p&gt;</code>
<code>&lt;h1&gt;</code> <code>&lt;h6&gt;</code>	Define encabezados de diferentes tamaños, siendo <code>&lt;h1&gt;</code> el más grande.	<code>&lt;h1&gt;Este es un encabezado de nivel 1.&lt;/h1&gt;</code>
<code>&lt;ul&gt;</code>	Define una lista no ordenada.	<code>&lt;ul&gt; &lt;li&gt;Elemento 1&lt;/li&gt; &lt;li&gt;Elemento 2&lt;/li&gt; &lt;li&gt;Elemento 3&lt;/li&gt; &lt;/ul&gt;</code>



<ol>	Define una lista ordenada.	<pre>&lt;ol&gt;   &lt;li&gt;Primer elemento&lt;/li&gt;   &lt;li&gt;Segundo elemento&lt;/li&gt;   &lt;li&gt;Tercer elemento&lt;/li&gt; &lt;/ol&gt;</pre>
<li>	Define un elemento de lista.	Se ocupa en conjunto con <ol> o <ul>
<a>	Define un enlace o vínculo a otra página o archivo.	<pre>&lt;a href="https://www.ejemplo.com"&gt;Este es un enlace de ejemplo&lt;/a&gt;</pre>
<img>	Inserta una imagen en la página.	<pre>&lt;img src="ruta-de-la-imagen.jpg" alt="Texto alternativo de la imagen"&gt;</pre>

Tabla 1. Etiquetas básicas  
Fuente: Desafío Latam

Profundizaremos en las etiquetas <a> e <img> a lo largo de la guía.

## Comentarios

Los comentarios no se ven en la página web y son un código que es ignorado por el navegador.

Estos sirven para hacer anotaciones en una página web, ya sea para indicar algo a otros programadores que participen en el proyecto o incluso para dejar un mensaje a los usuarios curiosos que revisen el código fuente de la página.

```
<!-- Aquí va la información para el navegador -->
```

## Etiquetas mal cerradas

¿Por qué es importante utilizar la etiqueta de cierre en HTML? ¿Qué sucede si se omite la etiqueta de cierre en una etiqueta que la requiere?

Es relevante emplear la etiqueta de cierre en HTML porque indica el final de un elemento o etiqueta. Cuando se omite la etiqueta de cierre en una que la requiere, el navegador no sabrá dónde termina el elemento y puede haber problemas en la visualización de la página.



Utilizando los conocimientos adquiridos recientemente, ¿podrías identificar cuál es el problema en el siguiente código?

```
<ul>
  <li> Elemento 1 </ul>
</li>
```

Este código es incorrecto porque las etiquetas `<ul>` y `<li>` están anidadas de manera incorrecta. La etiqueta `<ul>` se utiliza para indicar el inicio de una lista no ordenada, mientras que la etiqueta `<li>` se utiliza para definir cada uno de los elementos de la lista.

En este ejemplo específico, la etiqueta de cierre `</ul>` aparece antes de la etiqueta de cierre `</li>`, lo cual provoca que el navegador no sepa cómo interpretar correctamente el contenido. Como resultado, es probable que la lista no se muestre correctamente en la página web.

Para corregir el código, simplemente debemos asegurarnos de anidar correctamente las etiquetas. En este caso, la etiqueta de cierre `</li>` debería aparecer antes de la etiqueta de cierre `</ul>`, como se muestra en el siguiente ejemplo:

```
<ul>
  <li>Elemento 1</li>
</ul>
```

Es importante recordar que la correcta anidación de etiquetas es esencial para crear un código HTML válido y que funcione correctamente.

## <img> Imágenes

La etiqueta `img` permite agregar archivos de imagen en una página web. Estos archivos pueden encontrarse dentro de la carpeta (o subcarpeta) del proyecto o pueden ser archivos que se encuentren en la web.

