



# BIENVENIDOS TUTORIA IPC1

Pablo Alejandro Marroquin Cutz  
202200214  
Voluntariado



# HTML: LA ESTRUCTURA DE LA WEB

- HTML (HyperText Markup Language) es el lenguaje utilizado para crear la estructura de las páginas web. Cada elemento en la web está contenido dentro de una etiqueta HTML.
- Etiquetas básicas:
- <html>: Contenedor principal de todo el documento.
- <head>: Contiene metadatos, como el título de la página y enlaces a estilos o scripts.
- <body>: Donde reside el contenido visible de la página.





HTML5 introdujo etiquetas semánticas que mejoran la accesibilidad y SEO:

- <header>: Encabezado de la página o sección.
  - <nav>: Contiene enlaces de navegación.
  - <main>: El contenido principal.
  - <footer>: Pie de página.
  - <article>, <section>, <aside>: Para organizar mejor el contenido.
- 

# CSS: ESTILIZANDO LA WEB

- CSS (Cascading Style Sheets) define cómo los elementos HTML deben visualizarse.
- Se pueden aplicar estilos a través de tres métodos:

Interno: Dentro de una etiqueta `<style>` en el `<head>`

Inline: Dentro de las etiquetas HTML

```
<h1 style="color: red;">¡Hola, Mundo!</h1>
```

```
<style>
  h1 { color: blue; }
</style>
```

Externo: Usando archivos CSS externos enlazados.

```
<link rel="stylesheet" href="styles.css">
```

# FLEXBOX Y GRID PARA LAYOUTS AVANZADOS

Flexbox es una técnica para crear layouts flexibles.  
Alinea los elementos de manera dinámica en una  
fila o columna.

```
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}
```

CSS Grid permite crear layouts en  
dos dimensiones (filas y columnas).

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
}  
.grid-item {  
  border: 1px solid #333;  
}
```

# FRAMEWORKS Y BIBLIOTECAS

## ¿Qué es un framework?

- **Un framework es un conjunto de herramientas que facilita el desarrollo de aplicaciones, ya que ofrece una estructura predeterminada.**
- **Ejemplos de frameworks populares:**
  - **Frontend:**
    - **Angular (Google): Basado en TypeScript, ideal para aplicaciones complejas.**
    - **Vue.js: Ligero y fácil de aprender.**
  - **Backend:**
    - **Express.js: Framework minimalista para Node.js.**
    - **Django: Framework de Python que sigue el patrón MVC.**

# ¿QUÉ ES UNA BIBLIOTECA?

- **Bibliotecas son colecciones de funciones o módulos que los desarrolladores pueden usar, sin imponer una estructura rígida.**
- **React: Biblioteca de JavaScript para construir interfaces de usuario. Desarrollado por Facebook en 2013. Permite la creación de componentes reutilizables y usa un DOM virtual para mejorar el rendimiento.**
- **JSON (JavaScript Object Notation):**
  - **Es un formato de intercambio de datos ligero y fácil de leer.**
  - **Es usado por los servidores para comunicarse entre ellos y enviar datos al frontend.**
  - **Ejemplo de JSON**

```
{  
  "nombre": "Juan",  
  "edad": 25,  
  "profesion": "Desarrollador"  
}
```

# REACT: COMPONENTES Y ESTADO

- React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario interactivas.
- Principales características:
- Componentes: Bloques reutilizables de UI. Cada componente tiene su propio estado y lógica.
- Virtual DOM: Optimiza la actualización del DOM real, mejorando el rendimiento.

```
function Saludo() {  
  return <h1>¡Hola, Mundo!</h1>;  
}
```



## Estado (State): Información que cambia con el tiempo en un componente.

- Ejemplo de estado:

```
const [contador, setContador] = useState(0);
return (
  <div>
    <p>Contador: {contador}</p>
    <button onClick={() => setContador(contador + 1)}>Incrementar</button>
  </div>
);
```

Props: Parámetros que se pasan a los componentes.

```
function Tarea({ texto }) {
  return <li>{texto}</li>;
}
```

# TAILWIND CSS: DISEÑO CON CLASES UTILITARIAS

¿Qué es Tailwind CSS?

- Tailwind CSS es un framework de diseño utilitario que permite aplicar clases CSS directamente a los elementos HTML.
- Ventajas:
  - Ahorra tiempo en la creación de estilos personalizados.
  - Permite prototipar diseños rápidamente.
  - Elimina la necesidad de escribir mucho CSS personalizado.
- Ejemplo de clases utilitarias:

```
<button class="bg-blue-500 text-white font-bold py-2 px-4 rounded">  
    Enviar  
</button>
```

# APLICANDO ESTILOS CON TAILWIND

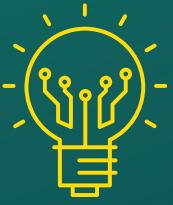
- Sistema de diseño móvil-first: Tailwind organiza sus clases para que el diseño funcione bien primero en móviles y luego se adapte a pantallas más grandes.
- Ejemplo de diseño con Tailwind



```
<div class="flex justify-center items-center h-screen">
  <div class="bg-white p-6 rounded shadow-lg">
    <h1 class="text-xl font-bold">Hola, Tailwind</h1>
    <p class="text-gray-700">Estilizado usando clases utilitarias.</p>
  </div>
</div>
```

# NODE.JS: EL BACKEND CON JAVASCRIPT

- Introducción a Node.js
- Node.js es un entorno de ejecución de JavaScript en el servidor. Te permite usar JavaScript en el backend.
- Ventajas:
- Asincronía: Node.js es muy eficiente manejando múltiples solicitudes al mismo tiempo gracias a su modelo basado en eventos.
- Usa el mismo lenguaje (JavaScript) tanto en frontend como en backend.
- Ejemplo básico de un servidor:



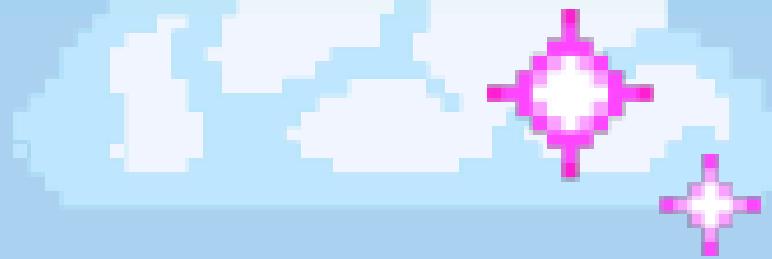
```
const http = require('http');
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('¡Hola desde Node.js!');
});
server.listen(3000);
```

# USANDO EXPRESS.JS CON NODE.JS

- Express.js: Framework de Node.js para crear servidores web rápidos y minimalistas.
- JSON y APIs REST: Express permite manejar rutas para enviar y recibir datos en formato JSON entre frontend y backend.
- Ejemplo de servidor usando Express:

```
const express = require('express');
const app = express();
app.get('/', (req, res) => {
  res.send('¡Hola, Mundo!');
});
app.listen(3000, () => {
  console.log('Servidor escuchando en el puerto 3000');
});
```





¿QUOAS?

