



# ARQUITECTURA DE ENTORNOS WEB

Tema: HTML

**Sumilla:**

La presente guía le permite al estudiante dominar las etapas de un proyecto web. Maneja eficientemente diseñando los elementos gráficos para un proyecto web. Elabora etiquetas CSS3 y HTML5 implementando páginas web, multimedia. Desarrollaran formularios en una página web con elementos input, button y textArea.

## Contenido

HTML.....	3
¿Qué es HTML? .....	3
Primeras especificaciones de HTML.....	3
Marcador HTML .....	4
Elementos.....	4
Atributos .....	5
Navegadores de Internet y Herramientas de Desarrollo .....	5
Navegadores .....	5
Herramientas de Desarrollo de los Navegadores .....	6
Etapas de un Proyecto Web .....	7
Etapas de los Proyectos Web .....	7
Arquitectura Web .....	8
Arquitectura de un Proyecto Web .....	8
¿Cómo plantear la arquitectura de contenidos? .....	9
Mapa Web.....	10
Mapa del Sitio Web .....	10
Ventajas y Técnicas de la Construcción Web.....	11
ACTIVIDAD 01 .....	13
ACTIVIDAD 02 .....	16
ACTIVIDAD 03 .....	21
ACTIVIDAD SEMANA.....	24
REFERENCIAS.....	25

## HTML

### ¿Qué es HTML?

HTML, acrónimo en inglés de HyperText Markup Language (“lenguaje de marcado de hipertexto”), hace referencia al lenguaje de marcado utilizado en la creación de páginas web. Este estándar que sirve de referencia para el software que interactúa con la elaboración de páginas web en sus diferentes versiones. Define una estructura básica y un código (denominado código HTML) para la presentación de contenido de una página web, que incluye texto, imágenes, videos, juegos, entre otros elementos. Este estándar es gestionado por el World Wide Web Consortium (W3C) o Consorcio WWW, una organización dedicada a la estandarización de la mayoría de las tecnologías asociadas a la web, especialmente en lo relacionado con su escritura e interpretación. HTML se considera el lenguaje web más importante y su invención crucial para el surgimiento, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que prevalece en la visualización de páginas web y es adoptado por todos los navegadores actuales.

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (Red Informática Mundial).

HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Las marcas HTML incluyen "elementos" especiales como <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> y muchos otros.

Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla. Por ejemplo, la etiqueta <title> se puede escribir como <Title>, <TITLE> o de cualquier otra forma..

### Primeras especificaciones de HTML

Tim Berners-Lee, en 1991, describe 18 elementos que conformaban el diseño inicial y relativamente simple de HTML. Trece de estos elementos aún perduran en HTML 4.

Berners-Lee consideraba el HTML una ampliación de SGML, pero no fue formalmente reconocida como tal hasta la publicación a mediados de 1993, cuando la IETF (en español: Grupo de Trabajo de Ingeniería de Internet) publicó una primera propuesta para la especificación de HTML: el borrador del Hypertext Markup Language de Berners-Lee y Dan Connolly. Esta propuesta incluía una Definición de Tipo de Documento SGML para establecer la sintaxis. A pesar de que el borrador expiró a los seis meses, fue notable por reconocer la etiqueta específica del navegador Mosaic usada para insertar imágenes sin cambiar la línea. Esto reflejaba la filosofía del IETF de basar los estándares en prototipos con éxito. De manera similar, el borrador competidor de Dave Raggett HTML+ (Hypertext Markup Format) (Formato de Marcaje de Hipertexto), de finales de 1993, sugería estandarizar características ya implementadas, como las tablas.

## Marcador HTML

El HTML se escribe utilizando “etiquetas”, encerradas por corchetes angulares (<, >, /). Además de describir la estructura de un documento, el HTML puede, hasta cierto punto, definir su apariencia y hacer referencia a un tipo de programa llamado script, el cual puede influir en el comportamiento de navegadores web y otros procesadores de HTML.

El término HTML, también se refiere al contenido del tipo de MIME text/html o de manera más amplia, como un término genérico para el HTML. Esto puede ser en forma derivada del XML (como XHTML 1.0 y versiones posteriores) o directamente derivado de SGML (como HTML 4.01 y versiones anteriores). El HTML consta de varios componentes esenciales, incluyendo elementos y sus atributos, tipos de datos y la declaración de tipo de documento.

## Elementos

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Tanto los atributos como los contenidos tienen ciertas restricciones para que el documento HTML se considere válido. Generalmente, un elemento se compone de una etiqueta de apertura (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre correspondiente (por ejemplo, </nombre-de-elemento>). Los atributos del elemento se especifican dentro de la etiqueta de apertura, mientras que el contenido se sitúa entre ambas etiquetas (por ejemplo, <nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>). Algunos elementos, como <br>, carecen de contenido y no requieren una etiqueta de cierre. Debajo se listan varios tipos de elementos de marcado usados en HTML.

El marcado estructural describe el propósito del texto. Por ejemplo, <h2>Golf</h2> establece «Golf» como un encabezado de segundo nivel, el cual se mostraría en un navegador de una manera similar al título «Marcador HTML» al inicio de esta sección. El marcado estructural no determina cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el formato de los elementos. Se puede aplicar un formato específico al texto por medio de hojas de estilo en cascada.

El marcado presentacional se encarga de definir la apariencia del texto, independientemente de su función. Por ejemplo, <b>negrita</b> indica que los navegadores web deben mostrar el texto en negrita, pero no especifica cómo deben interpretarlo los navegadores que presentan el contenido de otra manera distinta (como aquellos que leen el texto en voz alta). En contraste, en el caso de <b>negrita</b> e <i>itálica</i>, existen elementos con una naturaleza más semántica: <strong>énfasis fuerte</strong> y <em>énfasis</em>. Es evidente cómo un lector de pantalla debería interpretar estos dos últimos elementos. Sin embargo, a pesar de su similitud visual, son diferentes a equivalentes en presentación: un lector de pantalla no debería acentuar más el nombre de un libro, aunque este el nombre en itálicas en la pantalla. La mayoría del marcado en presentación ha sido desechado desde la versión 4.01, en favor de las hojas de estilo en cascada.

El marcado hipertextual se emplea para enlazar partes del documento con otros documentos o con otras secciones del mismo documento. Para crear un enlace es necesario utilizar la etiqueta de ancla <a> junto con el atributo href, que establecerá la dirección URL a la que apunta el enlace. Por ejemplo, un enlace que muestre el texto de la dirección y se dirija hacia la Wikipedia podría tener esta forma:

`<a href="<nowiki>https://www.wikipedia.org</nowiki>">https://www.wikipedia.org</a>`.

También es posible crear enlaces con otros objetos, como imágenes `<a href="enlace"></a>`.

## Atributos

En su mayoría, los atributos de un elemento son pares nombre-valor, separados por un signo de igual «=» y escritos dentro de la etiqueta de inicio de un elemento, después del nombre del elemento. El valor puede estar encerrado por comillas dobles o simples, aunque ciertos tipos de valores pueden carecer de comillas en HTML (pero no en XHTML). De todas maneras, dejar los valores sin comillas se considera poco seguro. Por otro lado, en contraste con los pares nombre-elemento, hay algunos atributos que afectan al elemento simplemente por su presencia, como el atributo ismap para el elemento img.

## Navegadores de Internet y Herramientas de Desarrollo

### Navegadores

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de páginas web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir. Un navegador web es un programa de software que permite visualizar la información que contiene una página web, la cual puede estar alojada en un servidor dentro de la World Wide Web, o en un equipo local.

La función de un navegador web es permitir la visualización de documentos de texto, los cuales pueden incluir imágenes, sonido, videos, hipervínculos, entre otros.

Como, por ejemplo, Internet Explorer (el cual fue reemplazado por Microsoft Edge), Google Chrome, Mozilla Firefox, Opera, Safari, los cuales son programas de software que permiten presentar textos, imágenes, videos, sonidos en páginas web, que se almacenan en la Internet o en una red local.