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

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

```

```

```

```



### Extensiones ocultas

Además del nombre, considera que los archivos 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. Para ver el nombre completo, puedes intentar cambiar la configuración o utilizar el explorador del proyecto en el editor de código (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 prueba. En ese caso, la dirección correcta 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.
- Existen espacios en blanco en el nombre del archivo, cuando esto ocurre, puedes cambiarlos por un guión.
  - Ejemplo: En lugar de "hola mundo.jpg" utiliza "hola-mundo.jpg"

También podemos agregar imágenes directamente desde la web copiando su dirección. Cuando encontremos una imagen que nos guste, utilizaremos el clic derecho sobre la imagen y seleccionaremos "copiar dirección de imagen" o "copiar URL de imagen". Luego, agregaremos la dirección URL entre las comillas del atributo src de la etiqueta img.

Por ejemplo, si deseamos mostrar la imagen de un gato, podemos buscarla en un motor de búsqueda y copiar su dirección URL. Luego, podemos utilizar la siguiente etiqueta img para incluir la imagen en nuestra página:

```

```

Al recargar la página, podremos ver la imagen del gato. Es importante recordar que, al igual que con cualquier otro recurso externo, debemos asegurarnos de tener permiso para utilizar la imagen y respetar los derechos de autor correspondientes.

### *Texto alternativo (Atributo alt)*

La etiqueta img tiene un atributo **alt** para especificar un texto alternativo. Este se utiliza para describir una imagen en caso de que no se pueda mostrar correctamente en la página web.

Si el archivo de imagen no se encuentra en la ubicación especificada en el atributo "src", o si el navegador del usuario tiene desactivada la carga de imágenes, la descripción proporcionada en el atributo alt permitirá al usuario comprender el contenido de la imagen.

Además, la descripción proporcionada en el atributo alt es empleado por los lectores de pantalla utilizados por personas con discapacidad visual para describir la imagen. En este caso, el texto alternativo debe ser lo suficientemente descriptivo para que la persona pueda entender el contenido de la imagen sin necesidad de verla.

- Ejemplo de una imagen con alt:

```

```

## **<a> Links**

La etiqueta se utiliza para crear links (también conocidos como **hipervínculos** o enlaces) a otras páginas web, archivos o ubicaciones dentro de la misma página web.

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 `<a>` tiene un atributo **href** que especifica la dirección de la página web o archivo al que se debe enlazar.

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

## Agregando un link a una página interna

Llamamos informalmente página interna a cualquier otra página web dentro del mismo proyecto.

Para crear una segunda página web, que llamaremos `nosotros.html`, desde el editor de código iremos al ícono de archivo nuevo, crearemos el archivo, agregaremos algún texto genérico y luego guardaremos la página. Es importante que esta página esté en la misma carpeta que la página `index.html`

Para agregar un link a la página nueva desde `index.html` agregaremos el siguiente código:

```
<a href="nosotros.html"> Nosotros </a>
```

Y en la página `nosotros.html` agregaremos un link a `index.html`

```
<a href="index.html"> Inicio </a>
```

## Logro desbloqueado



Ahora que ya sabes agregar más páginas web, puedes construir sitios web. Este consta de varias páginas web que están relacionadas entre sí y que se pueden navegar mediante enlaces o menús.

### *Rutas relativas y rutas absolutas*

En HTML (y en informática en general), existen dos tipos de rutas para especificar la ubicación de un archivo, cómo una imagen o una página web: las **rutas relativas** y las **rutas absolutas**.

#### **Rutas absolutas**

- Cuando se utilizan rutas absolutas, se especifica la ubicación completa del archivo o recurso, incluyendo el protocolo de red (`http`, `https`, `ftp`, etc.), el nombre del servidor y la ruta completa del archivo o recurso.

- Son útiles cuando se trabaja con recursos que se encuentran en un servidor externo o cuando se quiere asegurar que la ruta sea siempre la misma, independientemente del contexto de la página.

