



TÉCNICAS DE PROGRAMACIÓN INTERFAZ GRÁFICA (HTML)

Interfaz Gráfica - HTML





HTML - ¿Qué es?

HTML es el lenguaje más usado para construcción de páginas Web. La sigla **HTML** significa **Hyper Text Markup Language** (lenguaje de marcas de hipertexto).

HTML no es un lenguaje de programación, sino de marcas con sentido.

Llamamos a dichas marcas etiquetas. Éstas le indican al navegador qué es lo que debe mostrar en pantalla.



HTML5 - ¿Cuál es el origen?

El lenguaje HTML5 se desarrolló con la cooperación de dos entidades, World Wide Web Consortium (W3C) y Web Hypertext Application Technology Working Group (WHATWG).

WHATWG estaba trabajando en mejoras para los formularios en la web, mientras que W3C trabajaba en XHTML2.0. En el año 2006 decidieron juntarse a trabajar en una nueva y mejor versión del lenguaje HTML, dando así origen a la era de HTML5.



Reglas básicas de HTML5

- Las nuevas funcionalidades se basan en HTML, CSS, DOM, y JavaScript.
- Reduce el uso de plugins externos, por ejemplo: Flash Player.
- Mejora el manejo de errores.
- Tiene mejores etiquetas de marcación.
- HTML5 es independiente del dispositivo de visualización.
- El proceso de desarrollo es visible para el público



¿Qué incorpora HTML5 en su última versión?

- Incorpora **nuevas etiquetas** (canvas 2D y 3D, audio, video).
- Incluye etiquetas para **manejar grandes conjuntos de datos**, como Datagrid, Details, menú y Command, que permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Incorpora **mejoras en los formularios**: nuevos tipos de datos (eMail, number, url, datetime, etc.) y facilidades para validar el contenido sin JavaScript.
- Permite el **uso de visores**: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales).
- **Incluye Drag&Drop**: nueva funcionalidad para arrastrar objetos como imágenes.



¿Qué incorpora HTML5 en su última versión?

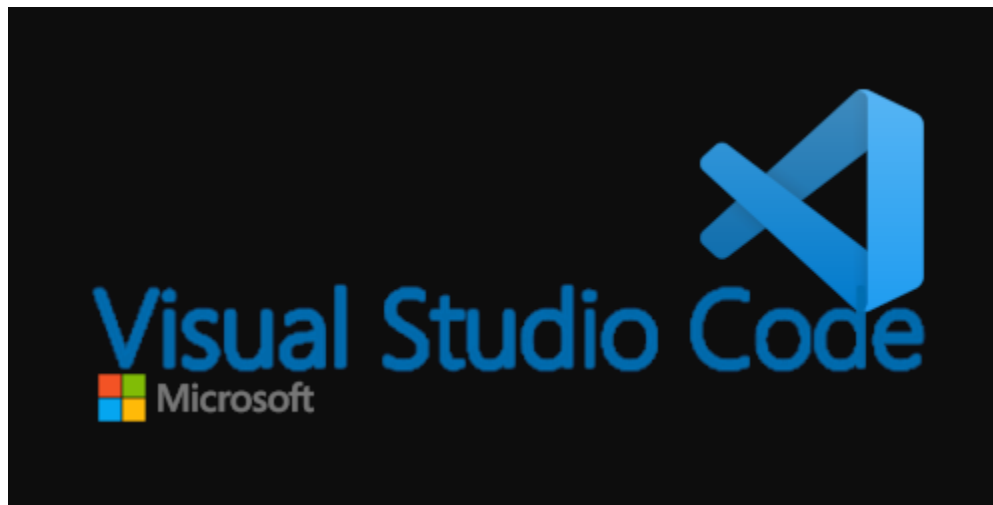
En definitiva, HTML5 incluye nuevos elementos y API de JavaScript para mejorar el almacenamiento, la multimedia y el acceso al hardware.

Fuente: [HTML5 - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web](#)





Introducción a Visual Studio Code





¿Qué es un editor de texto?

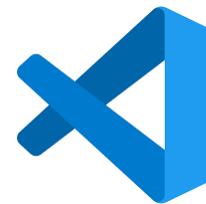
Antes de empezar a trabajar con HTML5, deberás conocer y descargar un editor de texto.

Los editores son programas que permiten editar o crear un texto sin formato. Son especialmente útiles para programar, ya que el código fuente de los programas está hecho en texto plano.

Nota: cualquier editor, incluso el Notepad de Windows o el bloc de Notas de MacOS, sirve para trabajar.

