

PROGRAMACIÓN II

1º AÑO

Clase N° 5: Introducción a HTML, CSS y JavaScript

Contenido:

En la clase de hoy trabajaremos los siguientes temas:

- Qué es **HTML** y cómo se estructura una página web.
- Los estilos con **CSS** para mejorar la apariencia visual.
- Aplicar **JavaScript** para agregar interactividad básica.
- Crear un formulario simple y una landing page.
- Uso de **Flask** integrando front-end con back-end.

1. Presentación:

Bienvenidos a la clase N.º 5 de Programación 2. Hoy comenzamos con una introducción al desarrollo web, utilizando tres tecnologías fundamentales: HTML, CSS y JavaScript. Estos lenguajes son la base de cualquier sitio web moderno y permiten estructurar, diseñar y dotar de interactividad a nuestras páginas.

Si bien nuestro foco principal es la programación en Python, en las próximas clases comenzaremos a utilizar **Flask**, un **microframework** que permite crear aplicaciones web. Para poder aprovechar **Flask**, es importante que comprendamos cómo funciona el lado del cliente: cómo se construye una página web, cómo se envían formularios y cómo se responde a eventos del usuario. Esta clase nos dará esas herramientas iniciales.

PROGRAMACIÓN II

1º 2º Cuatrimestre



A lo largo de la clase veremos cómo se estructura una página web con **HTML**, cómo se aplica diseño con **CSS** y cómo se puede usar **JavaScript** para validar datos o responder a clics del usuario. Todo esto será clave para crear aplicaciones web interactivas y funcionales con Python en el servidor.

Los invito a ver el siguiente video sobre HTML y reflexionar sobre el desarrollo web <https://www.youtube.com/watch?v=EvKm8yhM7V8>

2. Desarrollo

¿Qué es HTML y para qué sirve?

HTML (HyperText Markup Language) es el **lenguaje de marcado** estándar que se utiliza para crear y estructurar el contenido de las páginas web. A través de un sistema de **etiquetas** (o *tags*), HTML le indica al navegador qué tipo de elementos debe mostrar —como títulos, párrafos, listas, imágenes, enlaces, botones o formularios— y cómo están organizados dentro del documento.

Podemos pensar en HTML como la **estructura esquelética** de una página web. Define qué contenido hay y cómo se distribuye en la pantalla, pero **no se encarga de los estilos visuales ni de la lógica del comportamiento**. Para eso se complementa con:

- **CSS** (Cascading Style Sheets): que define la apariencia visual (colores, tamaños, márgenes, tipografías, animaciones, etc.).

PROGRAMACIÓN II

1º 2º Cuatrimestre



- **JavaScript:** que permite agregar interactividad y lógica (por ejemplo, validación de formularios, menús dinámicos o ventanas emergentes).

Aunque HTML no incluye estructuras de control como condicionales o bucles (por eso no se lo considera un lenguaje de programación), es **esencial** como punto de partida para cualquier sitio web. Es lo primero que interpreta el navegador al cargar una página y lo que le permite mostrar correctamente el contenido.

Además, HTML es un lenguaje **estándar, abierto y universal**, lo que significa que cualquier persona puede usarlo para crear contenido accesible desde cualquier navegador moderno, en cualquier dispositivo y sistema operativo.

Características principales de HTML

- **Es un lenguaje de marcado, no de programación:**

Su función es **marcar o etiquetar** cada parte del contenido, no ejecutar acciones o cálculos.

- **Se basa en etiquetas:**

Las etiquetas van entre signos de menor y mayor (< >) y muchas tienen apertura y cierre:

```
<p>Esto es un párrafo</p>
```

- **Estructura jerárquica (DOM):**

El contenido se organiza como un árbol de elementos donde unos contienen a otros. Esta estructura se llama **DOM** (Document Object Model) y es clave para interactuar con JavaScript más adelante.