**MICROSOFT EDGE:** Es un navegador web desarrollado por Microsoft que sirve como alternativa a Internet Explorer en sistemas operativos Windows. Se caracteriza por ser rápido, seguro y con funciones integradas para mejorar la productividad y la experiencia del usuario. Además, Edge utiliza tecnología de IA para ofrecer funciones como Copilot, que facilita la búsqueda y el acceso a información.

**INTERNET EXPLORER:** Internet Explorer o IE es un navegador web gratuito desarrollado por Microsoft. Funciona en el sistema operativo Windows, aunque ya está terminando su vida útil y se va a reemplazar con EDGE, el cual tiene las mismas características de IE.

**CHROME:** Es el navegador desarrollado por Google. Es un software gratuito para navegar por las páginas web, actualmente está disponible sólo para Windows; también existe una versión disponible para dispositivos móviles, sistema operativo Linux.

**MOZILLA FIREFOX:** Navegador de código abierto desarrollado por la fundación Mozilla. En sus comienzos, tuvo varios nombres, tales como Phoenix y Firebird. Ya con su nombre actual, Firefox es abreviado en forma oficial como FF, así como también, Fx o fx2. Es reconocible por su logo de un zorro.

**OPERA**

Es un navegador desarrollado por Opera Software. Puede ser instalado en Mac, Windows, FreeBSD, Linux y Solaris. Opera está disponible en varios idiomas.

**SAFARI**

Es un navegador web de código cerrado desarrollado por Apple Inc. El cual está disponible para Mac OS X, iOS (sistema usado por el iPhone, el iPod Touch y el iPad) y Windows.

**Herramientas de Desarrollo de los Navegadores**

Todos los navegadores web modernos incluyen un potente conjunto de herramientas para desarrolladores. Estas herramientas hacen una variedad de cosas, desde inspeccionar HTML, CSS y JavaScript actualmente cargados, hasta mostrar qué activos ha solicitado la página y cuánto tiempo tardaron en cargarse. Este artículo explica cómo utilizar las funciones básicas de las herramientas de desarrollo de tu navegador.

Las herramientas para desarrolladores (devtools) viven dentro de tu navegador en una subventana que se ve más o menos así, dependiendo del navegador que estés utilizando:



Se puede acceder a esa ventana de diferentes maneras. Aquí veremos la mas accesible que es presionando la combinación de teclas **Ctrl + Shift + I** solo en los navegadores que no sean ni Internet Explorer ni Edge en donde se aplica la tecla **F12**, sin embargo, esta última tecla se puede usar libremente en los demás navegadores con el mismo resultado.

## Etapas de un Proyecto Web

### Etapas de los Proyectos Web

Un proyecto web típico comprende varias etapas clave:

- Planificación,
- Diseño,
- Construcción,
- Pruebas,
- Lanzamiento,
- Mantenimiento.

La planificación implica definir los objetivos, el alcance, el público objetivo y la estructura del sitio. El diseño se centra en la experiencia del usuario (UX) y la interfaz de usuario (UI). La construcción, o desarrollo, convierte el diseño en código funcional. Las pruebas aseguran la calidad y el rendimiento, mientras que el lanzamiento pone el sitio web en funcionamiento. Finalmente, el mantenimiento implica correcciones, actualizaciones y mejoras continuas.

Etapas detalladas:

#### 1. Planificación:

**Análisis:** Determinar los objetivos, el público objetivo, las necesidades del cliente y el alcance del proyecto.

**Documentación:** Crear un brief del proyecto con los requisitos, los objetivos y las expectativas del cliente.

**Definición de la estructura:** Crear un mapa del sitio (sitemap) para organizar el contenido y la navegación.

#### 2. Diseño:

**Diseño de UX:** Centrarse en la experiencia del usuario, la usabilidad y la accesibilidad del sitio web.

**Diseño de UI:** Enfocarse en los elementos visuales, la estética y la funcionalidad de la interfaz.

**Prototipado:** Crear modelos interactivos del sitio web para visualizar la estructura y la navegación.

#### 3. Construcción/Desarrollo:

**Programación:** Convertir el diseño en código funcional, utilizando lenguajes de programación como HTML, CSS, JavaScript y frameworks.

**Desarrollo Backend:** Implementar la lógica del sitio web y la gestión de la base de datos.

**Desarrollo Frontend:** Crear la parte visual y la interacción del sitio web.

#### 4. Pruebas:

**Pruebas funcionales:** Validar que cada funcionalidad del sitio web funcione correctamente.

**Pruebas de rendimiento:** Asegurar que el sitio web sea rápido y eficiente.

**Pruebas de usabilidad:** Evaluar la facilidad de uso y la experiencia del usuario.

#### 5. Lanzamiento:

**Publicación:** Poner el sitio web en funcionamiento, alojándolo en un servidor web y asignando un nombre de dominio.

**Optimización:** Optimizar el sitio web para los motores de búsqueda (SEO).

#### 6. Mantenimiento:

**Correcciones de errores:** Resolver problemas y bugs que puedan surgir en el sitio web.

**Actualizaciones:** Mantener el sitio web actualizado con las últimas versiones del software y las mejores prácticas.

**Mejoras:** Implementar nuevas funcionalidades y mejoras en el sitio web.

#### Consideraciones adicionales:

**Estrategias:** La estrategia del proyecto debe estar alineada con los objetivos de negocio y la audiencia objetivo.

**Arquitectura:** La arquitectura del sitio web debe ser robusta, escalable y segura.

**Contenido:** El contenido del sitio web debe ser relevante, atractivo y optimizado para la experiencia del usuario.

**Herramientas:** Se utilizan diversas herramientas para el desarrollo web, incluyendo editores de código, sistemas de gestión de contenido (CMS), frameworks y plataformas de pruebas.

## Arquitectura Web

### Arquitectura de un Proyecto Web

La arquitectura de un proyecto web es la estructuración lógica y física de sus componentes (servidores, bases de datos, frontend, backend) para garantizar escalabilidad, seguridad y rendimiento. Define cómo se organiza la información (mapa del sitio, jerarquía) y cómo interactúan las partes para una mejor experiencia de usuario (UX) y posicionamiento SEO.

La arquitectura de contenidos web consiste en organizar, estructurar y jerarquizar la información dentro de un sitio web o aplicación digital. Su objetivo principal es que los usuarios encuentren fácilmente lo que buscan y que la navegación sea fluida e intuitiva.

En la práctica, la arquitectura de contenidos web consiste, a grandes rasgos, en definir los bloques de contenido, jerarquizar la información, diseñar la navegación, pensar en la experiencia del usuario (UX) y en la optimización para los buscadores (es decir, el SEO).

Digamos que es como el mapa o esqueleto de un proyecto digital. Igual que en la construcción de una casa no se empieza por decorar las paredes sin haber definido antes la distribución de

las habitaciones, en una web tampoco conviene diseñar sin tener claro cómo se organizará la información. Por eso, la implementación de la arquitectura de contenidos es uno de los primeros elementos a tener en cuenta en el desarrollo de un negocio digital.

Si te encuentras en esta fase, en este artículo te contamos cómo llevar a cabo la arquitectura de contenidos web según vayas a desarrollar un e-commerce, una página o una aplicación, pues el proceso es diferente. Y también te contamos una de las herramientas más interesantes para poder desarrollar este trabajo mucho mejor.