### Rutas relativas

- Cuando se utilizan rutas relativas, se especifica la ubicación del archivo o recurso en relación con la ubicación del archivo HTML que lo está utilizando.
- Son útiles cuando se trabaja con recursos que se encuentran en el mismo proyecto, por ejemplo imágenes en la misma carpeta
- Una ruta relativa se especifica en relación a la ubicación actual del archivo HTML que está utilizando. Por ejemplo, si un archivo HTML y una imagen se encuentran en el mismo directorio, podemos utilizar una ruta relativa para especificar la ubicación de la imagen, de la siguiente manera:

```

```

En este caso, la ruta es relativa al archivo HTML que está utilizando la imagen, y asume que esta se encuentra en el mismo directorio.

Por otro lado, una ruta absoluta especifica la ubicación completa de un archivo en el sistema de archivos, desde la raíz del sistema. Por ejemplo:

```

```

o

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

En este caso, la ruta es absoluta, especifica la ubicación completa de la imagen o de una página web.

La principal diferencia entre rutas relativas y rutas absolutas es que las rutas relativas son más flexibles y portátiles, ya que no dependen de la ubicación exacta del archivo en el sistema de archivos. Por lo tanto, si se mueve el archivo HTML o la imagen a otro directorio, la ruta relativa seguirá funcionando correctamente, mientras que la ruta absoluta deberá ser actualizada.

En resumen, la elección entre una ruta relativa y una ruta absoluta depende del contexto y de la ubicación de los archivos en el sistema de archivos. En general, se recomienda utilizar rutas relativas para los archivos del proyecto, ya sean páginas web o imágenes.

## 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"`.
- A continuación, busca en Google el logo del buscador, y copia la dirección de la imagen.
- Con esto, genera la etiqueta `<img>` con sus atributos `src` y `alt`, y colócala dentro del contenido de la etiqueta `<a>` tal como se muestra a continuación:

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

Si das clic en la imagen, te llevará a la página indicada en el link:

file:///C:/Users/.../Desktop/mentificofee/index.html  
Academia más robusta desarrollo web en W3Schools  
Academia más robusta desarrollo web en W3Schools  
Academia más robusta desarrollo web en W3Schools

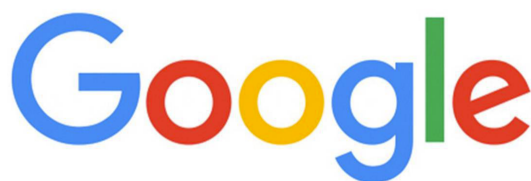


Imagen 8. Enlaces  
Fuente: Desafío Latam

## Estructura base de un documento HTML

Hasta ahora hemos ingresado etiquetas HTML sin respetar la estructura de un documento HTML, por suerte, los navegadores son inteligentes y pueden hacer suposiciones que a veces llevan a buenos resultados, pero un buen documento HTML debe tener una estructura base.

Esta estructura 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).

Cuando abres una página en tu navegador, este necesita saber qué tipo de documento estás intentando ver. Para ello, el navegador revisa la primera línea del archivo y, en este caso, ha encontrado que se trata de un documento HTML5.

Luego, la etiqueta `<html>` especifica que todo lo que está dentro de ella deberá ser interpretado como HTML. A continuación, el `<head>` contiene toda la información que es para el navegador y el `<body>` contiene todo el contenido que es para el usuario.

Es en la etiqueta `<body>` donde agregaremos todas las etiquetas que aprendimos previamente.