Si bien existen muchos tipos de editores, los de uso más frecuente son:

- [Brackets](#)
- [Sublime Text](#)
- [Visual Studio Code](#)





Editores online

¿Qué quiere decir *online*? Que no los tienes que descargar, los usas desde tu navegador, siempre que tengas conexión a Internet.

Algunos editores de este estilo son:

- [Plunker](#)
- [CodePen](#)



Cualquier editor que uses va a estar bien, siempre que te sea cómodo para trabajar.

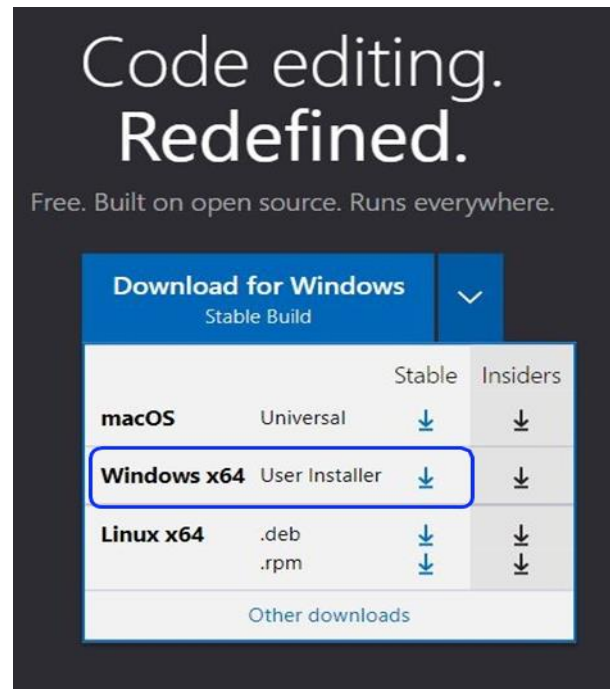
También **puedes utilizar GitHub** para crear un **documento HTML desde cero**, utilizando su vista *Editor*.



Visual Studio Code

En esta etapa trabajaremos con **Visual Studio Code**. Se puedes descargar desde su web oficial: [Visual Studio Code](https://code.visualstudio.com/).

Al elegir qué opción descargar hay que considerar el sistema operativo. Por ejemplo, si utilizas Windows, la opción será la que vemos en la imagen de la derecha.





Ventajas de trabajar con Visual Studio Code

- Autocompletado y sugerencias de código
- Extensiones útiles para el desarrollo
- Depuración integrada para JS
- Integración con GIT
- Vista previa en tiempo real
- Soporte para múltiples lenguajes y frameworks
- Herramientas de inspección y validación
- Terminal integrada
- Personalización con temas y atajos de teclado
- Documentación y referencias rápidas.



Trabajo con la vista Carpeta de Visual Studio Code

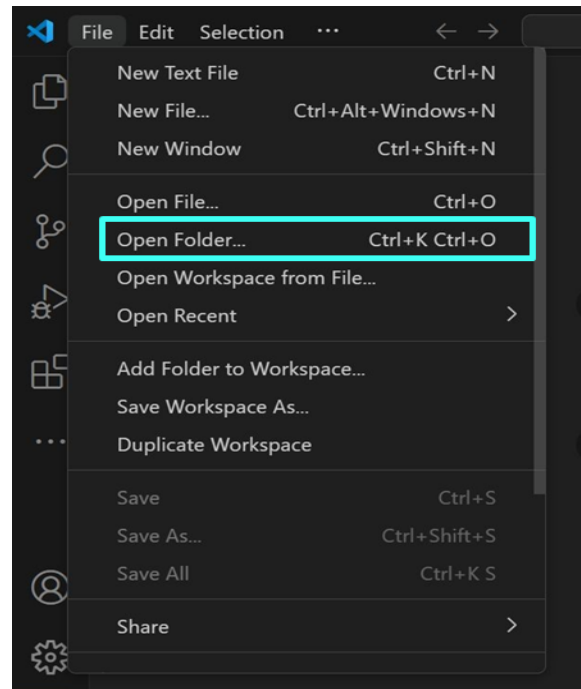
La **vista Workspace** es una excelente forma de trabajo, pero dada su envergadura y finalidad asociada con proyectos más grandes, no será utilizada en esta etapa para no complejizar sin sentido el aprendizaje.