### ¿Cómo plantear la arquitectura de contenidos?

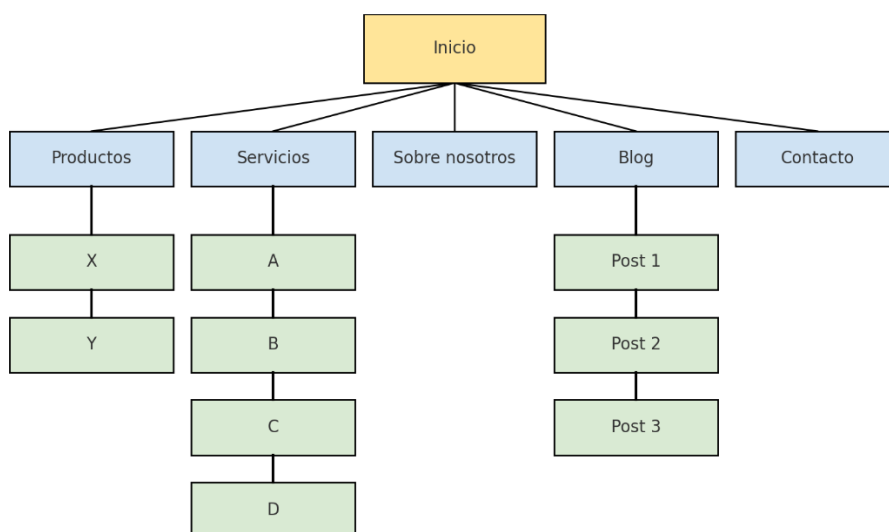
En el caso de una página web corporativa, institucional o de una marca personal, la arquitectura de contenidos debe centrarse en transmitir con claridad quiénes somos, qué hacemos y cómo podemos ayudar al usuario. Para ello, el primer paso es definir los apartados principales del menú de navegación, que suelen incluir secciones como “Inicio, Quienes somos, Servicios y/o Contacto”.

A partir de estos bloques, se crean los sub-apartados que profundizan en cada tema. Por ejemplo, dentro de los “Servicios” se pueden detallar los distintos ámbitos de actuación. Además, en cada página se deben planificar los contenidos específicos, teniendo en cuenta su objetivo: informar, generar confianza, captar un lead o facilitar el contacto.

Más allá de esta estructura básica, también hay que considerar la localización de una serie de elementos complementarios que, aunque no formen parte del menú principal, son indispensables para cualquier página web profesional:

- Textos legales (aviso legal, política de privacidad, política de cookies...).
- Cambio de idioma, en el caso de webs multilingües. Lo cual es uno de los factores clave en un proyecto digital y que más se suelen pasar por alto.
- Enlaces a redes sociales, para reforzar la presencia digital de la marca.
- Suscripción a una newsletter u otro mecanismo de captación de leads.

El conjunto de apartados, sub-apartados y elementos adicionales conforman lo que se conoce como mapa web, una representación (normalmente en forma de esquema) que facilita la visión global de la estructura y sirve de guía para diseñadores, desarrolladores y responsables de contenidos.



## Mapa Web

### Mapa del Sitio Web

Un mapa de sitio web (o mapa de sitio o mapa web) es una lista de las páginas de un sitio web accesibles por parte de los buscadores y los usuarios. Puede ser tanto un documento en cualquier formato usado como herramienta de planificación para el diseño de una web como una página que lista las páginas de una web (ya realizada), organizadas comúnmente de forma jerárquica. Esto ayuda a los visitantes y a los bots de los motores de búsqueda a hallar las páginas de un sitio web.

Los mapas de sitio pueden mejorar el posicionamiento en buscadores de un sitio, asegurándose que todas sus páginas puedan ser encontradas. Esto es especialmente importante si el sitio usa menús Adobe Flash o JavaScript que incluyan enlaces HTML.

También son una ayuda a la navegación por ofrecer una vista general del contenido de un sitio de un simple vistazo.

La mayoría de los motores de búsqueda sólo siguen un finito número de enlaces desde una página, por lo que si el sitio es muy grande la existencia del mapa del sitio puede ser necesaria para que tanto los motores de búsqueda como los usuarios accedan a todo el contenido del sitio.

Algunos desarrolladores afirman que índice de sitio web sería un término más apropiado, pero esta forma se usa mucho menos en español. A veces los "índices de sitio" muestran enlaces en orden alfabético hacia las páginas (o partes de ellas) de los sitios web, por lo que en estos casos pueden ser un complemento a los mapas de sitio (enfocados a las páginas de forma jerárquica, temática).



## Ventajas y Técnicas de la Construcción Web

La construcción web ofrece varias ventajas, como visibilidad, accesibilidad global y disponibilidad 24/7.

Las técnicas de construcción web incluyen la arquitectura web, que se enfoca en la organización y estructura de los contenidos para mejorar la experiencia del usuario y el posicionamiento SEO.

### Ventajas:

**Mayor visibilidad y alcance:** Un sitio web permite que un negocio o perfil personal sea accesible desde cualquier parte del mundo, aumentando la visibilidad y el alcance de la marca.

**Disponibilidad 24/7:** Los sitios web están disponibles las 24 horas del día, los 7 días de la semana, lo que permite a los usuarios acceder a la información en cualquier momento.

**Herramienta de promoción:** Un sitio web es una herramienta eficaz para promocionar productos y servicios, atraer clientes potenciales y generar ventas.

**Facilita la comunicación:** Permite la comunicación con los clientes, la recopilación de reseñas y la construcción de confianza.

### Técnicas:

**Arquitectura web:** La planificación y diseño de los componentes técnicos, funcionales y visuales de un sitio web, antes de su desarrollo e implementación, para asegurar una buena experiencia de usuario y optimizar el posicionamiento SEO.

**Diseño web:** La creación de la interfaz de usuario (UI) y la experiencia de usuario (UX) de un sitio web, incluyendo la disposición visual, la navegación y la interactividad.

**Desarrollo web:** La creación del código de la página web, que incluye HTML, CSS, JavaScript y otros lenguajes de programación, para hacerla funcional y dinámica.

**Optimización de motores de búsqueda (SEO):** La aplicación de estrategias para mejorar el posicionamiento de un sitio web en los resultados de los motores de búsqueda, como Google.

**Arquitectura de información:** La organización de los contenidos de un sitio web de manera lógica y jerárquica, para facilitar la navegación y la búsqueda de información.

**Desarrollo de aplicaciones web:** La creación de aplicaciones web que se ejecutan en un navegador web, ofreciendo funcionalidades similares a las aplicaciones nativas.

**Tecnología PWA (Progressive Web Apps):** La implementación de PWA, que permite crear aplicaciones web con características de aplicaciones nativas, como el acceso sin conexión y las notificaciones push.

**Construcción en la nube:** El uso de software de construcción basado en la nube, que ofrece flexibilidad, escalabilidad y acceso a las herramientas desde cualquier lugar.

**Técnicas de programación de construcción:** La utilización de técnicas de programación, como CPM (Critical Path Method), para la gestión de proyectos de construcción y la optimización de los plazos.