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



```
<h1> Primera página web </h1>
<p> Bienvenido a mi página </p>
<ol>
  <li><a href="index.html"> Inicio </a></li>
  <li><a href="nosotros.html"> Nosotros </a></li>
  <li><a href="https://desafiolatam.com"> Vamos con todo </a></li>
  <li><a href="https://www.google.com"> Ir a Google </a></li>
</ol>
</body>
</html>
```



En VSCode podemos utilizar el autocompletado para generar automáticamente la estructura base de un sitio web. Si el archivo tiene extensión .html y el editor lo identifica, al escribir ! (signo de exclamación) y luego tab o enter, automáticamente se autocompletará la siguiente estructura.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
</body>
</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 conserva 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:

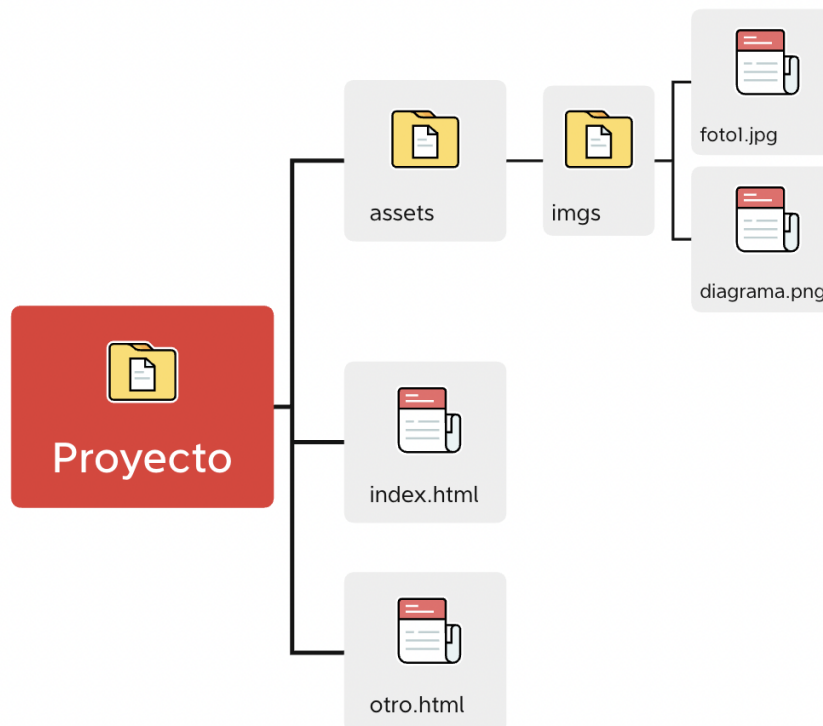


Imagen 9. Estructura de carpetas.  
Fuente: Desafío Latam.

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.
  - Si aún no descargas VSCode puedes hacerlo desde su [página de descarga](#).
- 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 el nombre de los directorios.

- Dentro de la carpeta assets, crea la carpeta imgs.
  - Utiliza minúsculas para el nombre de los directorios.
- Guarda alguna foto que tengas en tu computador en esa carpeta, si no tienes alguna, puedes descargar una de internet.
- Agrega la imagen al documento index.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 sub carpetas.
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 de HEAD

La etiqueta head nos permite especificar información para el navegador, como por ejemplo, el título o codificación del documento.

Las más utilizadas son las siguientes:

Etiqueta	Descripción
<title>	Define el título del documento.
<meta>	Define metadatos del documento, como la descripción, palabras clave y codificación de caracteres.
<link>	Define relaciones entre el documento actual y otros recursos, como hojas de estilo CSS e iconos de página.
<script>	Define un script o programa que se ejecutará en el documento, como JavaScript.
<style>	Define reglas de estilo para el documento, como las propiedades de las fuentes y los colores.