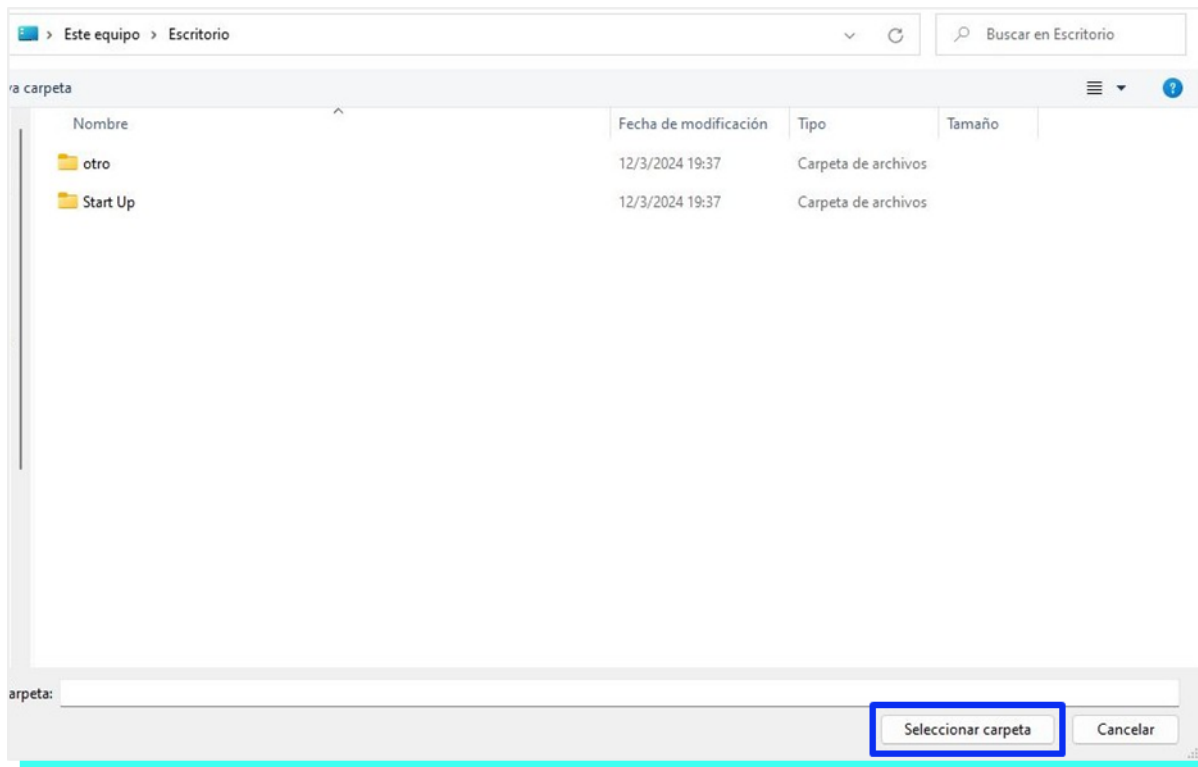
En cambio, se utilizará la **vista Carpeta** o **View Folder** permite implementar un trabajo ágil y organizado con un margen de error mínimo. Veamos cómo activarla, en el siguiente slide.



Pasos a seguir para habilitar la vista Carpeta

1. Ir al menú **Archivo / File** > **Abrir carpeta / Open folder**. **Visual Studio** te permitirá trabajar con una carpeta existente o crear una carpeta desde cero.
2. Luego de **ingresar a la carpeta** deseada, hacer clic en el botón **Seleccionar carpeta**, como se muestra en la siguiente pantalla.

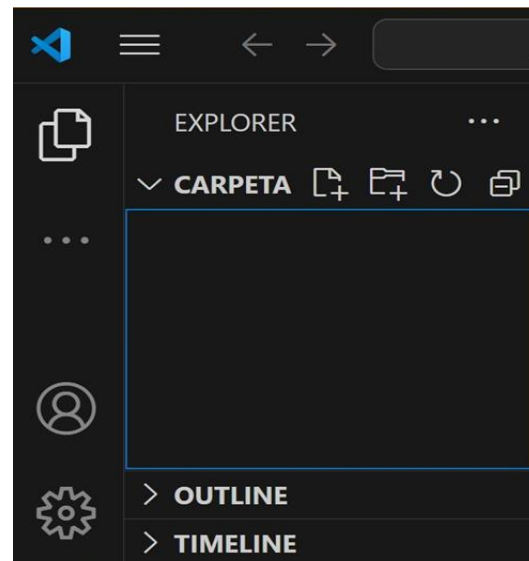






3. Finalmente, *Visual Studio* abrirá el **panel** *Carpeta* para trabajar.

Esto te ayudará a **ordenar el trabajo** a futuro y evitará problemas con rutas u olvidos innecesarios. Otra ventaja de esta vista es que ofrece una serie de íconos que **agiliza el trabajo**, y permite arrastrar, borrar y cambiar el nombre de los archivos fácilmente.



