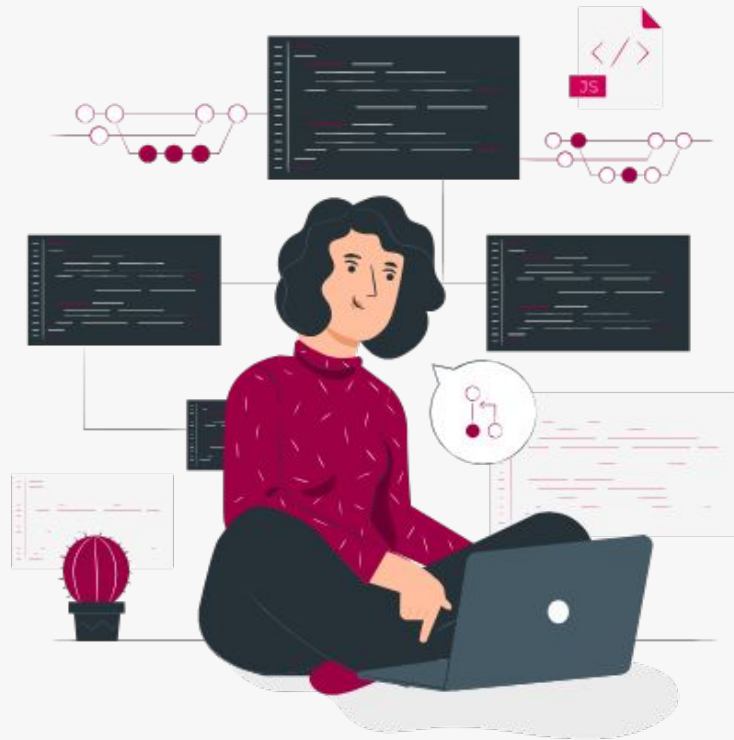


# BIENVENIDAS !



@Ada

## Un Poquito de Historia...

Somos una ONG que inició formalmente su labor en el 2017.

Creemos que la tecnología tiene un alto **poder transformador** y buscamos generar **oportunidades para todas las mujeres** alrededor del mundo, gracias a nuestra modalidad online.

Somos **la primera organización en Latinoamérica** que cumple con transparencia de datos en gestión de los cursos y los indicadores que informamos, gracias a la auditoría del Council on Integrity in Results Reporting (CIRR).

## Nuestra misión...

Llevamos más de 7 años formando a mujeres de toda la región en programación con perfiles **trainee/junior** para iniciar a trabajar en la industria tecnológica.

Nuestra propuesta se basa en **capacitar a mujeres y feminidades en tecnología**, y acompañar durante su proceso de inserción en el mercado de trabajo IT

Nos destacamos por formar a mujeres que, en su mayoría, presentan perfiles:

- Sin educación universitaria
- Sin empleo formal
- En alguna situación de vulnerabilidad socioeconómica

## Sobre nosotr@s...

En ADA, somos un equipo de 5 personas llenas de entusiasmo y dedicación, trabajando con el corazón ❤️ para ofrecerles la mejor experiencia posible.

Además, contamos con un increíble equipo de docentes y tutoras, siempre dispuestos a acompañarlas en su proceso de aprendizaje.

Queremos que sepan que nunca están solas en este camino. Si tienen dudas, necesitan apoyo o simplemente un consejo, no duden en pedir ayuda 🙋.



# Programa de la Carrera



## Estructura de la Carrera de Front End



Curso 1	Introduccion a la programacion con Javascript
Curso 2	Introduccion a Front End: Dominio del Maquetado Web (HTML + CSS)
Curso 3	Desarrollo Frontend Avanzado: Trabajando con APIs REST y Programación Asíncrona
Curso 4	Frontend Avanzado con React: Dominio de Componentes, Estado, Hooks y Estilos

La carrera tiene un tiempo de duracion de 9 meses, que abarcan clases teorico-practicas. Este tiempo no contempla el lapso de duracion entre Trabajos Practicos integradores.