Tabla 2. Etiquetas Head  
Fuente: Desafío Latam

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">)



### ¡Spoiler!

Las etiquetas link y style las estudiaremos en la próxima unidad, la etiqueta script la abordaremos en el módulo de javascript

## Título

La etiqueta <title></title> muestra el título de la página en la pestaña del navegador, y tiene atribuciones semánticas. Su estructura es la siguiente:

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

Se puede visualizar en las pestañas del navegador:



Imagen 10. 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 la tilde no se codificarán bien. Algo tan simple como este texto: Ñ á é í ó ú se vería así:

Ã‘ Ã¡ Ã© Ã© Ã³ Ã³

Imagen 11. Tildes  
Fuente: Desafío Latam

## Favicon

Es una pequeña imagen asociada al sitio web que se muestra en la pestaña correspondiente del navegador. Para poner el pequeño ícono es necesario 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">
```

Se puede visualizar en las pestañas del navegador:

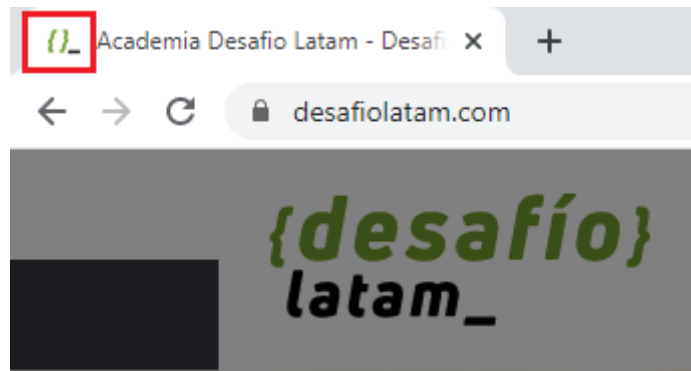


Imagen 12. Favicon Web Desafío Latam  
Fuente: Desafío Latam



En la página <https://www.favicon-generator.org> puedes generar un favicon para tu sitio a partir de una imagen, y 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 “cv2”, en el archivo index.html y el 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 se encuentren dentro del head de la página.

### <table> Tabla

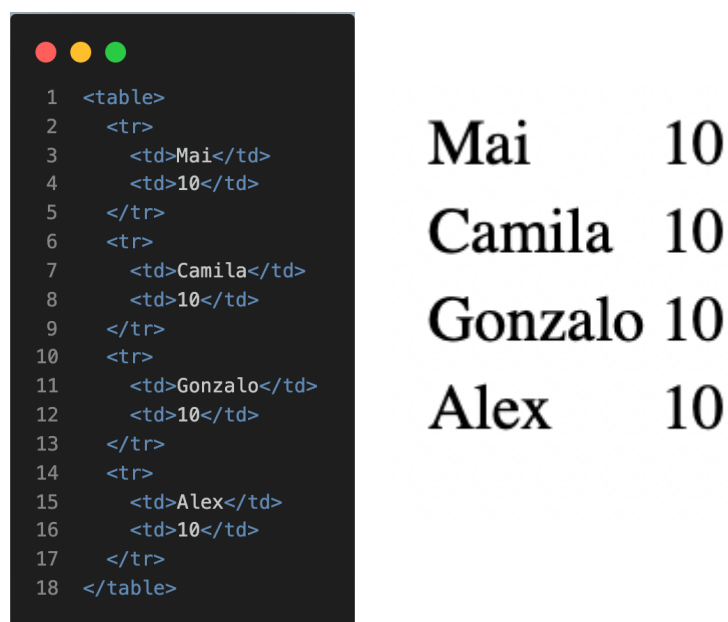
En HTML podemos agregar tablas. En estas hay varias etiquetas involucradas similar a las listas <ul> donde tenemos elementos con <li>.

En una tabla tenemos la siguiente estructura:

```
<table> Comienza tabla
  <tr> <!-- Comienza una fila -->
    <td> Primer elemento de la fila </td>
    <td> Segundo elemento de la fila </td>
  </tr> <!-- Termina una fila -->
</table> Termina tabla
```

El contenido va dentro de las etiquetas <td>

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 13. 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.



### 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 14. 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:** Agregar un link a una página interna del proyecto.
- **Paso 15:** 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 y fuertemente relacionados con el desarrollo web.

**¡Continúa aprendiendo y practicando!**