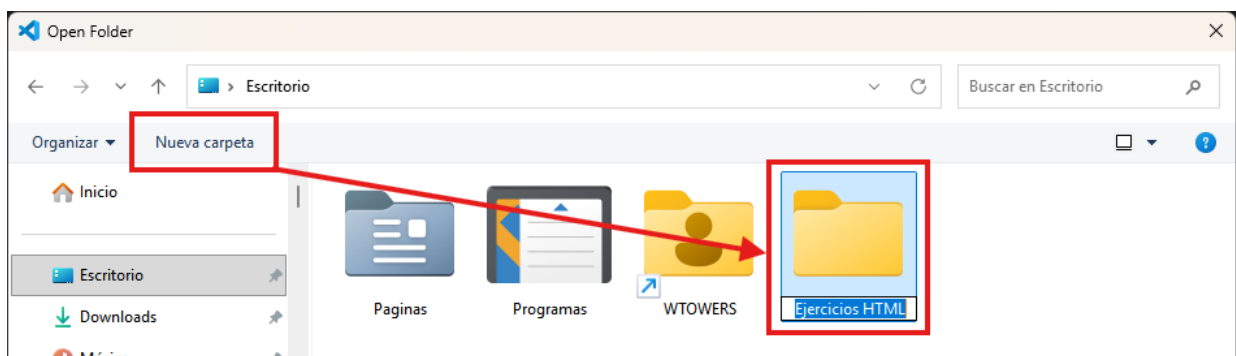
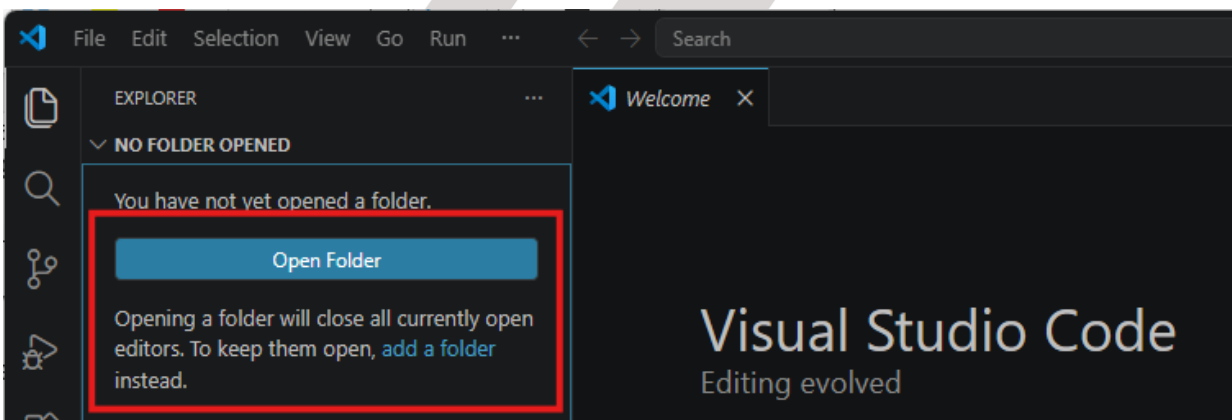
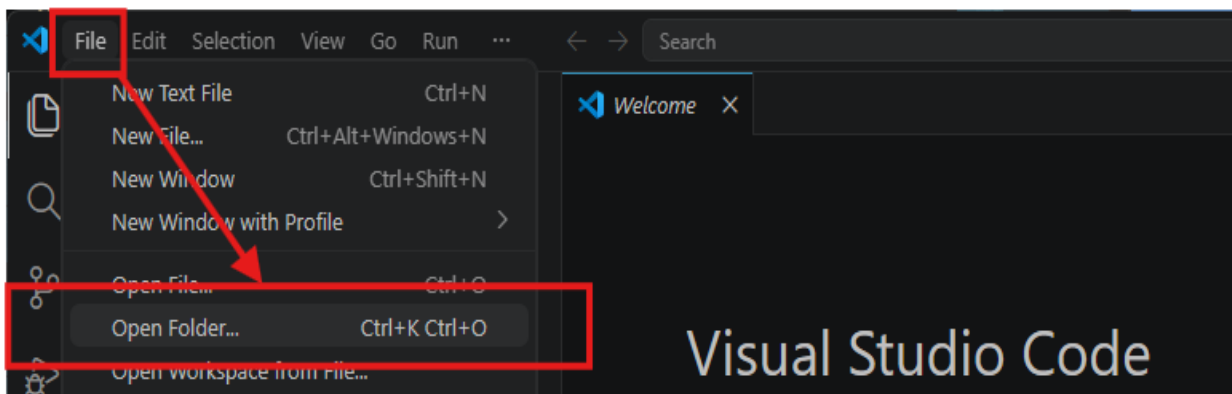


## ACTIVIDAD 01

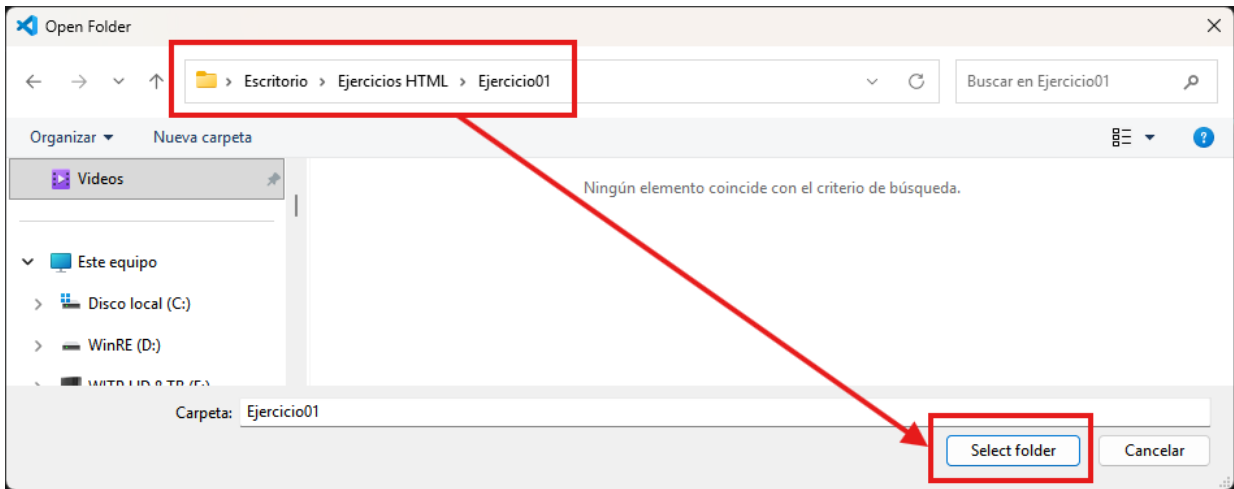
<b>LABORATORIO 01</b>	HTML Estructura de una Pagina
<b>TEMA</b>	HTML
<b>RECURSOS</b>	Visual Studio Code
<b>OBJETIVO</b>	Guiar al alumno en los principios de la programación HTML y la estructura básica para la creación de documentos Web

### Proceso

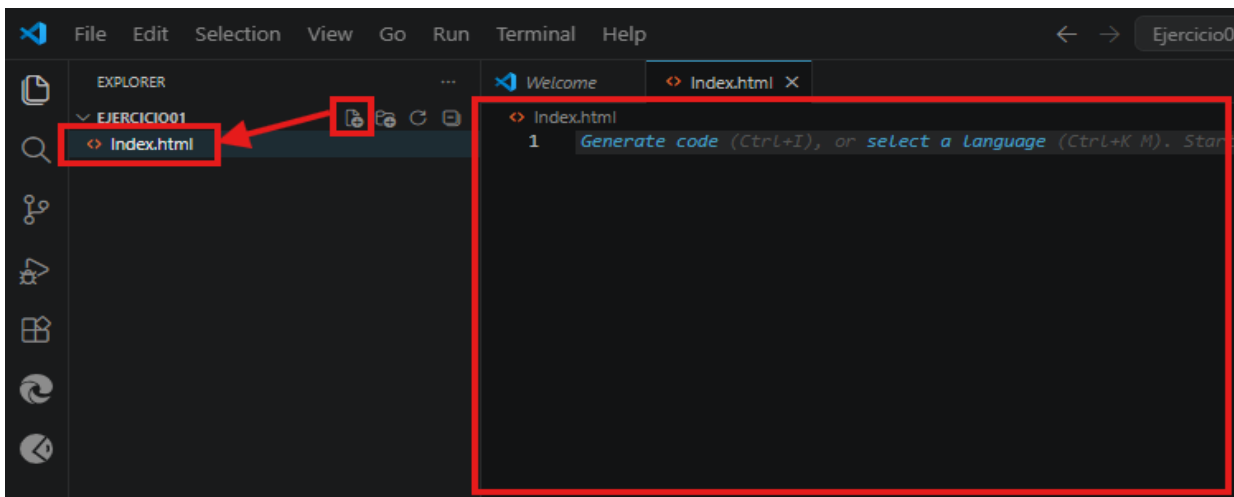
**Paso 01:** Primero abriremos Visual Studio Code y crearemos un nuevo proyecto seleccionando una Carpeta en una ubicación específica. Esto lo puedes hacer desde el menú File (Archivo) o también desde el panel Explorer (Explorador), en el botón Open Folder (Abrir Carpeta) y seguidamente crearemos una carpeta de nombre Ejercicios HTML en una ubicación específica.