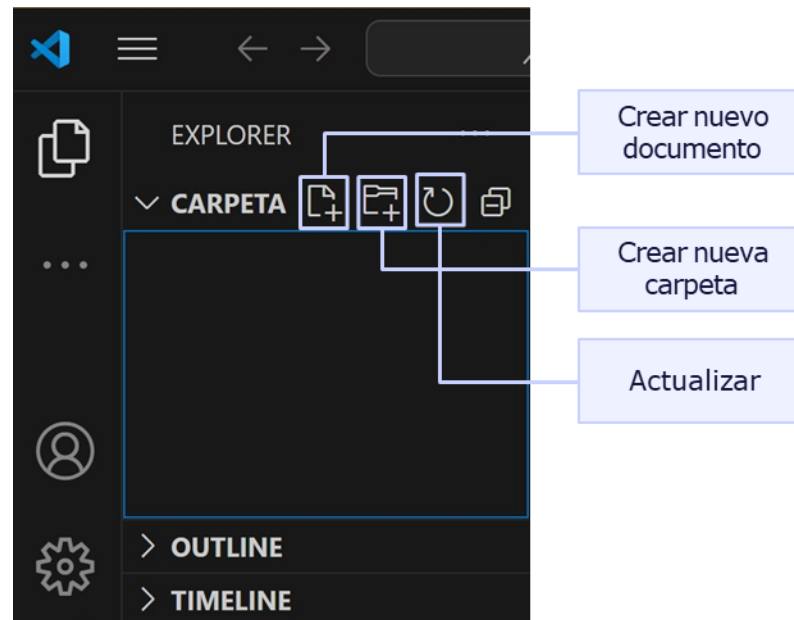


Creación de archivos

Ya instalaste Visual Studio y habilitaste la vista *Carpeta*. Desde allí, puedes utilizar el ícono de la hoja con el más “+” para **crear archivos nuevos**. (De igual modo, el ícono con la carpeta y el más “+”, permite crear carpetas nuevas).

Si realizaste cambios por fuera de *Visual Studio*, y deseas que se actualicen en la vista *Carpeta*, solo debes hacer clic en el tercer ícono (*Refresh*).

Desde *Visual Studio* (o en cualquier otro editor), podrás crear todo tipo de archivos, no solo HTML, sino JavaScript, PHP o Python, entre otros.

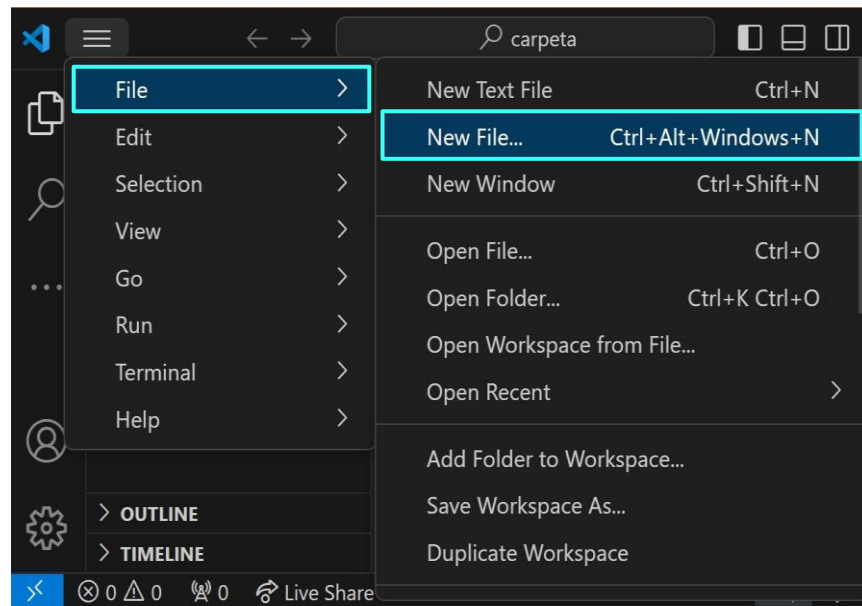




Guardado de archivos

Existe otro modo de crear **archivos**, desde el menú **File > New file (Nuevo archivo)**. Si se crean archivos de este modo y no desde el panel *Carpeta*, *Visual Studio Code* preguntará cómo nombrarlos y dónde.

