

## Práctica N 1. Primeros pasos en HTML.

Resultados de aprendizaje:

- Implementar la estructura básica de un archivo HTML
- Implementar etiquetas de texto para formatos específicos

### INTRODUCCION

**HTML** (HyperText Markup Language) es el lenguaje estándar para la creación de páginas web. Su principal propósito es estructurar y organizar el contenido en la web, como texto, imágenes, videos y enlaces, permitiendo que los navegadores los muestren correctamente. HTML no es un lenguaje de programación, sino un **lenguaje de marcado** que utiliza etiquetas (tags) para definir elementos y darle un significado semántico al contenido.

Fue desarrollado por Tim Berners-Lee en 1991, y desde entonces ha evolucionado a través de varias versiones, siendo **HTML5** la más reciente y ampliamente utilizada.

### COPONENTES BÁSICOS:

**HTML** se basa en una estructura jerárquica con elementos que siguen una sintaxis específica. A continuación, se explican sus componentes principales:

#### 1. Etiquetas (Tags)

Las etiquetas son el núcleo de HTML y se utilizan para definir los elementos. Generalmente, cada etiqueta tiene una **apertura** y un **cierre**, con el contenido en el medio.

*Ejemplo:*

`<p>Este es un párrafo</p>`

La etiqueta de apertura es `<p>` y la de cierre es `</p>`.

#### 2. Atributos

Los atributos proporcionan información adicional sobre un elemento. Se escriben dentro de la etiqueta de apertura y suelen tener un formato clave-valor.

*Ejemplo:*

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

**href** es el atributo que define la URL del enlace.

### 3. Elementos

Un elemento está compuesto por una etiqueta de apertura, un contenido (opcional) y una etiqueta de cierre.

**Ejemplo:**

`<h1>Título de la página</h1>`

Aquí, el elemento `<h1>` representa un título.

### 4. Doctype

El **Doctype** es una declaración que indica al navegador la versión de HTML utilizada en el documento.

**Ejemplo:**

`<!DOCTYPE html>`

Esto especifica que se está utilizando HTML5.

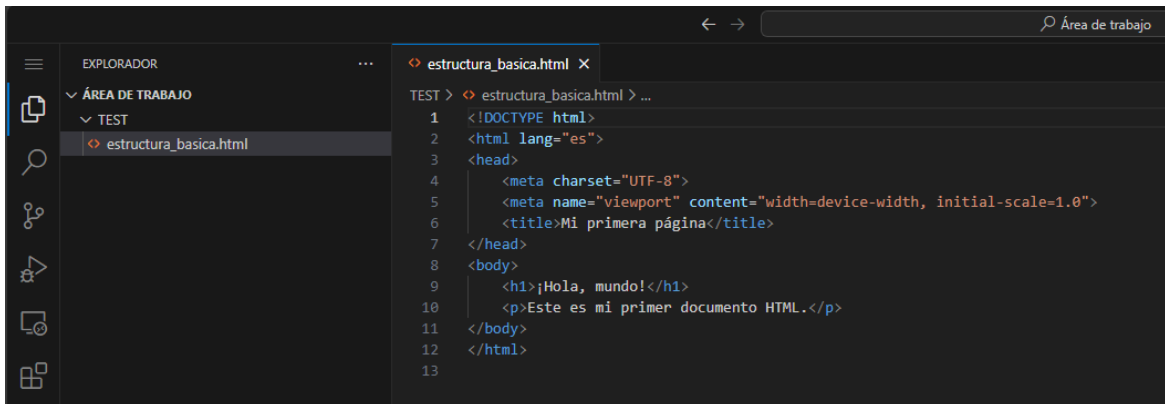
### 5. Comentarios

Se utilizan para **agregar notas al código** y no son visibles para los usuarios.

**Ejemplo:**

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

## Estructura básica de un documento HTML



**<!DOCTYPE html>:** Define que se está usando HTML5.

**<html>:** Es el elemento raíz que contiene todo el documento.

**<head>:** Contiene información "meta" sobre el documento, como el título y los enlaces a recursos externos (CSS, JavaScript).

**<body>:** Contiene el contenido visible de la página, como texto, imágenes y enlaces.

## EDITORES DE TEXTO PARA LA CREACIÓN DE SITIOS WEB

Los editores de texto son herramientas fundamentales para el desarrollo web, ya que permiten escribir, editar y organizar el código fuente de los sitios web. Van desde opciones básicas hasta entornos avanzados que ofrecen características específicas para mejorar la productividad y la calidad del desarrollo.