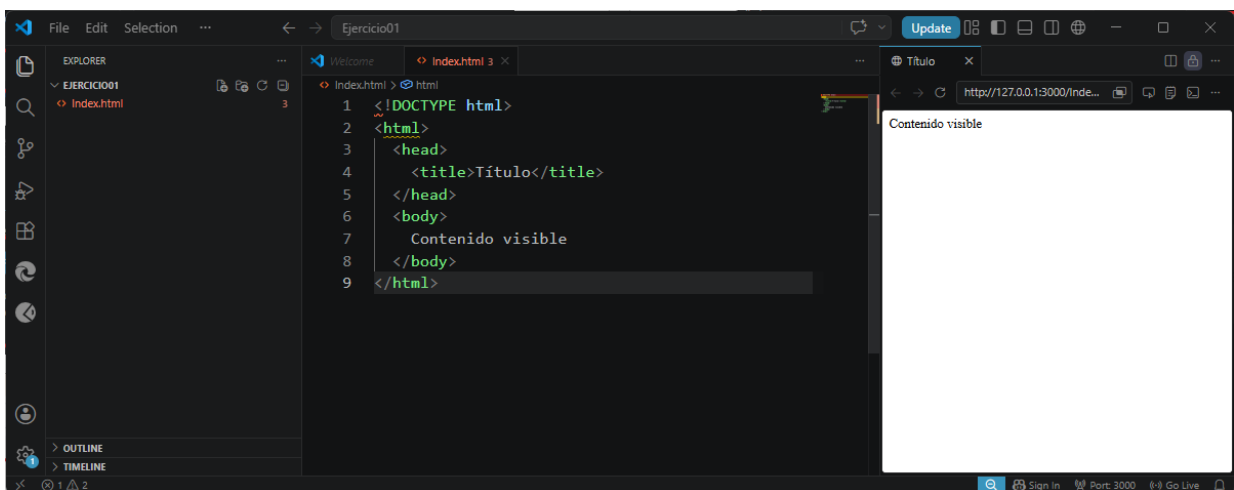
**Paso 02:** Ingresaremos a la carpeta creada y crearemos otra carpeta con el nombre Ejercicio01, ingresaremos en esa carpeta y presionaremos el botón **Select Folder**.



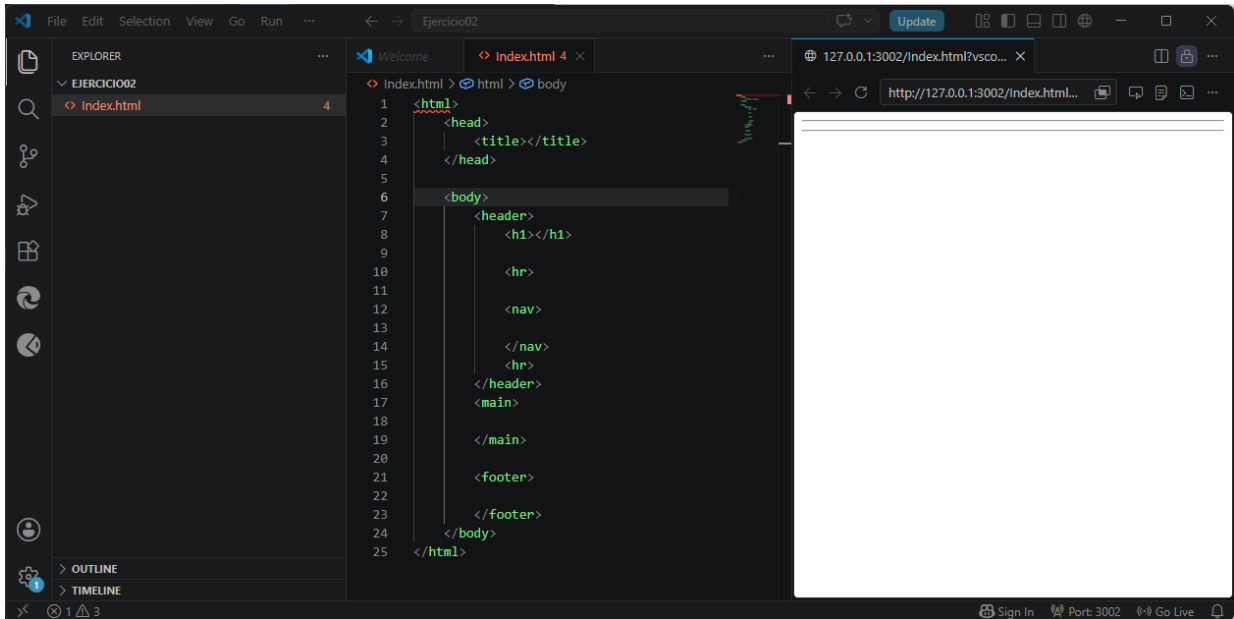
**Paso 03:** Creamos el archivo **Index.html** dentro de la carpeta del proyecto.



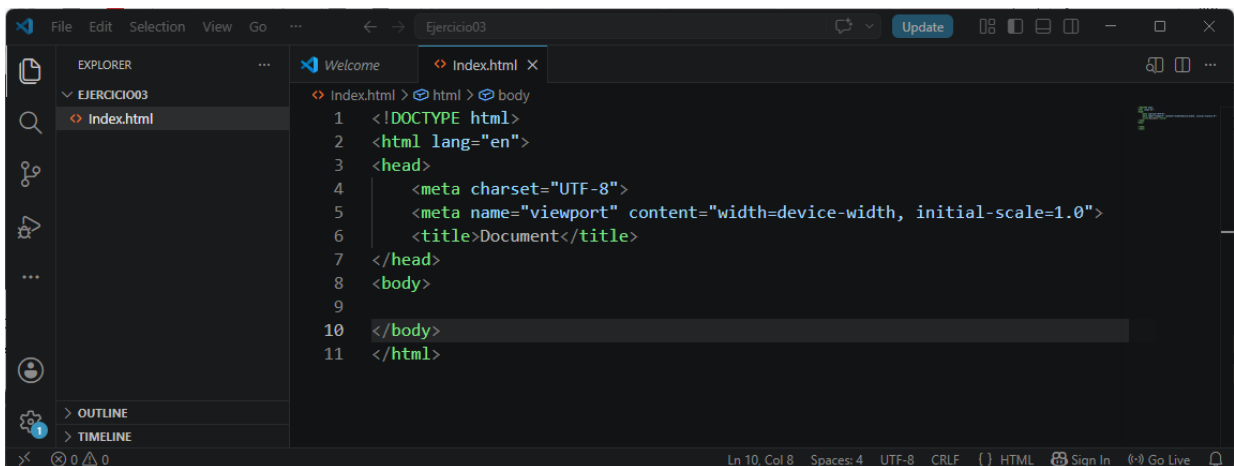
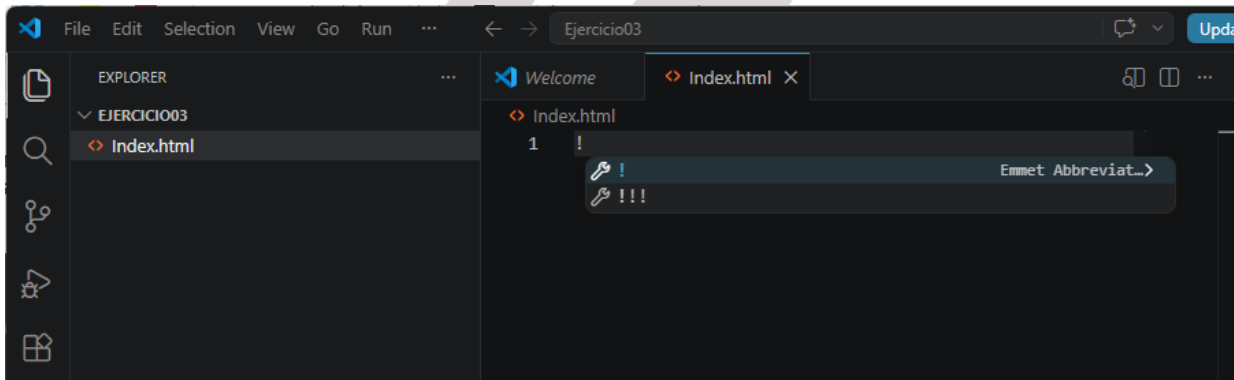
**Paso 03:** Copiaremos el código HTML correspondiente para nuestro ejemplo y el cual se puede ver en la imagen que vemos a continuación (podemos obtenerlo desde la siguiente ubicación <https://github.com/wildtowalls/EstructuraBasicaHTML.git>)