Los trabajos prácticos integradores van variando, ya que desde ADA los vamos adaptando según las demandas del mundo laboral IT

Horarios y Días de cursada:  
**Lunes y Miércoles de 19 a 22 hs.**  
(Horario de **Argentina**)

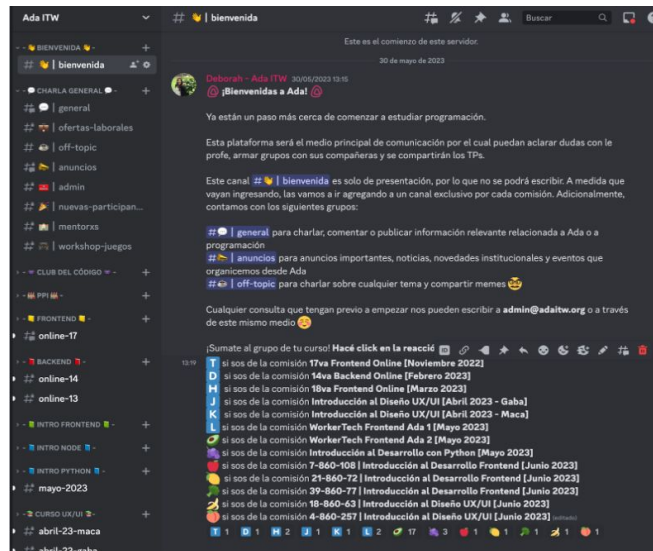
# Asistencia

## RESPECTO A LA CARGA DE HORAS:

- ☐ Debes ingresar a las clases en zoom con el Nombre, Apellido y Email con el que te inscribiste.
- ☐ Si tus horas no figuran correctamente, no te preocupes que en nuestra base interna sí figuran.
- ☐ En caso de cursar “Diferido” anotar su asistencia en Excel (El link será enviado por Discord)

# Discord

Es importante que en la primera sección llamada “Bienvenida”, des “click” en el emoji de nuestra carrera:



# Fonselp

Es la plataforma que utilizaremos en la carrera, allí encontrarán:

- ☐ El material/guía para sus clases,
- ☐ El link de zoom (mismo que les llegó a sus mails y se va a utilizar en todas las clases).
- ☐ Días y horarios de clases.
- ☐ Manual de alumnas (para despejar dudas sobre la cursada)
- ☐ Link de alta de casos por si tienen algún inconveniente o realizar alguna consulta
- ☐ Las clases grabadas
- ☐ El certificado, una vez finalizado y aprobado el curso/carrera.



## Grabaciones de clases

Las grabaciones se subirán de forma automática al Aula Virtual en el transcurso de **24/48 hs** y las tendrán siempre a disposición.

## Encuestas de satisfacción

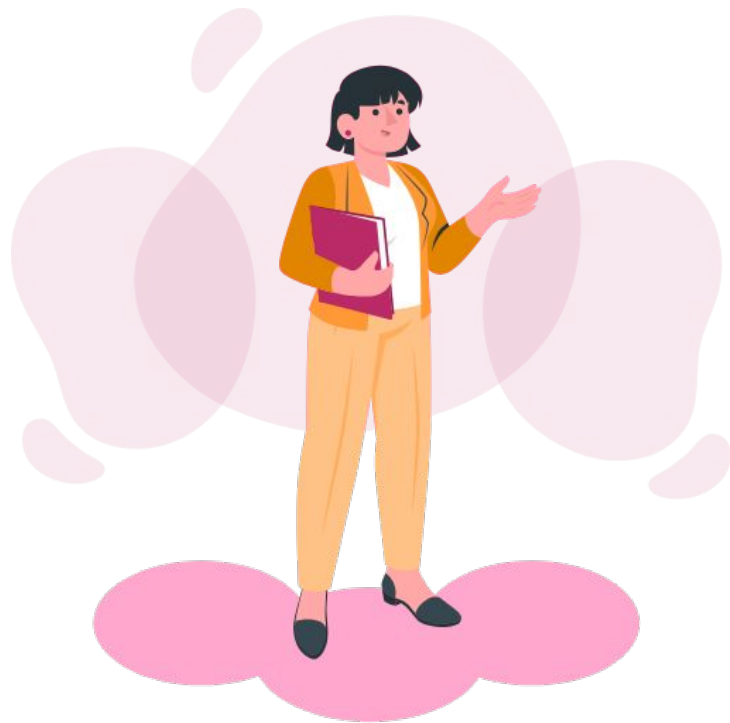
Se enviarán encuestas cada **15 días al mail**.

Este es el modo de evaluar su bienestar por ello es importante que nos compartan sus observaciones.