### Tipos de editores de texto:

#### *Editores de texto simples*

Son herramientas básicas que permiten escribir código sin funciones avanzadas.

#### **Ejemplos:**

- **Bloc de notas (Windows):** Simple, pero carece de resaltado de sintaxis.
- **TextEdit (macOS):** Similar al Bloc de notas, con opciones básicas.

#### **Ventajas:**

- ✓ Ligeros.
- ✓ Buenos para tareas muy simples.

#### **Desventajas:**

- ✓ No tienen funciones como **autocompletado**, **resaltado de sintaxis** o **previsualización**.

#### *Editores de texto enriquecidos (Code Editors)*

Están diseñados específicamente para programar, ofreciendo herramientas que facilitan el desarrollo web.

#### **Ejemplos populares:**

- **Visual Studio Code (VS Code):** Editor altamente personalizable con soporte para múltiples lenguajes, extensiones y herramientas de depuración.
- **Sublime Text:** Ligero, rápido y con características avanzadas como atajos y selección múltiple.
- **Atom:** Personalizable, desarrollado por GitHub, con gran comunidad de soporte.
- **Notepad++:** Sencillo pero potente, ideal para principiantes.
- **Gedit:** Sencillo para notas, pero potente para uso avanzado en entornos de desarrollo.

#### **Ventajas:**

- ✓ Resaltado de sintaxis para HTML, CSS, JavaScript, etc.
- ✓ Autocompletado para reducir errores.
- ✓ Integración con herramientas externas (control de versiones, previsualización).

#### **Desventajas:**

- ✓ Algunos pueden ser pesados en recursos para equipos con hardware limitado.

## Entornos de Desarrollo Integrado (IDEs)

Los **IDEs** son herramientas más completas que incluyen no solo un editor de texto, sino también herramientas de depuración, gestión de proyectos y control de versiones.

### Ejemplos:

- Visual Studio
- WebStorm
- Eclipse
- Netbeans
- Adobe Dreamweaver

### Ventajas:

- ✓ Excelente para proyectos grandes y complejos.
- ✓ Integración con múltiples tecnologías y herramientas.

### Desventajas:

- ✓ Curva de aprendizaje más pronunciada.
- ✓ Mayor consumo de recursos del sistema.

### Importancia de conocer estas herramientas:

#### 1. Eficiencia y productividad

Los editores avanzados facilitan el desarrollo mediante funciones como el autocompletado, la navegación entre archivos, el control de errores y la depuración. Esto permite escribir código más rápido y con menos errores.

#### 2. Mejor experiencia de usuario

Editores como VS Code permiten previsualizar cambios, probar código en tiempo real y detectar problemas antes de publicarlos. Esto mejora la calidad de los sitios web.

#### 3. Mantenimiento del código

Herramientas como el resaltado de sintaxis, el formato automático y la organización por carpetas hacen que el código sea más legible y fácil de mantener, especialmente en proyectos colaborativos.

#### 4. Compatibilidad con tecnologías modernas

Editores como Sublime Text o VS Code son compatibles con extensiones y plugins que soportan tecnologías actuales como frameworks (React, Angular), preprocesadores de CSS (Sass, Less) y control de versiones (Git).