**Paso 04:** Ahora crearemos dentro de la carpeta **Ejercicios HTML** una carpeta **Ejercicio02** y dentro de ella crearemos un archivo **Index.html**. Seguidamente colocaremos el código alojado en la siguiente ubicación <https://github.com/wildtowalls/EstructuraHTML.git>



**Paso 05:** Por último, crearemos una tercera carpeta dentro de **Ejercicios HTML** y la llamaremos **Ejercicio03**, ¡en su interior crearemos un archivo **Index.html** e introduciremos solamente el carácter **!** e inmediatamente presionaremos la tecla enter, veremos entonces que el código base se imprime inmediatamente en el editor.



## ACTIVIDAD 02

<b>LABORATORIO 02</b>	Etiquetas Básicas
<b>TEMA</b>	HTML
<b>RECURSOS</b>	Visual Studio Code
<b>OBJETIVO</b>	Guiar al alumno en creación de páginas básicas usando las etiquetas principales HTML.

### Proceso

**Paso 01:** Como ya conocemos el proceso de creación de proyecto, esta vez crearemos dentro de la carpeta creada en la **actividad 01** una carpeta **Ejercicio04** y en su interior crearemos nuestro ya conocido archivo **Index.html**, y pegaremos el código de la siguiente dirección: <https://github.com/wildtowalls/HTMLTexto.git>