El nuevo archivo deberá estar siempre dentro de una carpeta. No se subirá al servidor *la carpeta*, sino su contenido, es decir, los archivos dentro de ella. Por eso, no es importante el nombre de la carpeta en sí misma sino los archivos que se guardaran dentro.





Guardado de archivos

Como se mencionó, al crear el **nuevo archivo**, Visual Studio Code ofrecerá la opción de **Guardar**.

En este caso, crearás el primer archivo de tu web, con el nombre ***index.html*** (explicaremos el porqué más adelante).

En *Windows*, se vería así:

Nombre:

Tipo:

^ Ocultar carpetas

Guardar Cancelar



Guardado de archivos

Los nombres de archivos deben respetar las siguientes reglas de nomenclatura:

- Utilizar **nombres cortos y referenciales**.
- No usar **puntos**. Los puntos son **solo para separar el nombre de la extensión**.
- No usar **tildes o acentos**.
- No usar **caracteres "extraños"**. Por ejemplo: @, ñ, etc.
- No usar **espacios para separar las palabras**.
- Se puede usar el - (**guión**) o _ (**guión bajo**).



Comenzando a trabajar con HTML





Index.html

¿Que es el *index*?

El primer archivo siempre se llamará *index*. Esto indica que es el archivo que levantará el servidor **al ingresar al nombre de dominio**.

Por ejemplo, si escribes en la barra de dirección: **centro8smata.org.ar**

Lo que se muestra es el index de la página del Centro 8.

En este caso, se tuvo que haber generado un dominio con ese nombre. Los nombres de dominio son únicos y pueden tener diversas extensiones.

Se puede tramitar un dominio regional desde el sitio www.nic.ar.

Interfaz Gráfica - HTML



Declaración del tipo de documento

Al observar un documento HTML por primera vez, encontraremos una declaración llamada doctype.

A diferencia de las versiones anteriores de HTML, donde existía más de un tipo de doctype, **HTML5 cuenta con un solo tipo** de doctype:

- `<!DOCTYPE HTML>`
- `<!doctype html>`
- `<!doctype HTML>`

Las tres variantes anteriores son correctas, ya que en la versión 5 **las etiquetas no son case sensitive**, es decir que pueden estar tanto en mayúsculas como minúsculas.

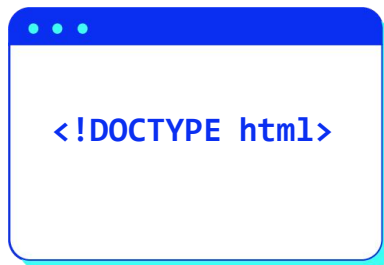


Declaración del tipo de documento

¿Para qué se utiliza la etiqueta *doctype*?

Un **DTD** (*Document Type Definition*) es un conjunto de **normas y restricciones que deben cumplir los documentos de un determinado tipo.**

Por ejemplo, si se define un *DTD* para los documentos relacionados con libros, se puede fijar como norma que cada libro tenga un único título, que tenga uno o más autores o que la información sobre el número de páginas pueda ser opcional.





Sintaxis básica de HTML

Etiquetas

HTML trabaja con etiquetas que, en su mayoría, tienen apertura y cierre:

- La etiqueta de apertura `<` marca el comienzo de una instrucción.
- La etiqueta de cierre `>` marca el fin de la instrucción.

La principal ventaja de los lenguajes de etiquetas es que **son muy sencillos de leer y escribir** para personas y sistemas electrónicos.

Las etiquetas que tienen apertura y cierre, encierran contenidos, como se ve en la figura.



También existen las llamadas “**etiquetas vacías**”, que no encierran ningún contenido y **no requieren etiquetas de cierre**.

Interfaz Gráfica - HTML



Sintaxis básica de HTML

Atributos

Las etiquetas en HTML tienen **atributos**, que **brindan valores o información adicional para configurar los elementos** o ajustar su comportamiento de alguna manera.