# Clase 1:

# Introducción a

# HTML



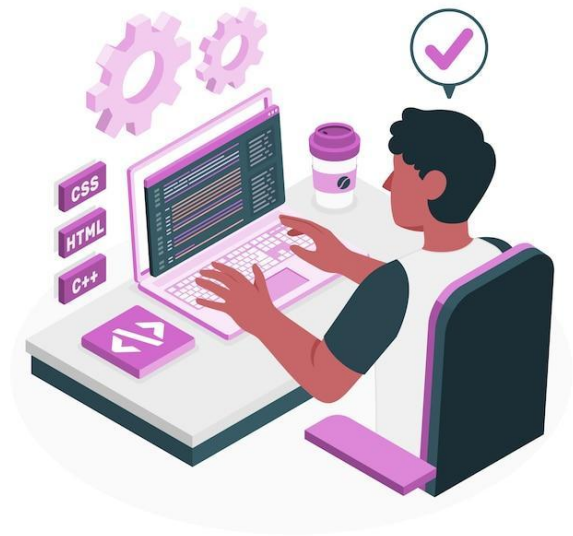
# Explicación del rol “Front – End”



# ¿Qué es ser un desarrollador front-end?

Un desarrollador front-end es la persona encargada de **construir la parte visual e interactiva de un sitio web o aplicación.**

Esto incluye todo lo que los usuarios ven y con lo que interactúan directamente, como botones, imágenes, formularios, menús y mucho más.



## Funciones principales:

- **Estructura:** Crear la estructura de una página web usando HTML.
- **Estilo:** Dar diseño y apariencia atractiva a esa estructura mediante CSS.
- **Interactividad:** Añadir funciones dinámicas usando JavaScript para que la página sea más interactiva (como animaciones, formularios interactivos, etc.).

## Habilidades clave:

- Dominio de HTML, CSS y JavaScript.
- Capacidad para trabajar con herramientas como navegadores, editores de código y control de versiones (Git).
- Comprender la experiencia de usuario (UX) y el diseño centrado en el usuario.

A decorative border of various tech-related icons in shades of blue, green, and orange surrounds the central text. The icons include a code editor window, a cloud, a USB drive, a Wi-Fi symbol, a smartphone, a camera, a mouse, a monitor, a folder, and a keyboard.

**¿Qué es HTML?**

# ¿Qué es HTML?

HTML (HyperText Markup Language) es el **lenguaje de marcado estándar para crear páginas web**.

Su función principal es describir la **estructura y el contenido** de un sitio web. Es como el "esqueleto" de cualquier página web.

## Importancia:

- Es esencial para todos los sitios web.
- Sirve para organizar el contenido en bloques que pueden incluir texto, imágenes, videos, tablas, enlaces, etc.

