



Front-End JS

Clase 01 - “ Conceptos Básicos de HTML”

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase

Clase 01.

HTML 1: Conceptos Básicos de HTML

- ▶ 1. Conceptos básicos sobre Full Stack
2. Herramientas a utilizar
3. Instalación del software
4. Explicación del proyecto final
5. Introducción a HTML:
Etiquetas semánticas
6. Etiquetas básicas más comunes

Clase 02.

HTML 2: Listas, Rutas, Multimedia y Tablas

- ▶ 1. Listas y enlaces
2. Rutas absolutas y relativas
3. Elementos en bloque y en línea
4. Multimedia con HTML:
imágenes, video, audio, iframes
5. Accesibilidad en Multimedia
(etiquetas alt, subtítulos en video)
6. Tablas



¿Ya viste la “Introducción al programa” disponible en el campus virtual?

La visualización y resolución de un breve cuestionario es de carácter obligatorio para desbloquear los contenidos de las primeras 2 clases



Métodos de evaluación

Nuestro objetivo es prepararte para enfrentar los desafíos del siglo XXI y facilitar tu inserción en el mercado laboral. Para lograrlo, hemos desarrollado un programa que enfatiza la ejercitación constante y el seguimiento continuo. A continuación, te explicamos cómo serás evaluado a lo largo de la cursada:



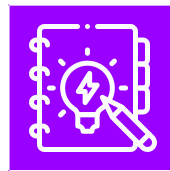
Ejercicios Prácticos



Cuestionarios

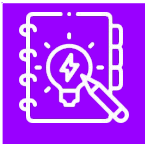


Ruta de Avance



Proyecto Integrador

La información detallada de cada evaluación está disponible en el apartado
“Introducción” dentro del Campus Virtual.



Proyecto Integrador



Obligatorio | Entregable

¿Cómo se construye el proyecto integrador?

Al final de la cursada, serás evaluado mediante la entrega de un Proyecto Integrador, que es fundamental para completar el curso y cumplir con los requisitos de egreso. Este proyecto se construirá de manera progresiva, combinando la resolución de **Ejercicios** y el seguimiento de las 4 **Ruta de Avance** presentes a lo largo de la cursada.

Las Rubricas de Evaluación de este proyecto final integrador estará constituido por las 4 Rutas de Avance presentes a lo largo de la cursada.

Desarrollarás una página o sitio web completa, que combine todos los conocimientos adquiridos a lo largo del curso. Este proyecto consistirá en la creación de un sitio web de e-commerce dinámico e interactivo, que consuma datos de una API REST para mostrar productos, y permita a los usuarios añadir productos a un carrito de compras. Tu proyecto deberá demostrar una sólida comprensión de la estructuración semántica, el diseño responsivo, la manipulación del DOM, la interacción con APIs, y la implementación de funcionalidades específicas de un e-commerce.



After Class

El espacio "After Class" está diseñado para ofrecerte apoyo adicional y facilitar tu progreso durante la cursada. Aunque es opcional, te recomendamos que utilices este espacio para optimizar tu aprendizaje y el desarrollo de tu proyecto integrador.

Beneficios de asistir:

- **Consultas y Asesoría:** Aprovecha este tiempo para resolver cualquier duda o consulta que tengas sobre el contenido de las clases, ejercicios prácticos, o cualquier aspecto relacionado con tu proyecto integrador. Podrás recibir orientación más personalizada de los instructores y obtener aclaraciones que te ayudarán a comprender mejor los conceptos y mejorar tu desempeño.



Frecuencia: Una vez por semana en un día distinto y en la franja horaria de la cursada regular.

Front-End

JavaScript

Este trayecto está diseñado para personas con conocimientos básicos en lógica de programación, proporcionándoles una formación completa y práctica desde cero. Cubriendo desde los fundamentos esenciales de programación, hasta el desarrollo de interfaces de usuario con tecnologías modernas de Front-End.





Conceptos básicos



Conceptos básicos

HTML



HyperText Markup Language: Lenguaje de marcado de hipertexto. Define la estructura, semántica y contenido de las páginas Web. El navegador (cliente) interpreta el HTML y lo representan en pantalla. Permite agregar texto, imágenes, enlaces, tablas, listas, formularios, etc.

Última versión del estándar: HTML 5.