#### 5. Adaptación al mercado laboral

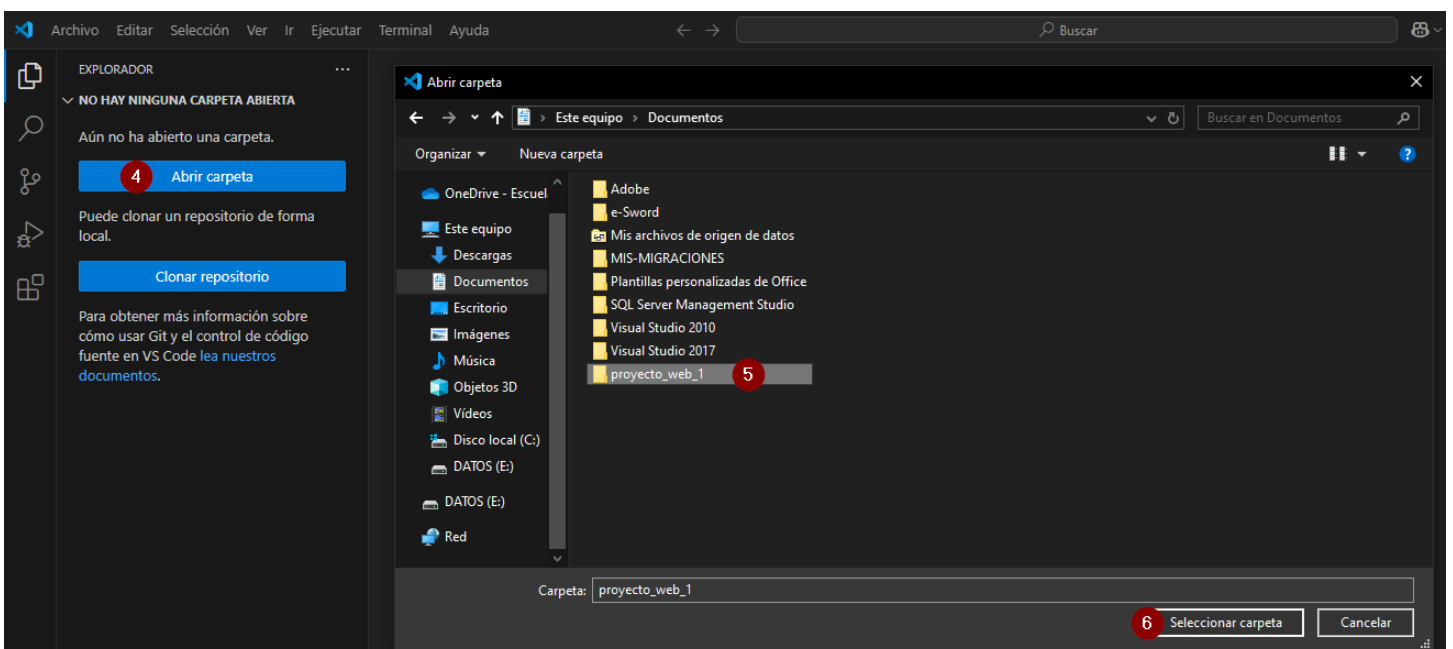
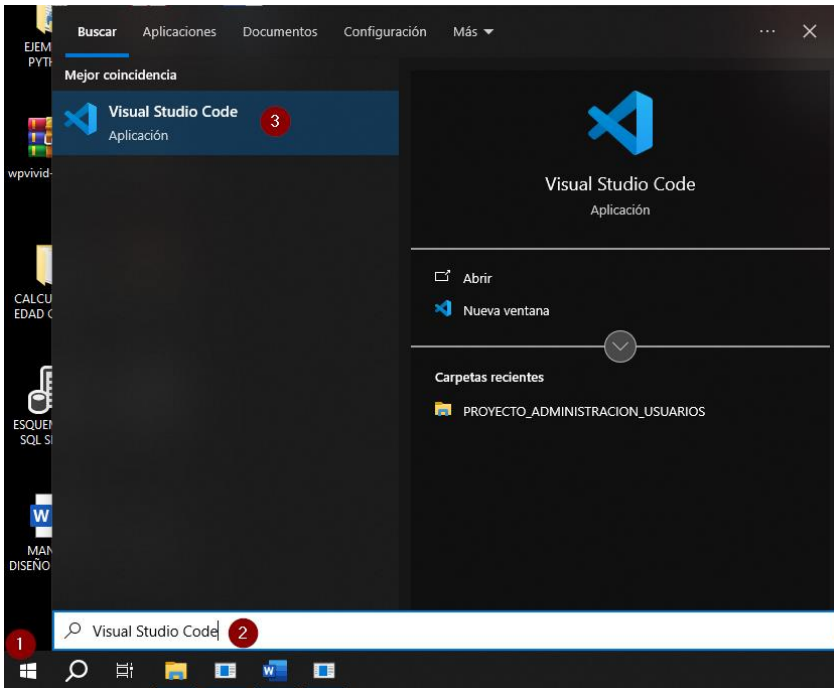
Conocer herramientas modernas como VS Code o WebStorm es fundamental, ya que son ampliamente utilizadas en la industria. Esto mejora las oportunidades de empleo y facilita la colaboración en equipos.