# HTML



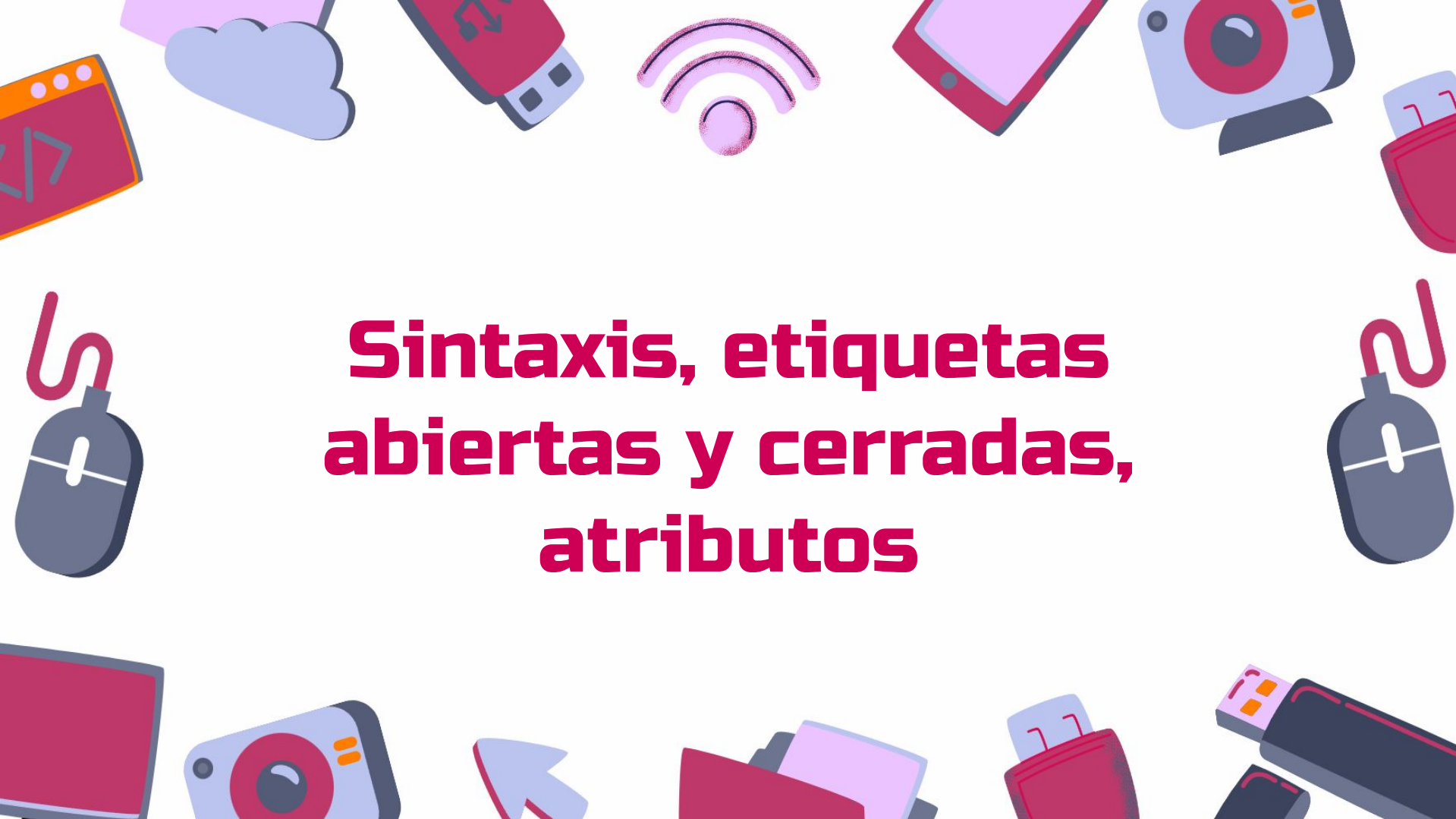
# Ejemplo Sencillo

A small icon representing code, consisting of the characters '</>' in a pink color.

```
<p>¡Hola, mundo!</p>
```

Este código muestra una línea de texto en la página web.  
No te preocupes si no entiendes el como ni porque, ahora vamos a eso!



A decorative border of various tech-related icons in shades of blue, green, and orange, including a cloud, USB drives, a Wi-Fi symbol, a smartphone, a camera, a mouse, a folder, and a monitor, surrounding the central text.

# Sintaxis, etiquetas abiertas y cerradas, atributos

# Sintaxis básica de HTML

La sintaxis en HTML es el conjunto de reglas que define cómo deben estructurarse los elementos para que los navegadores los interpreten correctamente.

## Componentes básicos:

1. **Etiquetas:** Representan los elementos de la página web, como títulos, párrafos, imágenes, etc. Siempre van entre los símbolos < >.
2. **Contenido:** Es el texto o elementos que se muestran en la página web y que se colocan entre las etiquetas de apertura y cierre.
3. **Atributos:** Proveen información adicional sobre un elemento HTML. Siempre están dentro de la etiqueta de apertura.

# Estructura de un tag de HTML (etiqueta)

Una etiqueta actúa como un contenedor y describe los datos o información que contienen entre la etiqueta de apertura y la de cierre.

```
<tipoDeEtiqueta>Contenido</tipoDeEtiqueta>
```

# Una etiqueta esta compuesta por:

## 1. Etiqueta de apertura:

- `<` un signo menor que se conoce como apertura de la etiqueta
- `tipoDeEtiqueta` puede ser una o varias letras que describen el tipo de etiqueta que queremos crear (título, link, imagen, etc)
- `>` un signo mayor que se conoce como cierre de la etiqueta

```
<tipoDeEtiqueta>
```

# Una etiqueta esta compuesta por:

## 2. Contenido:

- **Contenido de la etiqueta:** puede ser un texto u otro/s elemento/s de HTML

```
<tipoDeEtiqueta>Bienvenidos a HTML!!
```

# Una etiqueta esta compuesta por:

## 3. Etiqueta de cierre:

- Tiene la misma estructura que la etiqueta de apertura
- Se tiene que llamar igual que la estructura de apertura para que el browser entienda que acá termina lo que comenzó con la etiqueta de apertura
- / tiene una barra inclinada para determinar que es una etiqueta de cierre (sino sería igual que la de apertura)

```
</tipoDeEtiqueta>
```