No todos los navegadores son capaces de interpretar un mismo código de una manera unificada (siempre debemos probar que funcione en la mayoría de ellos).

Utiliza etiquetas (tags) que definen la estructura del documento.

Texto plano con extensión .html

Diferencias entre Página Web y Sitio Web

Página Web

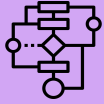
- Es un documento HTML (texto, imágenes, videos, CSS, Javascript).
- Pueden ser estáticas (sólo código) o dinámica (con, por ejemplo, contenido extraído de una base de datos).



Sitio Web

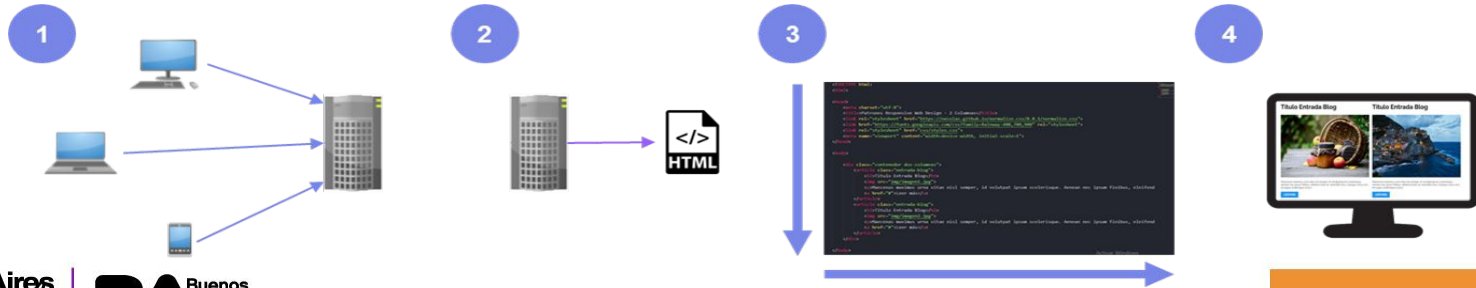
- Son un conjunto de páginas Web estructuradas en un dominio.
- Se pueden crear escribiendo código o mediante CMS o builders.
- Pertenecen a empresas, organizaciones o personas.





HTML: Funcionamiento

1. El navegador (cliente) le pide información al servidor.
2. El servidor devuelve la información al cliente en un archivo HTML.
3. El navegador en el cliente lee el archivo de arriba hacia abajo y de izquierda a derecha para interpretar la información.
4. Tiene en cuenta las etiquetas que tiene el documento y las va renderizando en la pantalla (lo que se ve en el navegador).



Editores de texto

Necesitamos utilizar un editor de texto, que genere archivos de “texto plano”.

Hay muchos disponibles, pero para esta cursada, **Visual Studio Code** ya que es uno de los más recomendables.

Nota: 💡 Haz click en la imagen para descargar el IDE.