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Actividad con encabezados</title>
6 </head>
7 <body>
8
9   <h1>Guía básica de HTML</h1>
10  Introducción general al tema de la página.
11
12  <h2>¿Qué es HTML?</h2>
13  Explicación breve sobre el lenguaje de marcado para la web.
14
15  <h3>Estructura mínima de un documento</h3>
16  Descripción de las partes principales: html, head y body.
17
18  <h4>La etiqueta head</h4>
19  Información sobre metadatos, título de la página y otros elemento
20
21  <h4>La etiqueta body</h4>
22  Explicación de dónde va el contenido visible de la página.
23

```

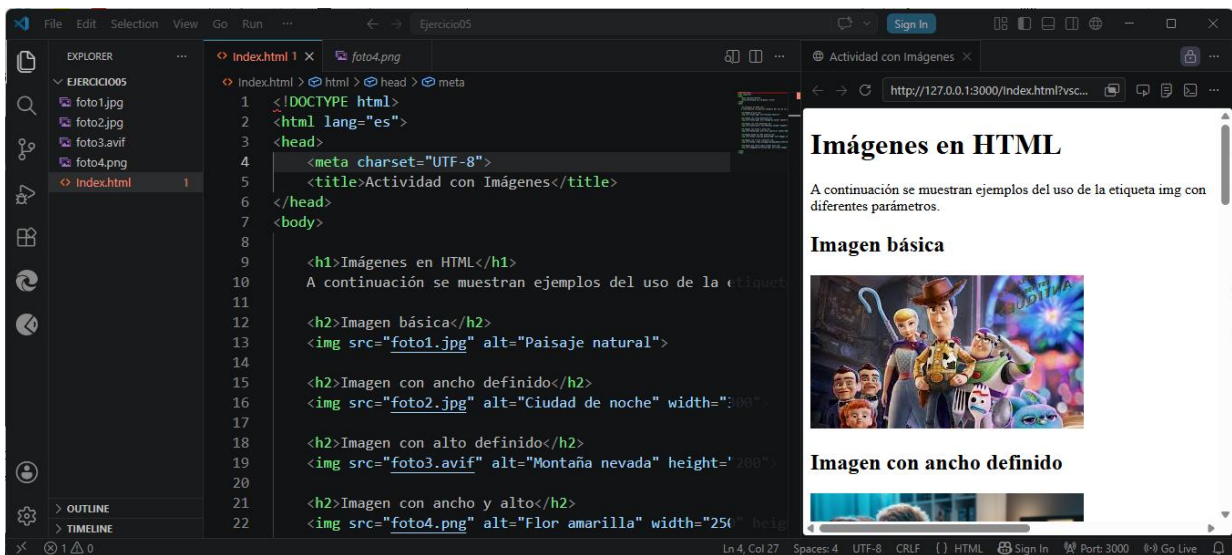
**Paso 02:** Haremos una vista previa de lo realizado

**Paso 03:** Haremos algunos cambios aplicando parámetros como color de relleno, color de texto, alineaciones, uso de texto de prueba (texto lorem).

**Paso 04:** Deberás cambiar los siguientes elementos:

- Cambiar los textos de cada encabezado para otro tema (por ejemplo, “Fútbol”, “Programación en Python”, etc.) respetando la jerarquía.
- Agrega nuevas secciones usando solo h2, h3 y texto, sin otras etiquetas.
- Modifica los parámetros empleados indicados por el profesor.

**Paso 05:** Ahora crearemos dentro de la carpeta **Ejercicios HTML** una carpeta **Ejercicio05** y dentro de ella crearemos un archivo **Index.html**. Seguidamente colocaremos el código alojado en la siguiente ubicación <https://github.com/wildtowalls/HTMLImagenes.git>

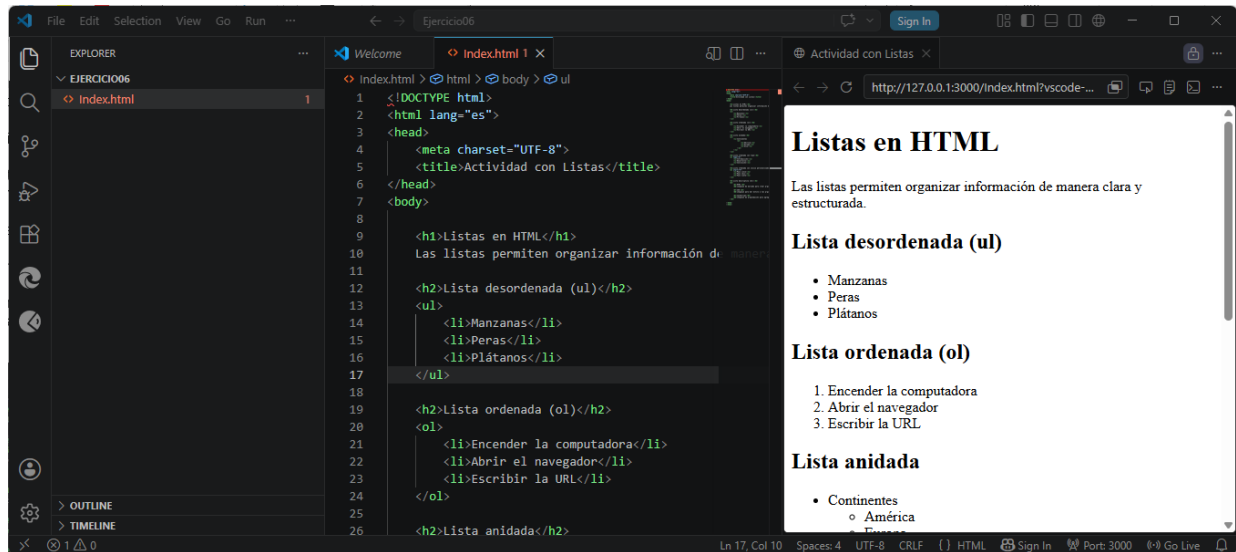


**Paso 06:** Haremos algunos cambios aplicando parámetros.

**Paso 07:** Deberás cambiar los siguientes elementos:

- Reemplazar imágenes por otras de su elección (locales o URLs).
- Modifica los parámetros **width**, **height** y **title** para ver cómo cambian los resultados.
- Agrega tres imágenes nuevas:
- Una con solo **src** y **alt**.
- Una con tamaño fijo.
- Una desde un enlace externo.
- Toma nota sobre qué ocurre cuando la ruta es incorrecta.

**Paso 08:** Ahora crearemos dentro de la carpeta **Ejercicios HTML** una carpeta **Ejercicio06** y dentro de ella crearemos un archivo `Index.html`. Seguidamente colocaremos el código alojado en la siguiente ubicación <https://github.com/wildtowalls/HTMLListas.git>

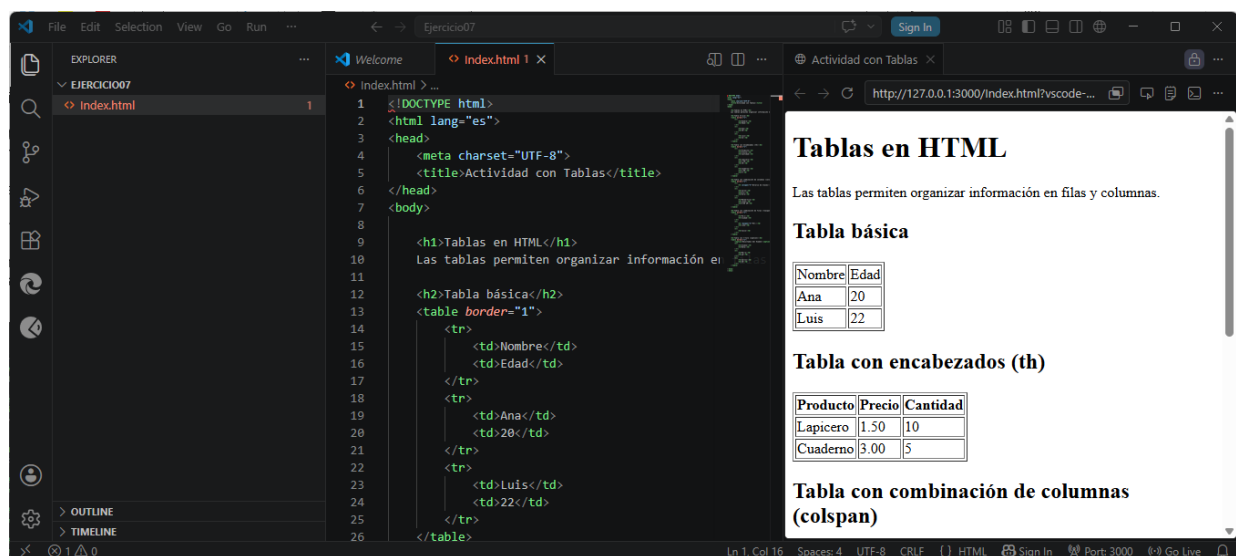


**Paso 09:** Haremos algunos cambios aplicando parámetros.

**Paso 10:** Deberás cambiar los siguientes elementos:

- Modifica los elementos de cada lista por contenido propio.
- Agreguen una lista anidada con al menos dos niveles.
- Creen una lista ordenada que empiece en un número distinto a 1.
- Usen los tipos de listas: `type="A"`, `type="a"`, `type="I"`, `type="i"`.
- Incluyan una lista descriptiva con tres términos y sus definiciones.

**Paso 11:** Ahora crearemos dentro de la carpeta **Ejercicios HTML** una carpeta **Ejercicio07** y dentro de ella crearemos un archivo `Index.html`. Seguidamente colocaremos el código alojado en la siguiente ubicación <https://github.com/wildtowalls/HTMLTablas.git>

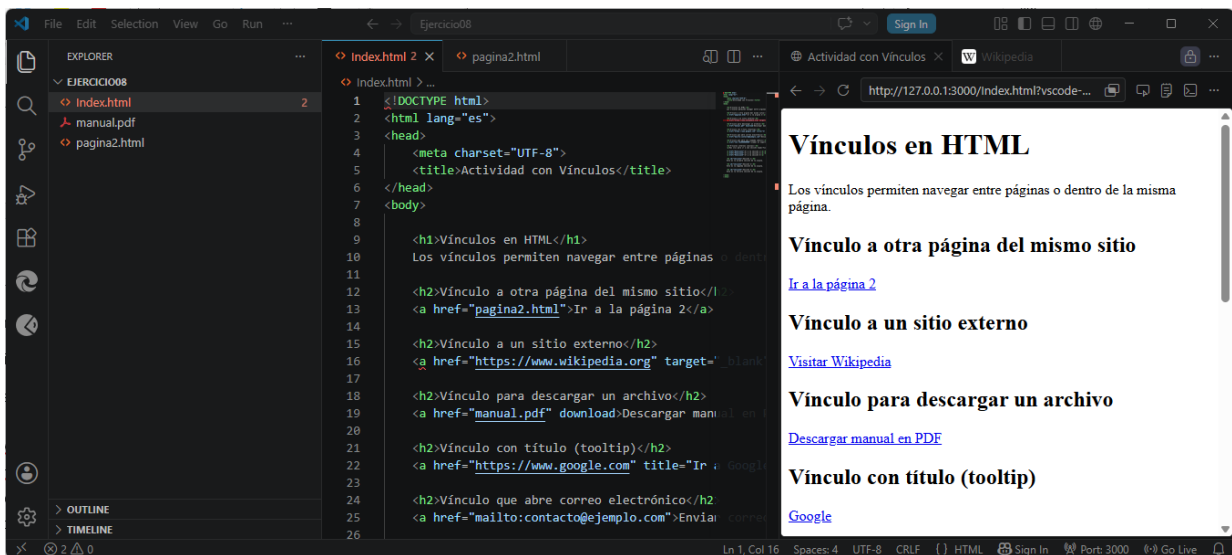


**Paso 12:** Haremos algunos cambios aplicando parámetros.

**Paso 13:** Deberás cambiar los siguientes elementos:

- Modifica los datos de cada tabla con información propia.
- Agrega una tabla nueva con al menos 4 columnas y 5 filas.
- Usa colspan y rowspan en una tabla creada por ellos.
- Incluye un caption que describa el contenido de la tabla.
- Identifica visualmente (en papel o en el código) qué partes corresponden a filas, columnas y encabezados.

**Paso 14:** Ahora crearemos dentro de la carpeta **Ejercicios HTML** una carpeta **Ejercicio08** y dentro de ella crearemos un archivo `Index.html`. Seguidamente colocaremos el código alojado en la siguiente ubicación <https://github.com/wildtowalls/HTMLHyperLinks.git>

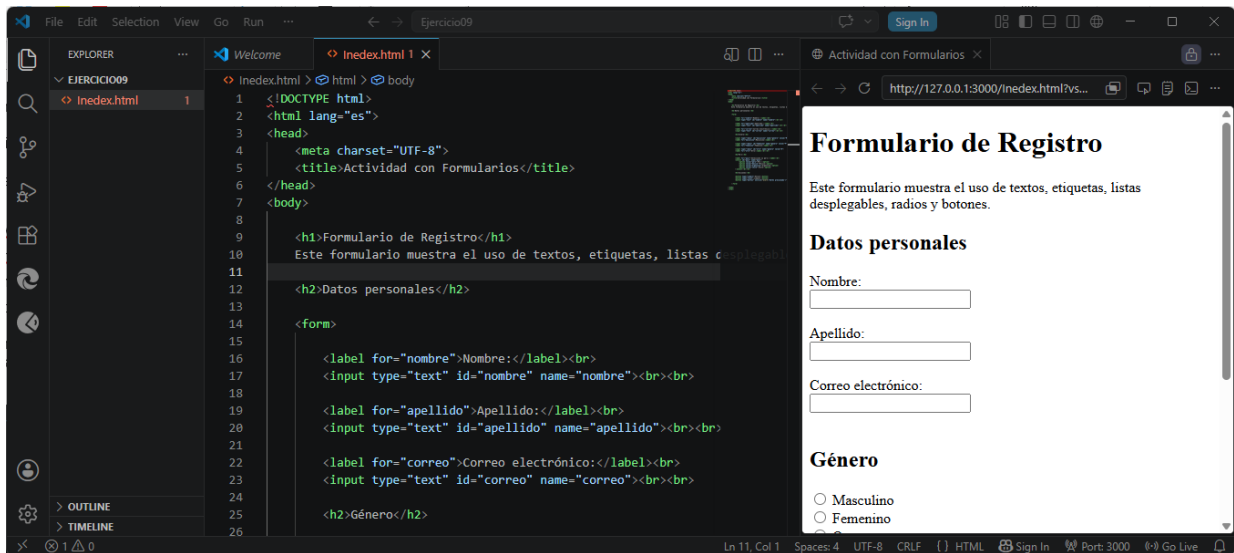


**Paso 15:** Haremos algunos cambios aplicando parámetros.

**Paso 16:** Deberás cambiar los siguientes elementos:

- Cambia los enlaces externos por sitios que ellos visiten frecuentemente.
- Crea una segunda página (pagina2.html) y modifiquen el enlace interno para que funcione.
- Agrega tres anclas nuevas dentro de la misma página.
- Incluye un enlace de descarga que apunte a un archivo real dentro de su proyecto.
- Usa `target="_blank"` en algunos enlaces y expliquen qué hace.
- Agrega un enlace de correo y uno de teléfono con datos ficticios.

**Paso 17:** Ahora crearemos dentro de la carpeta **Ejercicios HTML** una carpeta **Ejercicio09** y dentro de ella crearemos un archivo `Index.html`. Seguidamente colocaremos el código alojado en la siguiente ubicación <https://github.com/wildtowalls/HTMLForms.git>



**Paso 18:** Haremos algunos cambios aplicando parámetros.

**Paso 19:** Deberás cambiar los siguientes elementos:

- Agrega dos campos de texto más, por ejemplo: “Dirección” y “Teléfono”.
- Modifica el combobox para incluir más países o categorías diferentes.
- Agrega un grupo de radio nuevo, por ejemplo: “Estado civil”.
- Cambia el botón simple para que muestre otro mensaje.
- Creen un segundo formulario con otros datos (por ejemplo, encuesta de satisfacción).

## ACTIVIDAD 03

<b>LABORATORIO 03</b>	Creación de la estructura de un Sitio Web
<b>TEMA</b>	HTML
<b>RECURSOS</b>	Visual Studio Code
<b>OBJETIVO</b>	Guiar al alumno en la creación de una Sitio Web y la estructura del sitio.

### Proceso

**Paso 01:** Esta vez crearemos desde una ubicación específica una carpeta de nombre **Mi Sitio**

**Paso 02:** Dentro de la carpeta del paso anterior crearemos la siguiente estructura (solo las carpetas):