Por ejemplo, el atributo **lang**, encontrado en la etiqueta `<html>`, **define el idioma del contenido de un elemento**.

```
<html lang="es">
```

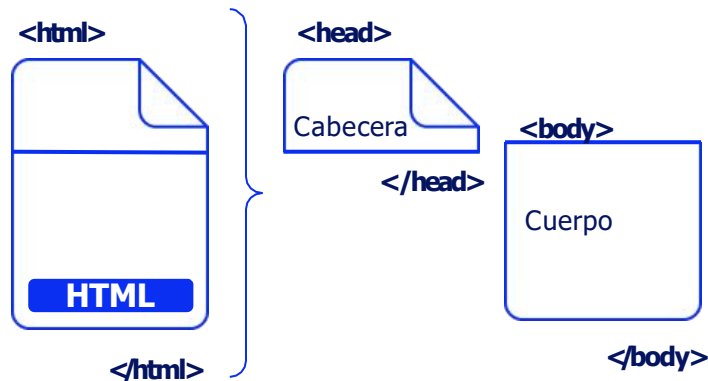


Estructura básica de un HTML

Todo documento HTML se divide en dos partes principales: el encabezado y el cuerpo.

- **El encabezado, o <head>**, contiene información sobre el propio documento HTML, que **no es visible para el usuario**, a excepción de la etiqueta <title>.
- **El cuerpo, o <body>**, contiene en su interior toda la información **visible para el usuario**.

Estructura básica de un HTML





Etiqueta head

Dentro del encabezado, encontraremos etiquetas como <title>, o <meta>, que dan al navegador información sobre el documento. Veámoslo:

Etiqueta title

- Se utiliza para indicar el **título del documento**, que se muestra en la parte superior izquierda de la ventana de los navegadores.
- En los navegadores modernos también se muestra como **nombre de la solapa**.

Veamos un ejemplo:

```
<head>  
    <title> Título de la página </title>  
</head>
```



Etiqueta *meta*

El atributo **charset** del elemento `<meta>` indica la codificación en la cual está escrito el documento HTML.

Atributo *viewport*

Si bien durante esta introducción a HTML no se trabajará de forma avanzada con CSS, es importante entender que **meta viewport permite detectar el tamaño de pantalla desde el cual se visualiza la interfaz**. Esto servirá para tomar las órdenes que le enviaremos desde CSS y, de ese modo, **realizar una interfaz correctamente responsiva**.

viewport, entonces, **hace referencia al área visible de nuestra interfaz**.

Por eso, desde el inicio debemos asegurarnos que este **meta** se encuentre presente.

Por ejemplo:

```
<meta name="viewport"  
content="width=device-width,  
initial-scale=1.0">
```



Etiqueta body

Marca el **cuerpo del documento HTML**. El cuerpo **encierra todos los contenidos que se muestran al usuario** (párrafos de texto, imágenes, tablas).

La cantidad y selección de etiquetas contenidas en dentro del **<body>** depende de la información y el diseño de la página web.

En general, el **<body>** de un documento contiene cientos de etiquetas HTML, mientras que el **<head>** no contiene más que unas pocas.



Estructura general de un documento HTML

Las etiquetas antes mencionadas se organizarán en el HTML de la siguiente forma:

```
<> index.html X
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
12
13
```



Extensión Emmet

Una de las ventajas de trabajar con Visual Studio es que se encuentra integrada a Emmet.

Esta extensión permite **trabajar de forma ágil a través de *snippets*, o ayudas de escritura.**

Durante esta cursada se enseñarán algunos de estos *snippets*, pero se sugiere visitar el [sitio oficial de Emmet](#) para conocer más a fondo las posibilidades que brinda esta extensión.



Implementación de Emmet en Visual Studio

Abre el archivo html en Visual Studio.

Escribe un signo de admiración de cierre “!”.

Aparecerá un desplegable con opciones, como se muestra en la siguiente imagen:





Implementación de Emmet en Visual Studio

3. Al presionar en la primera opción, se generará una estructura básica de manera automática. El .html se verá similar a éste:

```
EXPLORER  ...  Welcome  index.html X
└─ CARPETA
   └─ index.html

index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```



Visualización - Navegadores

Son programas que permiten **interpretar HTML**.

Existen múltiples navegadores, por ejemplo:

- [Google Chrome](#)
- [Mozilla Firefox](#)
- [Opera](#)





Visualizar el HTML desde el navegador

Una de las maneras es **hacerle doble clic al archivo**, luego de guardarlo en el editor de texto.

En ese caso, siempre se abrirá el navegador que tenemos configurado como predeterminado. Por ejemplo, si tu navegador predeterminado es Opera, ahí es donde se abrirá tu archivo HTML.

Otra opción es **copiar y pegar la ruta del archivo en la barra de direcciones** del navegador.

C:\Users\usuario\Desktop\ejemplo.html