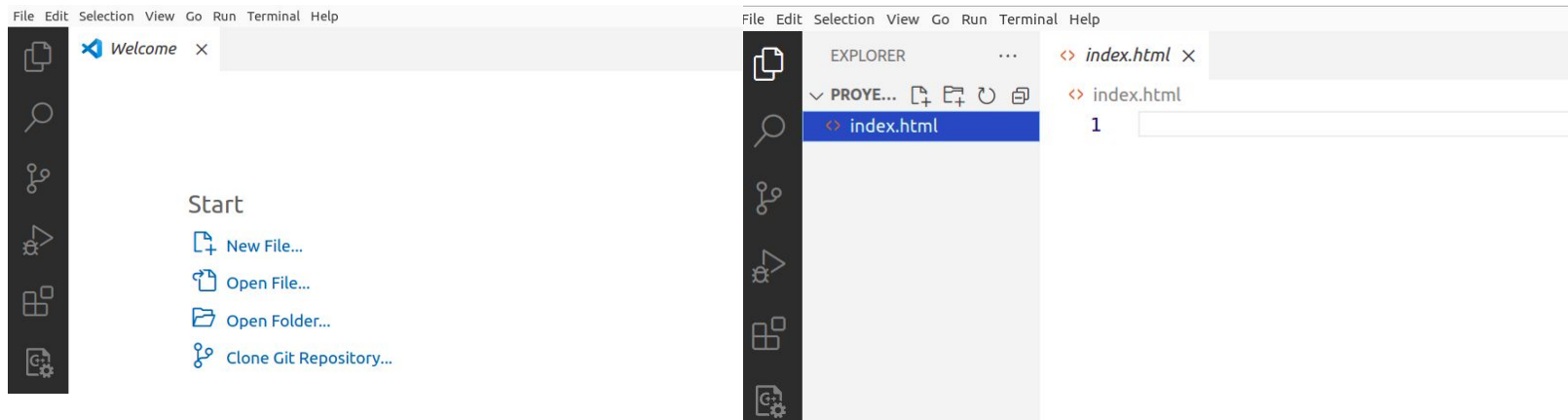


Visual Studio Code

Primer archivo HTML

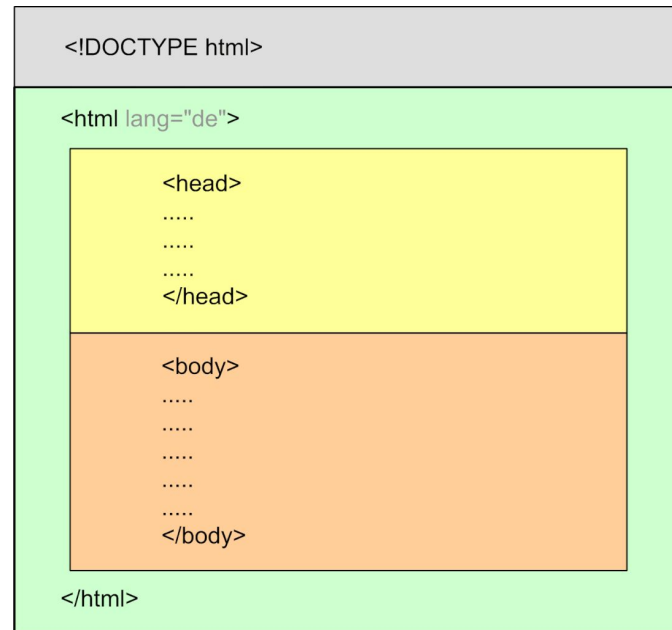


Abrimos VSCode, vamos a 'File' y seleccionamos 'New File'. Luego, escribimos index.html como nombre del archivo. Es fundamental incluir la extensión .html al final para que VSCode reconozca y trate nuestro código como HTML



Estructura Básica de una Web

- **<!DOCTYPE html>** Indica que la versión corresponde a HTML5.
- **<html lang= "es">** Es la etiqueta principal que engloba al resto de las etiquetas, el atributo lang define el tipo de lenguaje.
- **<head>** Es la cabeza del documento que contiene los metadatos de la página web.
- **<meta charset="utf-8">** Indica al navegador qué tipo de caracteres contiene la página, con el atributo charset vamos a indicar el conjunto de caracteres que vamos a usar y con el valor "utf-8" abarcamos a la mayoría de los sistemas de escritura.
- **<title>** Indica el título de la página Web, que se visualiza en la barra de título del navegador.
- **<body>** Es el cuerpo del documento donde va a estar todo el contenido que vamos a mostrar.



¿Cómo comenzar una página Web ?

En VSC si escribimos ! y presionamos Enter este código se escribirá automáticamente.