# Una etiqueta esta compuesta por:

## 4. Elemento:


- Podemos diferenciar que las etiquetas son las de apertura y cierre
- Llamamos elemento al conjunto de etiquetas de apertura y cierre que tiene contenido

```
<tipoDeEtiqueta>Bienvenidos a HTML!!</tipoDeEtiqueta>
```

# Una etiqueta esta compuesta por:

## 5. Etiquetas sin contenido:

- Algunas etiquetas no tienen contenido y funcionan como placeholder (un lugar donde vamos a poner algo).
- Este tipo de etiquetas utilizan la apertura y cierre en una sola estructura
- Un ejemplo de este tipo de etiquetas es el contenedor para insertar una imagen.



```
<img />
```



## Ejemplo de un elemento con otro elemento:

En el siguiente ejemplo tenemos una etiqueta (no sabemos de qué tipo) que tiene como contenido un texto (Bienvenidos a HTML!!) y una imagen que por ahora no sabemos cual es.

En este ejemplo vemos que el texto y la imagen tienen un margen izquierdo para visualizar de forma rápida y fácil que son contenido del elemento llamado tipoDeEtiqueta.

Esta técnica se conoce como indentación o indentado y es una buena práctica para evitar tener problema a la hora de armar documentos con estructuras más complejas.

```
<tipoDeEtiqueta>  
  Bienvenidos a HTML!!  
  <img />  
</tipoDeEtiqueta>
```

# Atributos de los elementos

Los elementos en HTML pueden tener atributos que nos sirven para describirlo mejor, agregar funcionalidad o alterar su comportamiento.

- El atributo **id** nos permite identificar de forma única un elemento dentro del documento. (Sólo deberíamos tener un elemento de este tipo por documento, ejemplo: Un título principal con el id: titulo)
- El atributo **class** nos permite establecer a que clase de CSS pertenece un elemento. De esta forma podemos agrupar elementos por la forma que se ven o comportan. (Podemos utilizar la misma clase en varios elementos)

Ejemplo usando ID y class:

```
<tipoDeEtiqueta id="titulo-principal" class="rojo"></tipoDeEtiqueta>
```

A decorative border of various tech-related icons in shades of blue, green, and orange surrounds the central text. The icons include a code editor window, a cloud, a USB drive, a Wi-Fi signal, a smartphone, a camera, a mouse, a monitor, a folder, and a keyboard.

**Estructura inicial:  
(DOCTYPE - html- head  
- title- meta- body)**

# Estructura

Ahora vamos a desglosar cada una de las partes de la estructura inicial de un documento HTML.

## 1. `<!DOCTYPE html>`

El DOCTYPE es una declaración que indica al navegador que el documento sigue las reglas del lenguaje HTML5.

Aunque no es una etiqueta HTML como tal, es esencial para que el navegador interprete correctamente el código.

### Detalles:

- Va siempre en la primera línea del archivo HTML.
- No tiene etiqueta de cierre.
- Su único propósito es garantizar que el navegador use el modo estándar en lugar de un modo más antiguo o "quirks mode".

Es como una "bandera" que le dice al navegador: "Esta página usa las reglas modernas de HTML". Si no lo incluyes, algunos navegadores podrían interpretar el contenido de forma incorrecta.

# Estructura

## 2. `<html>`

- La etiqueta `<html>` es el contenedor principal de todo el contenido HTML.
- Es como la raíz del árbol en el que todo el documento está organizado.

### Detalles:

- Todo el código debe estar dentro de las etiquetas `<html>` y `</html>`.
- Puede incluir atributos, como el idioma del contenido, por ejemplo:  
`<html lang="es">`

# Estructura

## 3. `<head>`

- La etiqueta `<head>` contiene metadatos y configuraciones sobre la página web.
- Estos datos no se ven directamente en la página, pero son esenciales para su correcto funcionamiento.

### ¿Por qué es importante?

- El `<head>` actúa como un "centro de control" que prepara todo lo necesario antes de mostrar el contenido visible de la página.

# Estructura

## 4. <title>

- La etiqueta <title> define el título del documento que aparece en la pestaña del navegador y en los marcadores (favoritos).

### Detalles:

- Es parte del <head>.
- Tiene un impacto en el SEO (Optimización para Motores de Búsqueda).