Estructura básica de una página HTML

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Mi primera web</title>
</head>
<body>
  <h1>Hola mundo</h1>
  <p>Este es mi primer sitio en HTML.</p>
</body>
</html>
```

Explicación:

- `<!DOCTYPE html>`: Le indica al navegador que el documento es HTML5.
- `<html>`: Elemento raíz del documento.
- `<head>`: Contiene información "invisible" para el usuario (título, codificación, enlaces a estilos).
- `<body>`: Contiene el contenido visible: textos, imágenes, formularios, etc.
- `<h1>` y `<p>`: Un título y un párrafo dentro del cuerpo.

Etiquetas HTML comunes

- <h1> a <h6>: Encabezados, del más grande (h1) al más pequeño (h6).
- <p>: Párrafo de texto.
- : Enlace hacia otra página o recurso.
- : Imagen insertada en la página.
- , , : Listas desordenadas y ordenadas.

Ejemplo:

```
<h2>Mis hobbies</h2>
<ul>
  <li>Leer</li>
  <li>Programar</li>
  <li>Jugar al ajedrez</li>
</ul>
<a href="https://www.ejemplo.com">Visita mi blog</a>
```

¿Qué es un formulario en HTML?

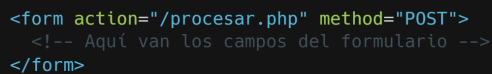
Un **formulario** es un conjunto de campos que permiten al usuario **ingresar datos** y luego enviarlos al servidor o a una aplicación. Es un componente fundamental de la mayoría de los sitios web interactivos. Se utiliza, por ejemplo, para:

- Iniciar sesión con un usuario y contraseña.
- Registrarse en un sitio web.

- Enviar un mensaje de contacto.
- Realizar compras.
- Completar encuestas o formularios administrativos.

El formulario en HTML se define con la etiqueta `<form>`. Dentro de esta etiqueta colocamos distintos **controles de entrada**: cajas de texto, botones, casillas de verificación, menús desplegables, etc.

⚙ Estructura de un formulario



```
<form action="/procesar.php" method="POST">
  <!-- Aquí van los campos del formulario -->
</form>
```

- **action**: indica la URL o archivo a donde se enviarán los datos del formulario. Puede ser una ruta interna (por ejemplo, un archivo PHP o un endpoint de una API).
- **method**: define el método HTTP que se usará para enviar los datos. Los más comunes son:
 - o **GET**: envía los datos en la URL (menos seguro, pero útil para búsquedas).
 - o **POST**: envía los datos ocultos en el cuerpo del mensaje (más seguro, ideal para contraseñas, formularios largos, etc.).

PROGRAMACIÓN II

1º 2º Cuatrimestre



Actividad 2: Explorando etiquetas HTML con ayuda de la IA

Instrucciones:

Abrí ChatGPT y escribí la siguiente pregunta:

“¿Cuáles son las 10 etiquetas HTML más comunes y para qué sirven?”

Leé la respuesta y elegí 3 etiquetas que no conocías o que te parecieron interesantes.

Armá un breve archivo .html en tu computadora donde las uses en un ejemplo simple (puede ser un mini currículum, una lista de compras, una presentación personal, etc.) y guárdalo como repaso_etiquetas.html.

Objetivo: Reforzar el reconocimiento y uso básico de etiquetas HTML con apoyo del chatbot.

Componentes principales de un formulario

1. label: etiqueta del campo

La etiqueta <label> describe el propósito de un campo. Mejora la accesibilidad y la experiencia de usuario.

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

El atributo for debe coincidir con el id del campo correspondiente. Así, al hacer

clic sobre la etiqueta, se selecciona el campo.

2. input: campo de entrada

Es el elemento más usado para recoger datos. El atributo type determina el tipo de dato que se espera.

Algunos tipos comunes:

Tipo	Descripción
• text	Texto corto
• email	Verifica que sea una dirección de correo
• password	Oculto el contenido al escribir
• number	Solo acepta números
• checkbox	Casilla para marcar o desmarcar
• radio	Selección de una opción entre varias
• submit	Botón para enviar el formulario

Ejemplo:

```
<input type="email" name="correo" placeholder="Ej: usuario@correo.com">
```

- **name:** clave con la que se envía el dato.
- **placeholder:** texto de ejemplo dentro del campo.

3. Otros elementos útiles

- **Textarea:** para textos largos.

```
<textarea name="mensaje" rows="4" cols="40"></textarea>
```

- **Select y option:** para listas desplegables.