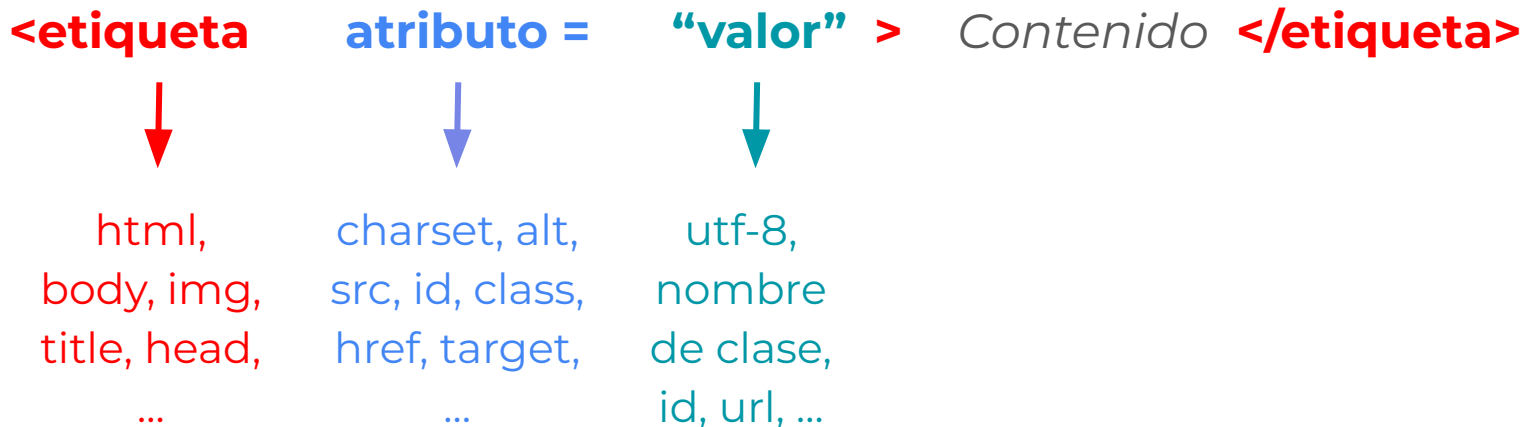
```
<> index.html ● Settings
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8  </head>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




Etiquetas

Etiquetas Semánticas

Las etiquetas semánticas definen qué función cumple cada elemento HTML.

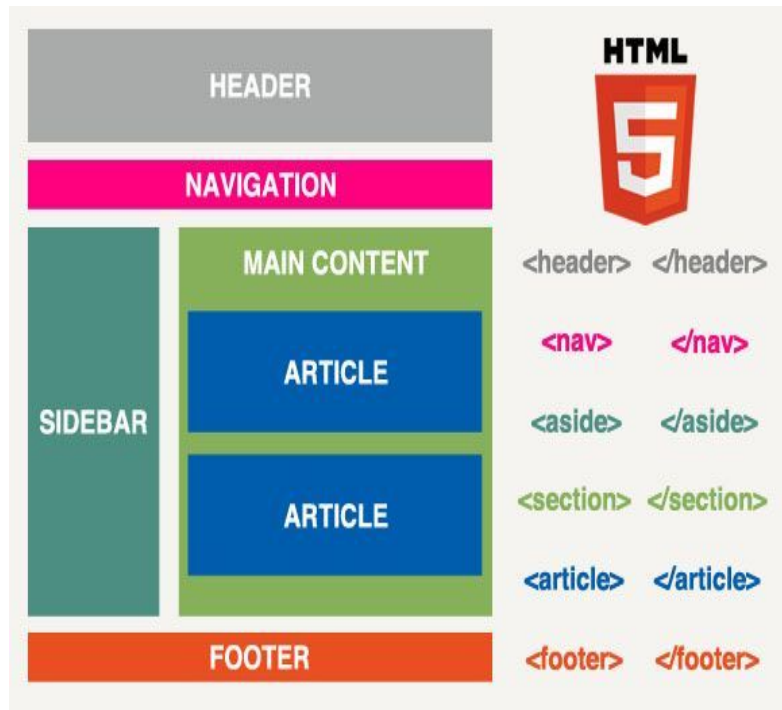


Estructura de una Página Web

Estas etiquetas ayudan a definir la clase de contenido tendrá una página Web

Describen su significado tanto para el navegador como para el desarrollador. A través de ellas los navegadores y buscadores reconocen patrones y una estructura determinada.

Debemos respetarlas porque ayudan al navegador a entender su significado para mostrarlo en pantalla y ayudan a los buscadores a reconocer el contenido y la estructura del sitio.



Etiquetas semánticas



- **<header>** Es la cabecera visual del documento. No confundir con **<head>**, que es el encabezado del documento HTML.
- **<nav>** Apartado de navegación (enlaces de secciones, categorías, etc...). También permite dividir en categorías una sección.
- **<main>** Contenido principal del documento.
- **<footer>** Pie de página.
- **<section>** Define una sección en un documento.
- **<aside>** Agrupación de contenido no relacionado con el tema principal del documento. Suele usarse para agregar publicidad.
- **<article>** Artículo. Parte principal de un escrito (posts en blogs, artículos en diarios, mensajes en foros, comentarios, etc.)
- **<address>** Agrupación con la información de contacto del autor del artículo o documento.

Etiquetas más comunes



- Definen qué función cumple cada elemento dentro de la página.
- Estructura básica, tienen una apertura y un cierre que describen la información contenida entre ellas, aunque algunos casos solamente tienen apertura, como la etiqueta `
` (line break o salto de línea):