Ahora que ya conocemos las herramientas necesarias para empezar con el desarrollo de sitios web, crearemos nuestra primera página web usando el editor de texto **Visual Studio Code** siguiendo los pasos siguientes:

### Configuración de la carpeta contenedora de nuestro proyecto web.

1. Crear una carpeta en el Sistema Operativo y colocarle el nombre “**proyecto\_web\_1**”. Esta carpeta puede ser creada en cualquier ubicación de nuestro sistema (documentos, escritorio, etc.)
2. Abrir el editor de código **Visual Studio Code** a través del menú inicio del sistema.
3. Abrir la carpeta previamente creada con el editor de código **VS Code**.

### Ejemplo:



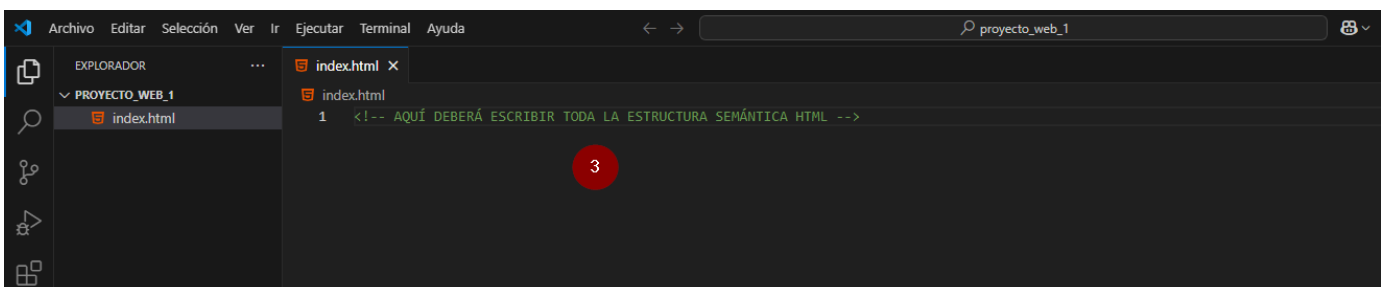
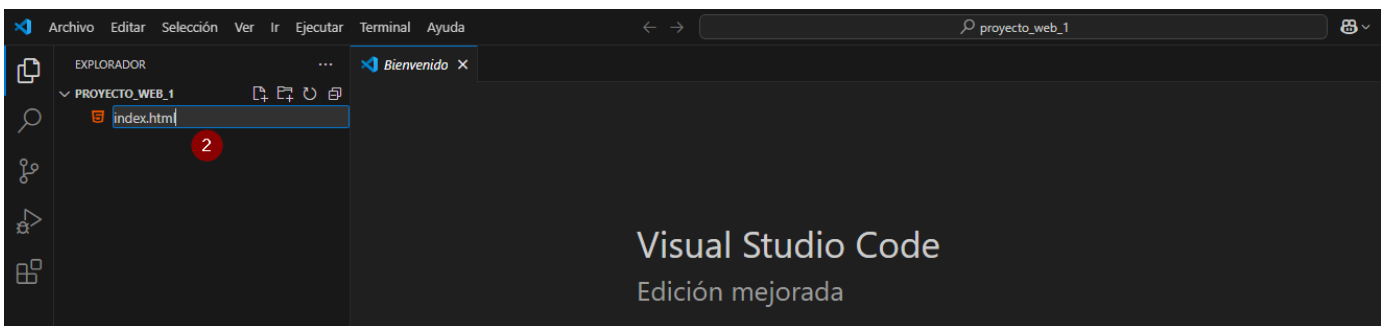
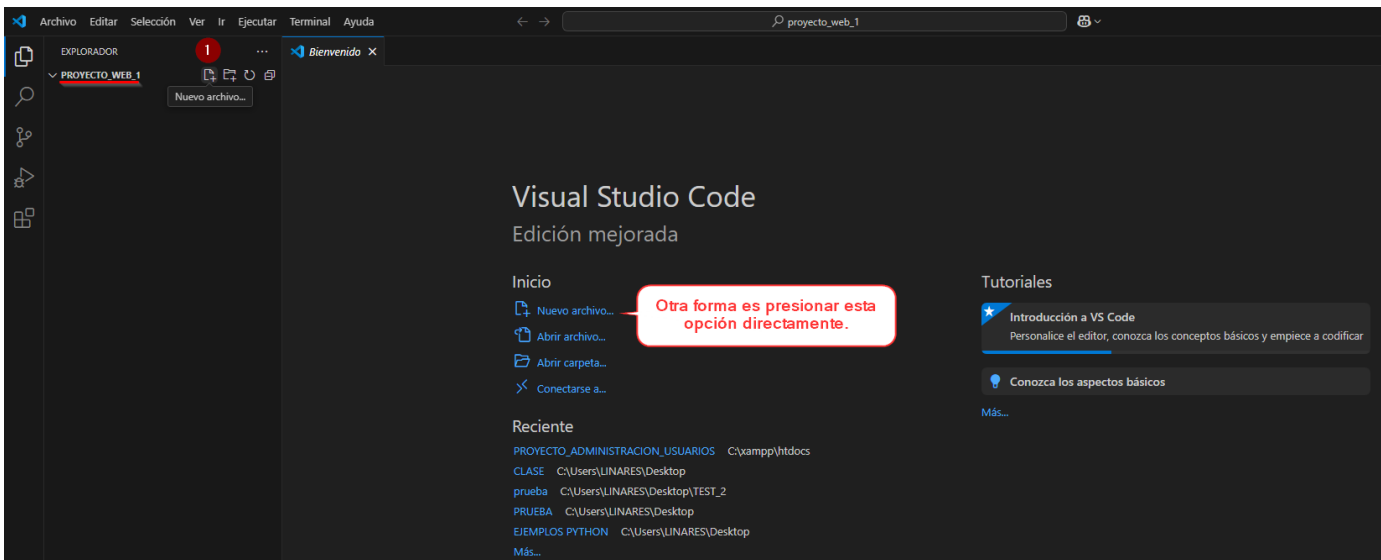
Ahora que ya establecimos la carpeta de nuestro proyecto web, continuaremos con la creación de nuestra primera página web.