### ¿Por qué es importante?

- Sin un <title>, tu página podría mostrarse con un título genérico como "Sin título", lo cual no es profesional y dificulta que los usuarios identifiquen tu página.

# Estructura

## 5. <meta>

Las etiquetas <meta> son elementos que contienen información adicional (metadatos) sobre la página, como por ejemplo:

- **Codificación de caracteres:** Asegura que el texto (como acentos o caracteres especiales) se muestre correctamente.
- **Descripción del contenido:** Define un resumen de la página que se usa en los resultados de búsqueda.
- **Palabras clave:** Proporciona palabras clave relacionadas con el contenido (menos usado hoy en día).
- **Configuración del viewport:** Es crucial para que las páginas se vean bien en dispositivos móviles.



# Estructura

## 6. `<body>`

- La etiqueta `<body>` contiene todo el contenido visible de la página web.
- Aquí es donde escribes lo que quieres mostrar en el navegador, como texto, imágenes, enlaces y más.

### Detalles:

- Todo lo que esté dentro de `<body>` aparecerá en la página web.
- Puede incluir etiquetas como: `<h1>`, `<h6>`, `<p>`, `<img>`, `<a>`
- El `<body>` es como el "escenario" de una obra de teatro: todo lo que los espectadores ven ocurre dentro de él.

# Estructura

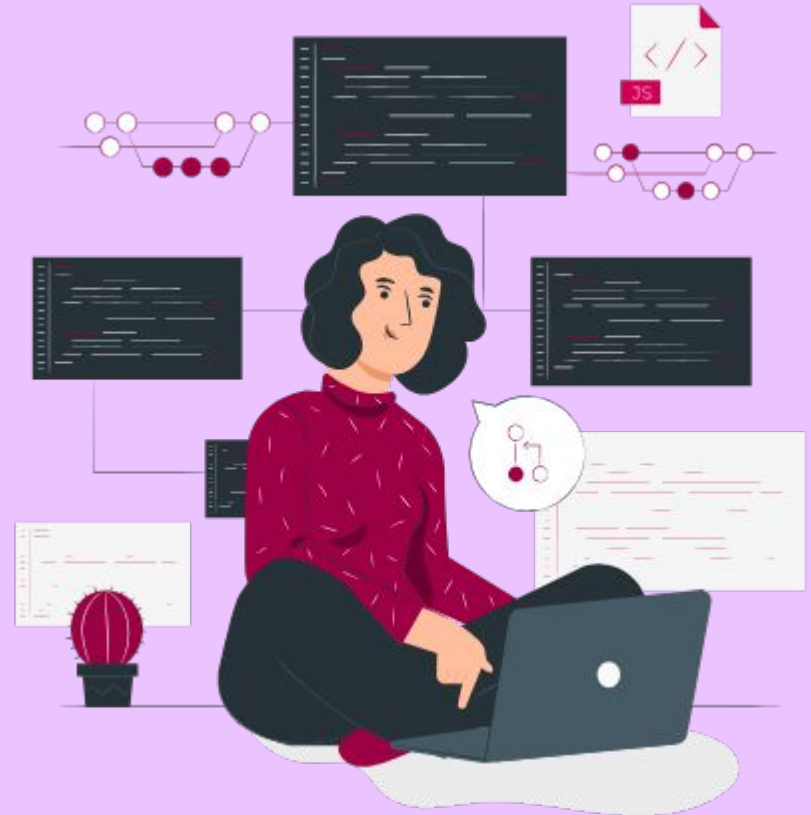


```
<html>
  <!-- Esta etiqueta es el elemento padre de todo nuestro documento.
  Establecemos que estamos armando una estructura en HTML. -->
</html>
```

```
<html>
  <head>
    <title>Título de nuestro documento y se ve en la pestaña del navegador.</title>
    <!-- Todo lo que no vemos en el navegador.
    Incluye información sobre nuestro documento, como el título (<title>),
    enlaces a sripts y hojas de estilos. -->
  </head>

  <body>
    <!-- Todo lo que sí vemos en el navegador.
    Incluye todo el contenido que queremos mostrar en nuestra página. -->
  </body>
</html>
```

**Veamos un  
ejemplo de  
este tema**



Ahora es  
momento  
de aplicar **lo**  
**aprendido en**  
**clase**



# Ejercitación

- <https://github.com/Ada-IT/ejercicios-frontend/blob/master/modulo-1/ejercicios/04-introduccion-a-html.md>