```
<b>Texto de Ejemplo</b>
```

Texto de ejemplo

```
<p>Hola, estamos en el párrafo 1</p>
```

Hola, estamos en el párrafo 1

```
<p>Ahora hemso cambiado de párrafo</p>
```

Ahora hemos cambiado de párrafo

```
<p>Este es un párrafo <br> que continua  
en el siguiente renglón</p>
```

Este es un párrafo
que continúa en este renglón

Heading

<h1>, <h2>, <h3>....<h6>: Encabezados, numerados del 1 al 6 por orden de relevancia. Es importante respetar ese orden para que el navegador entienda la estructura de la página.

```
<h1>Encabezado en H1</h1>
```

```
<h2>Encabezado en H2</h2>
```

```
<h3>Encabezado en H3</h3>
```

```
<h4>Encabezado en H4</h4>
```

```
<h5>Encabezado en H5</h5>
```

```
<h6>Encabezado en H6</h6>
```

Encabezado en H1

Encabezado en H2

Encabezado en H3

Encabezado en H4

Encabezado en H5

Encabezado en H6

Tip: 💡 En VSC si escribimos **h1** y presionamos TAB la etiqueta de cierre se escribe automáticamente.

Párrafos

<p>: Representa un párrafo.

```
<p>Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Expedita sequi reiciendis eius  
sapiente placeat, esse illum quas  
tenetur cupiditate fugit vero,  
minima quo? Tenetur sed dolore  
error maxime. Expedita, beatae.</p>
```



Tip #1: Lorem Ipsum

En VSC si escribimos Lorem100 se escribirá un texto de ejemplo de 100 palabras. Y escribiendo p*3 se crearán 3 etiquetas <p>.



Tip 02: Word Wrap

Para ajustar automáticamente el texto al ancho de la ventana del editor en Visual Studio Code, utiliza el siguiente atajo:

- . Windows/Linux: Alt + Z
- . macOS: Option + Z

¡Así, tu código siempre estará visible sin necesidad de desplazarte horizontalmente!

Etiquetas básicas (formatos de texto)

```
<b>Texto en negrita o bold</b>  
<u>Texto subrayado</u>  
<mark>Texto marcado</mark>  
<ins>Texto insertado</ins>  
<small>Texto más pequeño</small>  
<i>Texto en Itálica o cursiva</i>  
<del>Texto tachado</del>  
Texto normal <sup>Texto en superíndice</sup>  
Texto normal <sub>Texto del subíndice</sub>  
Este es un <em>texto con énfasis</em>  
Este es un <strong>texto importante</strong>
```

Texto en negrita o bold
Texto subrayado
Texto marcado
Texto insertado
Texto más pequeño
Texto en Itálica o cursiva
~~Texto tachado~~
Texto normal Texto en superíndice
Texto normal Texto del subíndice
Este es un *texto con énfasis*
Este es un **texto importante**

¡Vamos a la práctica!





Ejercicios Prácticos



Optativos | No entregables

Estructura básica

Crear la estructura básica del proyecto, incluyendo las etiquetas `<header>`, `<main>`, `<nav>` y `<footer>`. Dentro de `<header>`, incluye un título `<h1>` con el nombre del proyecto.

Para resolver este ejercicio debes:

1. **Abrir Visual Studio Code:**
 - Abre Visual Studio Code y crea un nuevo archivo.
 - Guarda el archivo con el nombre `index.html`.
2. **Escribir la estructura básica del documento HTML:**
 - A continuación, debes escribir el código HTML que conformará la estructura básica de tu proyecto. El código debe incluir las etiquetas `<header>`, `<nav>`, `<main>`, y `<footer>`.



Ejercicios Prácticos



Optativos | No entregables

README.md

Agregar un archivo README.md en el proyecto, explicando brevemente de qué tratará la página que se va a desarrollar.

Para resolver este ejercicio debes:

1. **Crear el archivo README.md:**
 - Dentro del mismo directorio donde guardaste `index.html`, crea un nuevo archivo llamado `README.md`.
2. **Escribir la descripción del proyecto:**
 - Abre el archivo `README.md` en Visual Studio Code y escribe una breve descripción del proyecto.