### Creación del primer documento HTML (página web).

1. Ubicarnos sobre el nombre de nuestra carpeta configurada, luego presionar clic sobre el icono de página con un signo más que aparece dinámicamente.
2. Colocar un nombre al archivo sin olvidar que al final del nombre deberá escribir la extensión **(.html)** para que éste sea reconocido como un documento html y pueda ser abierto con un navegador web.
3. Luego de colocar el nombre del archivo más su extensión presionar la tecla **ENTER**.

**NOTA:** Es importante comprender y tener en cuenta que el archivo que servirá como puerta de entrada hacia el sitio web tenga el nombre de **“index.html”**, los demás archivos podrán llamarse según convenga.

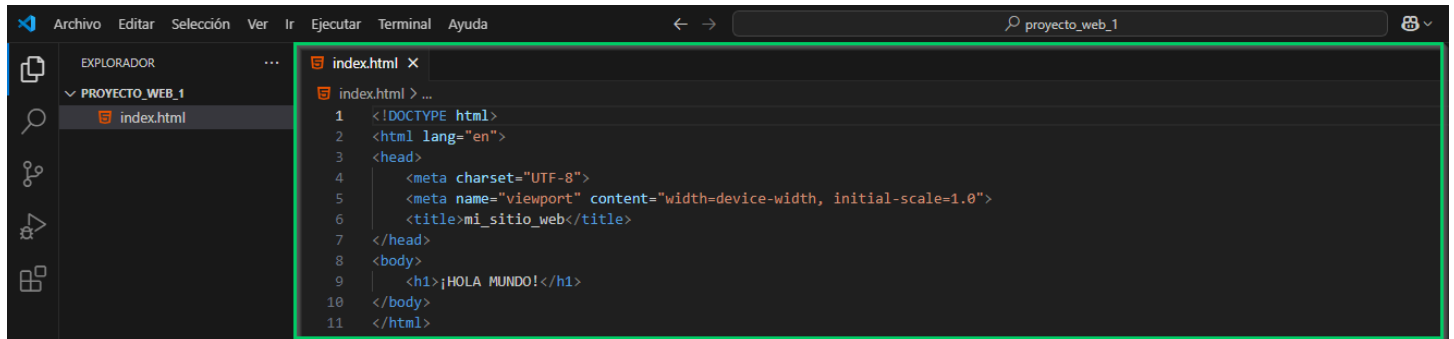
### Ejemplo:



**Agregar una estructura básica html con un Título presentando nuestro primer ¡HOLA MUNDO! en el documento “index.html” previamente creado.**

Para escribir la estructura de su página web, presionar clic en el panel derecho (panel central) del editor de código, luego escribir las etiquetas html correspondientes:

### Ejemplo:



The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left shows a project named 'PROYECTO\_WEB\_1' with a file named 'index.html'. The main editor area displays the content of 'index.html', which is a basic HTML document structure. The code is as follows:

```
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>mi_sitio_web</title>
7 </head>
8 <body>
9   <h1>¡HOLA MUNDO!</h1>
10 </body>
11 </html>
```

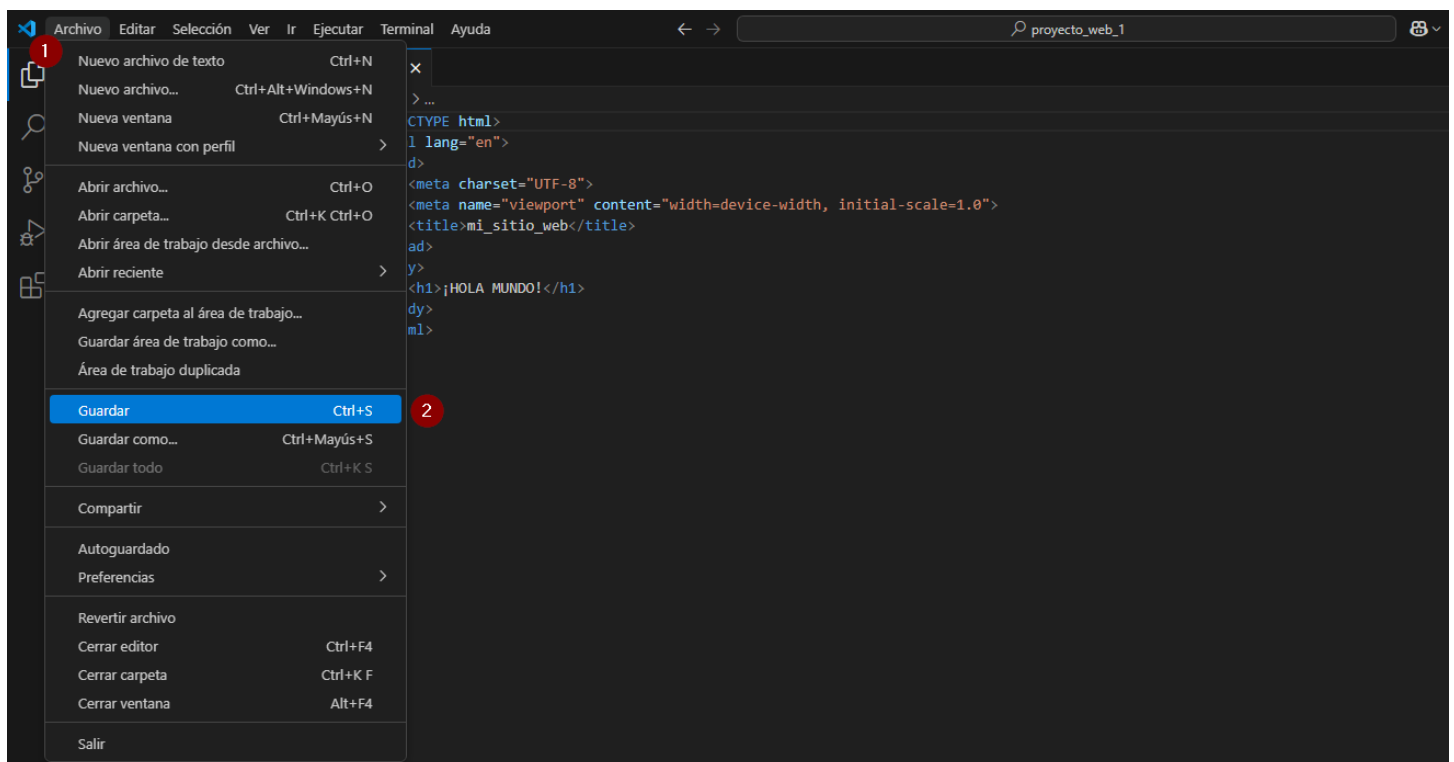
**Guardar el documento HTML previamente creado (o los cambios realizados en el archivo).**

Antes de guardar el archivo, siempre es importante asegurarnos que el archivo cuenta con la extensión **.html**, de lo contrario, el navegador no interpretará el código fuente escrito en el, por lo tanto, no podrá renderizar el **DOM** de ese archivo.