```
<select name="pais">
  <option value="ar">Argentina</option>
  <option value="cl">Chile</option>
  <option value="uy">Uruguay</option>
</select>
```

- **Botón de envío:**

```
<input type="submit" value="Enviar formulario">
```

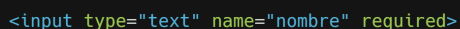
¿Qué sucede cuando se envía un formulario?

1. El usuario completa los campos del formulario.

2. Al hacer clic en el botón de envío (submit), el navegador toma todos los datos cargados.
3. Dependiendo del método (GET o POST), los datos se envían al servidor indicado en action.
4. El servidor procesa la información: puede guardarla en una base de datos, validar credenciales, enviar un correo, etc.
5. Luego, el usuario puede ser redirigido a otra página o recibir una respuesta en pantalla.

Consideraciones de seguridad y accesibilidad

- No se debe confiar únicamente en la validación del lado del cliente (JavaScript o atributos de HTML). Siempre se debe validar la información también en el servidor.
- Usar etiquetas <label> y placeholder ayuda a los usuarios y mejora la accesibilidad.
- El atributo required permite hacer campos obligatorios:



```
<input type="text" name="nombre" required>
```

Actividad 3: Formularios en HTML – Verdadero o Falso

Respondé si las siguientes afirmaciones son verdaderas o falsas.

La etiqueta <input type="email"> permite verificar que se haya ingresado una

PROGRAMACIÓN II

1º 2º Cuatrimestre



dirección de correo válida.

El atributo action en un formulario define el color del botón de envío.

<label> mejora la accesibilidad y se vincula con un campo mediante el atributo for.

method="GET" envía los datos ocultos en el cuerpo del mensaje.

Un formulario sin botón de tipo submit no se puede enviar correctamente.

Objetivo: Reafirmar los conceptos básicos sobre la estructura y funcionamiento de formularios en HTML.

PROGRAMACIÓN II

1º 2º Cuatrimestre



Programacion II - Web para Flask

Introducción a HTML, CSS y JavaScript

Inscríbete ahora

¿Qué vas a aprender?

- ✓ Cómo estructurar una página web con HTML
- ✓ Estilizarla con CSS
- ✓ Agregar interactividad con JavaScript

Formulario de Inscripción

Nombre:

Correo electrónico:

Mensaje:

Enviar

© 2025 Centro Politécnico Malvinas Argentinas

 Ejemplo completo

PROGRAMACIÓN II

1º 2º Cuatrimestre



```
<h2>Formulario de contacto</h2>

<form action="/contacto" method="POST">
  <label for="nombre">Nombre:</label><br>
  <input type="text" id="nombre" name="nombre" required><br><br>

  <label for="email">Correo electrónico:</label><br>
  <input type="email" id="email" name="email" required><br><br>

  <label for="mensaje">Mensaje:</label><br>
  <textarea id="mensaje" name="mensaje" rows="5" cols="40"></textarea><br><br>

  <input type="submit" value="Enviar">
</form>
```

¿Qué es CSS y cómo estiliza la web?

CSS (Cascading Style Sheets) es el lenguaje encargado de definir el **aspecto visual** de las páginas web. Mientras que **HTML** se ocupa de la estructura y el contenido (títulos, párrafos, formularios, etc.), **CSS se encarga de cómo se ve todo eso en pantalla**: colores, tamaños, tipos de letra, espaciados, márgenes, bordes, alineaciones, posiciones, animaciones y mucho más.

Una de las principales virtudes de CSS es que permite **separar el contenido del diseño**, lo que mejora la organización del código, facilita el mantenimiento y permite aplicar estilos de manera consistente en todo el sitio.

📌 Ejemplo 1: cambio de color y estilo de fuente

```
<p style="color: green; font-family: Arial;">Este texto es verde y tiene una fuente diferente.</p>
```

PROGRAMACIÓN II

1º 2º Cuatrimestre



Ejemplo 2: estilo interno

```
<head>
  <style>
    h1 {
      color: navy;
      font-size: 36px;
    }
    p {
      color: gray;
    }
  </style>
</head>
```

Ejemplo 3: CSS externo - archivo estilos.css:

```
body {
  background-color: #f2f2f2;
  font-family: Verdana;
}

.container {
  margin: 20px;
  padding: 10px;
  background-color: white;
  border-radius: 8px;
}
```

En HTML:

```
<link rel="stylesheet" href="estilos.css">
<div class="container">
  <h1>Bienvenido</h1>
  <p>Este es un ejemplo estilizado desde un archivo externo.</p>
</div>
```

- Conceptos clave:
- Selectores: indican a qué elementos se aplica el estilo (por etiqueta, clase, ID).

PROGRAMACIÓN II

1º 2º Cuatrimestre



- Propiedades: lo que se quiere modificar (color, tamaño, etc.).
- Valores: asignación a la propiedad (color: red;).

💡 ¿Qué es JavaScript y por qué es importante?

JavaScript (JS) es un **lenguaje de programación** que se ejecuta principalmente en el navegador del usuario. Junto con **HTML** (estructura) y **CSS** (estilo), JavaScript es una de las **tecnologías fundamentales** del desarrollo web.

Gracias a JavaScript, las páginas web pueden **responder a las acciones del usuario, modificar su contenido en tiempo real, realizar cálculos, validar formularios, cargar datos de forma dinámica**, y mucho más.

Mientras que HTML define **qué se muestra**, y CSS define **cómo se ve**, JavaScript define **cómo se comporta** la página web.

🚀 ¿Por qué JavaScript es tan importante?

1. **Se ejecuta en el navegador (lado del cliente)**: No necesita instalar nada adicional.
2. **Hace que las páginas sean interactivas**: Permite detectar clics, escribir texto, mover el mouse, etc.
3. **Permite comunicación con servidores sin recargar la página** (con tecnologías como **Fetch** o **Ajax**).
4. **Es universal**: Todos los navegadores modernos lo soportan.






PROGRAMACIÓN II

1° 2° Cuatrimestre



5. **Forma parte del stack de tecnologías modernas:** se usa también en el backend (Node.js), en apps móviles (React Native), servidores, videojuegos y más.

Funciones comunes de JavaScript en una página web

Función	Descripción
 Detectar eventos	Saber cuándo el usuario hace clic, escribe o se desplaza
 Validar datos	Verificar que un formulario esté bien completo antes de enviarlo
 Modificar contenido dinámicamente	Cambiar textos, imágenes o estilos sin recargar la página
 Crear efectos visuales	Mostrar u ocultar elementos, animaciones, sliders, etc.
 Comunicar con el servidor	Cargar o enviar datos usando APIs, sin recargar (Fetch/Ajax)

Ejemplos básicos de JavaScript

Ejemplo 1: Mostrar un mensaje al hacer clic

PROGRAMACIÓN II

1º 2º Cuatrimestre



```
<!-- Cuando el usuario hace clic en el botón, aparece un mensaje emergente -->  
<button onclick="alert('¡Bienvenido a la página!')">Saludar</button>
```

alert() es una función de JavaScript que muestra una ventana emergente con un mensaje. onclick es un **evento** que se ejecuta cuando se hace clic en el botón.

Ejemplo 2: Cambiar el contenido de un elemento



```
html  
CopiarEditar  
<!-- Texto inicial -->  
<p id="mensaje">Texto original</p>  
  
<!-- Al hacer clic, cambia el contenido del párrafo -->  
<button onclick="document.getElementById('mensaje').innerText = 'Texto actualizado'">Cambiar texto</button>
```

document.getElementById('mensaje') busca el elemento con ese ID en el HTML. .innerText permite cambiar el texto que se muestra.

Ejemplo 3: Validación de campos en un formulario

PROGRAMACIÓN II

1º 2º Cuatrimestre



```
<!-- Formulario básico con validación -->
<form onsubmit="return validarFormulario()">
  <input type="text" id="nombre" placeholder="Nombre">
  <input type="submit" value="Enviar">
</form>

<script>
function validarFormulario() {
  // Tomamos el valor del campo de texto
  const nombre = document.getElementById("nombre").value;

  // Verificamos si el campo está vacío
  if (nombre.trim() === "") {
    alert("El campo nombre es obligatorio.");
    return false; // Evita que se envíe el formulario
  }