```

mi_sitio_web/
├── index.html
├── paginas/
│   ├── acerca.html
│   └── contacto.html
├── imagenes/
│   ├── logo.png
│   └── foto1.jpg
└── documentos/
    └── guia.pdf
    
```

**Paso 03:** Ahora crearemos los archivos **index.html**, **Acerca.html** y **Contacto.html** dentro de cada carpeta según el esquema indicado en el paso anterior, luego descargaremos el código de cada página desde la siguiente ubicación: <https://github.com/wildtowalls/SitioEjemplo.git>

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Inicio</title>
</head>
<body>

  <h1>Página principal</h1>

  <h2>Menú de navegación</h2>
  <a href="paginas/acerca.html">Acerca de</a> |
  <a href="paginas/contacto.html">Contacto</a>

  <h2>Imagen desde la carpeta</h2>
  

  <h2>Descarga de documento</h2>
  <a href="documentos/guia.pdf" download>Descargar guía en PDF</a>

</body>
</html>
    
```

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Acerca de</title>
</head>
<body>

  <h1>Acerca de este sitio</h1>
  <p>Este sitio fue creado como parte de una actividad de práctica.</p>

  <a href="../index.html">Volver al inicio</a>

</body>
</html>

```

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Contacto</title>
</head>
<body>

  <h1>Contacto</h1>

  <form>

    <label for="nombre">Nombre:</label><br>
    <input type="text" id="nombre"><br><br>

    <label for="mensaje">Mensaje:</label><br>
    <input type="text" id="mensaje"><br><br>

    <button type="submit">Enviar</button>








  </form>

  <br>
  <a href="../index.html">Volver al inicio</a>

</body>
</html>

```

**Paso 04:** A continuación, descargaremos de la misma ubicación del paso anterior los archivos necesarios y los colocaremos en las carpetas respectivas.

 wildtowalls	Delete foto.jpg	8f00ed5 · 6 minutes ago	🔄 7 Commits
	Acerca.html	Create Acerca.html	13 minutes ago
	Contacto.html	Create Contacto.html	13 minutes ago
	Index.html	Create Index.html	14 minutes ago
	foto1.jpg	Add files via upload	7 minutes ago
	guia.pdf	Add files via upload	12 minutes ago
	logo.png	Add files via upload	7 minutes ago

**Paso 05:** Probaremos el sitio y los vínculos respectivamente.

**Paso 06:** Deberás agregar los siguientes elementos:

- Crear una nueva página dentro de /paginas llamada **Servicios.html**.
- Agregar una nueva carpeta llamada /extras y dentro un archivo vacío llamado nota.txt.
- Mover **foto1.jpg** a una nueva carpeta llamada **/imagenes/galeria** y actualizar las rutas.
- Agrega 4 imágenes en la página servicios. Las imágenes deberán estar en la ruta **/imagenes/servicios**.

**Paso 07:** Sube tu sitio a tu servicio reemplazando el archivo que subiste en la actividad de la semana 01



## ACTIVIDAD SEMANA

- Desarrolla un Sitio Web usando una estructura como se desarrolló en clase, pero emplea más páginas (unas 5 páginas debidamente enlazadas).
- Emplea en el diseño de cada página elementos de etiquetas usadas en clase (textos, títulos, listas, tablas, vínculos, imágenes y formularios)
- Recuerda usar únicamente HTML en el desarrollo de tu actividad.
- Agrega a tu proyecto web imágenes y documentos que puedas descargar.
- Puedes guiarte de alguna página existente e imitarla no en su totalidad recordando no usar mas que HTML



## REFERENCIAS

- Mozilla Contributors. (2023). HTML: HyperText Markup Language. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTML>
- Google Developers. (2023). Chrome DevTools Overview. Google. <https://developer.chrome.com/docs/devtools/>
- W3Schools. (2023). HTML Tutorial. <https://www.w3schools.com/html/>
- Smashing Magazine. (2022). The Complete Guide To Planning Your Web Project. <https://www.smashingmagazine.com/2022/01/complete-guide-planning-web-project/>
- Nielsen Norman Group. (2021). Information Architecture Basics. <https://www.nngroup.com/articles/information-architecture-basics/>
- FreeCodeCamp.org. (2023). Responsive Web Design Principles. <https://www.freecodecamp.org/learn/responsive-web-design/>
- Bootstrap Contributors. (2023). Introduction to Bootstrap. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- GitHub Docs. (2023). About GitHub. <https://docs.github.com/en/get-started/quickstart/about-github>
- Google Search Central. (2023). Create a Sitemap. <https://developers.google.com/search/docs/advanced/sitemaps/overview>
- W3C. (2023). HTML and CSS Validation. <https://validator.w3.org/>