Para guardar el archivo que se está editando con **VS Code**, únicamente presionar la combinación de teclas **CTRL+S**. Esto hará que todo el documento o los cambios realizados en él queden correctamente guardados.

También podrá guardar el archivo presionando clic en el menú **Archivo** del editor de texto VS Code y luego eligiendo la opción **Guardar**.

### Ejemplo:

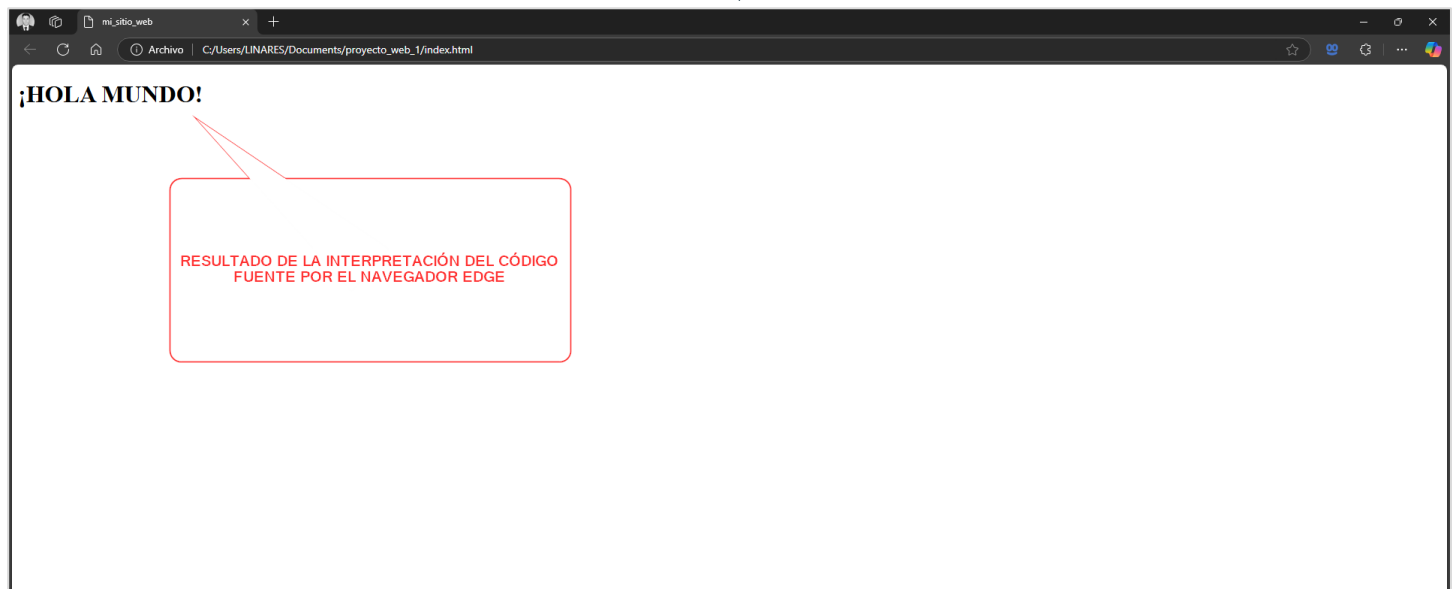
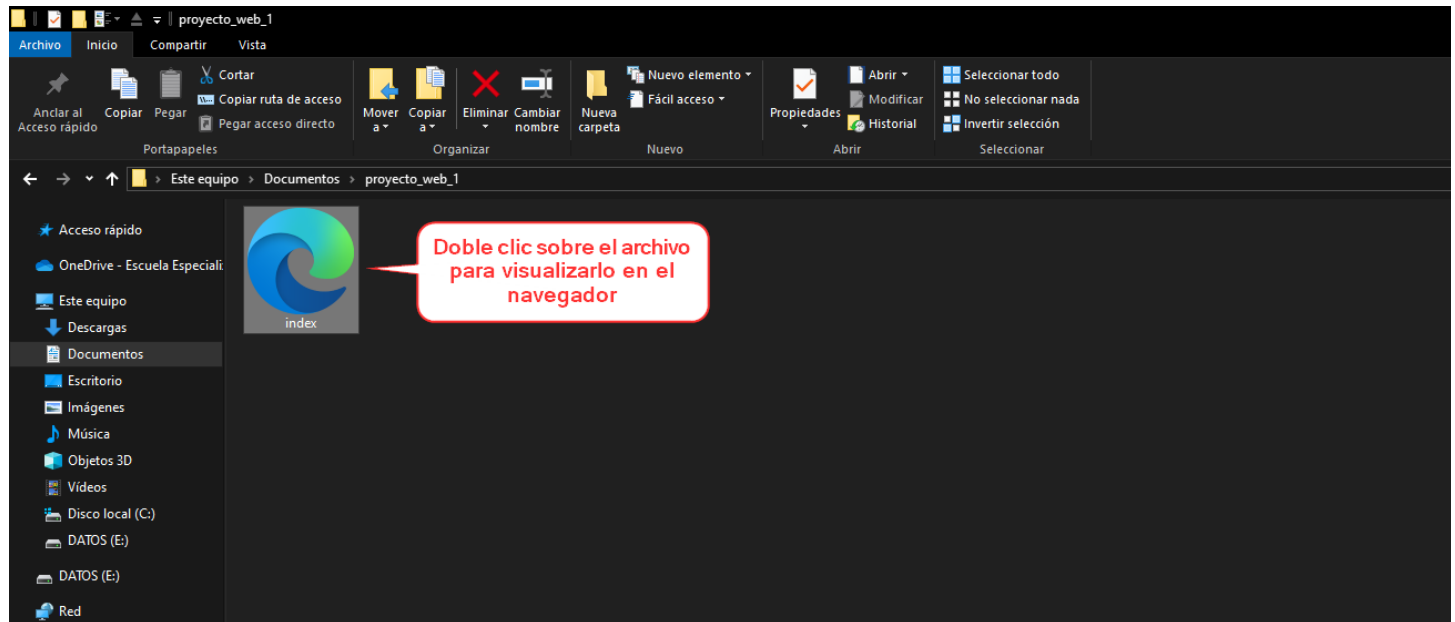