  return true; // Si todo está bien, se envía
}
</script>
```

Este ejemplo muestra cómo JavaScript puede **intervenir antes de enviar un formulario**. Si el campo está vacío, aparece una alerta y no se envía.

🎓 ¿Qué es un evento en JavaScript?

Un **evento** es una acción que ocurre en la página y que podemos "escuchar" con JavaScript. Algunos ejemplos de eventos:

- onclick: cuando se hace clic.
- oninput: cuando el usuario escribe en un campo.
- onmouseover: cuando el mouse pasa por encima.
- onsubmit: cuando se envía un formulario.

Con JavaScript, podemos **asignar funciones a estos eventos**, de forma que la página "responda" a lo que hace el usuario.

Avanzando más allá: el DOM

El **DOM** (Document Object Model) es la representación de la estructura del HTML como un árbol de objetos. JavaScript puede acceder y modificar cualquier parte del DOM. Por eso, podemos:

- Cambiar textos, estilos o atributos.
- Agregar o eliminar elementos.
- Reaccionar ante interacciones del usuario.

HTML + CSS + JavaScript: una combinación poderosa

Cuando usamos estos tres lenguajes juntos, obtenemos una experiencia web completa:

- HTML estructura el contenido.
- CSS embellece y organiza visualmente.
- JavaScript agrega interacción, lógica y respuesta.

 Ejemplo final: mini landingpage con formulario validado

PROGRAMACIÓN II

1º 2º Cuatrimestre



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Landing del Curso</title>
  <style>
    body { font-family: sans-serif; background: #eef; padding: 30px; }
    form { background: white; padding: 20px; border-radius: 10px; width: 300px;
margin: auto; }
    input, textarea { width: 100%; margin-bottom: 10px; padding: 8px; }
    h1 { text-align: center; }
  </style>
</head>
<body>
  <h1>Curso de Flask</h1>
  <form onsubmit="return validar()">
    <input type="text" id="nombre" placeholder="Nombre">
    <input type="email" id="email" placeholder="Correo electrónico">
    <textarea id="mensaje" placeholder="Mensaje"></textarea>
    <button type="submit">Enviar</button>
  </form>

  <script>
    function validar() {
      let nombre = document.getElementById("nombre").value;
      let email = document.getElementById("email").value;

      if (nombre === "" || email === "") {
        alert("Por favor completa los campos requeridos.");
        return false;
      }
      return true;
    }
  </script>
</body>
</html>
```

● Cuestionario de Autoevaluación – Obligatorio

Para cerrar esta clase, les proponemos cuestionario de autoevaluación en la plataforma. Este cuestionario tiene como objetivo reforzar los conocimientos adquiridos sobre formularios HTML, etiquetas básicas y validación de datos.

Criterios de Evaluación

1. Claridad y Coherencia en las Respuestas :

- Las respuestas deben ser claras, bien estructuradas y fáciles de seguir. Se evaluará la capacidad del estudiante para expresar sus ideas de manera coherente y lógica.

2. Profundidad del Análisis:

- Se valorará la profundidad del análisis crítico en relación con el artículo. Los estudiantes deben demostrar que han comprendido los puntos clave y que pueden relacionarlos con sus propias experiencias y conocimientos.

3. Argumentación y Justificación:

- Los estudiantes deben respaldar sus opiniones con argumentos sólidos y ejemplos concretos. Se evaluará la capacidad para justificar sus afirmaciones y reflexiones de manera efectiva.

4. Participación y Colaboración en el Foro:

- Se tendrá en cuenta la participación activa en el foro, incluyendo la capacidad de responder a las intervenciones de otros compañeros

PROGRAMACIÓN II

1º 2º Cuatrimestre



de manera constructiva y respetuosa.

Cierre:

En esta clase nos introdujimos en el mundo del desarrollo web, aprendiendo los fundamentos de HTML, CSS y JavaScript. Estos tres lenguajes son la base para poder construir interfaces visuales, formularios y páginas que respondan a las acciones de los usuarios. Esta comprensión es clave para dar el siguiente paso: integrar nuestras páginas web con Python a través de Flask.

En la próxima clase, comenzaremos a crear nuestras primeras aplicaciones web completas, utilizando todo lo aprendido hasta ahora. Vamos a ver cómo se conectan el front-end y el back-end, cómo se reciben y procesan formularios, y cómo Flask nos ayuda a organizar nuestro código.

¡Excelente trabajo hasta acá! Seguimos avanzando paso a paso en el camino del desarrollo de software moderno.

PROGRAMACIÓN II

1º 2º Cuatrimestre



Bibliografía Obligatoria:

- Duckett, J. (2014). HTML and CSS: Design and Build Websites. Wiley.
- Freeman, Eric (2021). Head First JavaScript Programming. O'Reilly.

Recursos adicionales:

- **HTML: formularios básicos:**

https://developer.mozilla.org/es/docs/Learn/Forms/Your_first_form

👉 Excelente guía práctica de MDN (Mozilla) para crear formularios correctamente.

- **CSS para principiantes:**

<https://www.w3schools.com/css/default.asp>

👉 Referencia clara para estilos, selectores, propiedades y ejemplos visuales.

- **JavaScript básico – Manipular formularios:**

https://developer.mozilla.org/es/docs/Learn/Forms/Form_validation

👉 Introducción a la validación de formularios con JS, paso a paso.

- **Curso gratuito de HTML + CSS + JS (en español)**

PROGRAMACIÓN II

1º 2º Cuatrimestre



<https://www.freecodecamp.org/espanol/>

👉 Plataforma para practicar con ejercicios guiados y recibir certificados.