## Visualizar el documento HTML previamente creado

Para visualizar el contenido del archivo se necesitará el uso de un navegador web (Microsoft Edge, Google Chrome, Mozilla Firefox, Opera, etc.)

Tras crear y guardar el archivo en formato **.html**, cualquier navegador web será capaz de abrir e interpretar su código fuente. Para ello únicamente deberá presionar **Doble Clic sobre el archivo html creado**.

### Ejemplo:





## EJERCICIO PROPUESTO: CREA TU PÁGINA DE BIOGRAFÍA

### INSTRUCCIONES:

Desarrolla una página web en un solo archivo **HTML** que contenga información sobre ti. Asegúrate de organizar y estructurar el contenido utilizando las etiquetas **semánticas** de HTML5. Sigue los requisitos a continuación:

#### Requisitos del ejercicio:

**1. Título de la página:**

- ✓ Establece un título claro y descriptivo en la etiqueta `<title>` (por ejemplo: "Biografía de [Tu Nombre]").

**2. Encabezado principal:**

- ✓ Usa la etiqueta `<header>` para incluir tu nombre como un encabezado principal (`<h1>`) y una breve frase que te describa.

**3. Cuerpo de la página:** (Dentro del `<main>` incluye)

- ✓ Un párrafo introductorio (`<p>`) donde hables brevemente quién eres.
- ✓ Una sección de educación con una lista ordenada (`<ol>`) de tus logros académicos o intereses relacionados con tus estudios.
- ✓ Una sección de hobbies con una lista no ordenada (`<ul>`) que describa tus pasatiempos.
- ✓ Un enlace (`<a>`) a alguna página externa que te inspire o que esté relacionada contigo (por ejemplo, tu universidad o algo que te interese).

**4. Pie de página:** (Usar la etiqueta `<footer>` para incluir información como)

- ✓ Una frase personal.
- ✓ La fecha actual utilizando texto simple.

**5. Semántica:**

- ✓ Utilizar las etiquetas semánticas adecuadas para cada sección: `<header>`, `<main>`, `<section>`, `<footer>`, `<h1>`, `<h3>`, `<p>`, `<ul>`, `<ol>`, entre otras.