



Actividad 1. Definición de Framework



Actividad 2. Cuadro comparativo Frameworks

Criterio/Aspecto	Django	React Native
Tipo de framework		
Lenguaje		
Arquitectura		
3 ventajas principales	1.	1.
	2.	2.
	3.	3.
Mejor para proyectos		
¿Incluye ORM?		
¿Hot Reloading?		
Filosofía		

Preguntas:

- Un framework es como un _____ que nos proporciona _____ para _____, permitiéndonos enfocarnos en _____ en lugar de _____.
- Si tuvieras que crear una red social como Instagram, ¿qué framework usarías para el backend y cuál para la app móvil?
Respuesta: _____
- Explica con tus palabras qué significa "Aprende una vez, escribe en todas partes" en React Native:
Respuesta: _____
- ¿Por qué crees que Django incluye medidas de seguridad por defecto?
Respuesta: _____
- ¿Cuál framework elegirías para crear una API REST? ¿Por qué?
Respuesta: _____
- Verdadero o Falso:
 - Django es un framework frontend para aplicaciones móviles. → ()
 - ORM Característica de React Native. → ()
 - Hot Reloading Característica de Django. → ()
- Completa:
React Native permite desarrollar apps para _____ y _____ con un mismo código.
Un framework es como un _____ que nos proporciona _____ para _____, permitiéndonos enfocarnos en _____ en lugar de _____.



Actividad 3. Cuadro comparativo Django (*Backend*) Vs React (*Frontend*)

	Django (Backend)	React (Frontend)
Definición		
Principales características		
Estructura		

Actividad 4. Integración de la práctica de las pizzas

Integración de práctica de las pizzas																								
🔥 Estación 1: La Cocina del Pizzero (Django – Back-end)																								
Pizza	Modelos de Django	Campos de modelos - CharField → Nombre de la pizza (<i>Margarita</i>). - DecimalField → Precio (<i>12.99</i>). - TextField → Ingredientes (<i>Queso, tomate, albahaca</i>). - BooleanField → "¿Es vegana?" (checkbox ☒).	# models.py (simulado en cartón) <pre>class Pizza(models.Model): nombre = CharField("Margarita") # Imanes pegados precio = DecimalField("9.99") ingredientes = TextField("Mozzarella, tomate, albahaca") es_vegana = BooleanField(False)</pre>																					
Contenido del libro	Django Admin	<i>Django Admin</i> <i>Organiza y edita datos sin tocar código</i>	<table><tr><th>ID</th><th>Nombre</th><th>Precio</th><th>Ingredientes</th><th>¿Vegana?</th></tr><tr><td>1</td><td>Margarita</td><td>9.99</td><td>Queso, tomate, albahaca</td><td>☒</td></tr><tr><td>2</td><td>Pepperoni</td><td>12.99</td><td>Queso, pepperoni, salsa</td><td>☒</td></tr><tr><td>3</td><td>Vegana</td><td>10.99</td><td>Champiñones, tofu, espinaca</td><td>☒</td></tr></table>		ID	Nombre	Precio	Ingredientes	¿Vegana?	1	Margarita	9.99	Queso, tomate, albahaca	☒	2	Pepperoni	12.99	Queso, pepperoni, salsa	☒	3	Vegana	10.99	Champiñones, tofu, espinaca	☒
ID	Nombre	Precio	Ingredientes	¿Vegana?																				
1	Margarita	9.99	Queso, tomate, albahaca	☒																				
2	Pepperoni	12.99	Queso, pepperoni, salsa	☒																				
3	Vegana	10.99	Champiñones, tofu, espinaca	☒																				
Utensilios del Pizzero	Herramientas de Django	ORM = Paleta para manipular datos. Migraciones = Horno que actualiza la DB.																						
🍷 Estación 2: Los Meseros (React – Front-end)																								
Carta de las pizzas	Componentes React	Componentes reutilizables: - Header: nombre de la pizza. - Body: (precio + imagen). - Footer: Trozo inferior (botón de acción). Props: - nombre: "Pepperoni". - precio: "\$12.99". - imagen: Dibujo de una pizza.	// TarjetaPizza.js (simulado con piezas físicas) <pre>const TarjetaPizza = ({ nombre, precio, imagen }) => (<div className="pizza"> <Header>{nombre}</Header> <!-- Post-it "Pepperoni" --> <Body> <Precio>{precio}</Precio> <!-- Post-it "\$12.99" --> <Imagen src={imagen} /> <!-- Dibujo de pizza --> </Body> <Footer> <Boton>★ Añadir al carrito</Boton> <!-- Ícono de papel --> </Footer> </div>);</pre>																					
Toppings Especiales	Bibliotecas de React	paquetes npm - react-icons: Íconos de carrito 🛒, estrella ★, corazón ❤️. - react-toastify: Notificación de "¡Pedido añadido!".	// Ejemplo simulado en el Footer de TarjetaPizza <pre>import { FaShoppingCart } from "react-icons/fa"; <Footer> <button> <FaShoppingCart /> <!-- Ícono pegado del sobre --> Añadir al carrito </button> </Footer></pre>																					
Tomando Pedidos	API Fetch	Mesero (React): Envía mensaje por tarjeta: <i>¡Necesito 2 Pizzas Pepperoni!</i> (fetch('/api/pedidos')). Cocina (Django): Responde con la tarjeta JSON (respuesta HTTP 200). Mesero: Sirve los datos en la interfaz (muestra la tarjeta al cliente).	// Código simulado en React <pre>const [pizza, setPizza] = useState(null); useEffect(() => { fetch("https://api-pizzeria.com/pedidos/1") // Walkie-talkie .then((res) => res.json()) // Cocina responde .then((data) => setPizza(data)); // Muestra datos }, []); return (<div> {pizza && <TarjetaPizza nombre={pizza.nombre} precio={pizza.precio} />} </div>);</pre>																					



Actividad 5. Comparación de Frameworks

Pega aquí tu
cuadro comparativo

Actividad 6. Resumen técnico

Análisis de la plataforma _____

Tipo de API:

Versiones:

Posibles problemas

Actividad 7. Cuestionario fundamentos Web

1. ¿Qué es internet?
2. ¿Qué es un protocolo?
3. ¿Qué es el protocolo TCP/IP?
4. ¿Qué es un DNS?
5. ¿Qué es HTTP y HTTPS?
6. ¿Qué es cliente y servidor Web?
7. ¿Cómo funciona un sitio o página Web?
8. ¿Tipos de lenguajes de programación en el desarrollo Web? Y ¿Tipos del lado del servidor y del lado del cliente Web?
9. ¿Qué es un diseño responsivo?

Respuestas:

Conocimientos previos sobre Fundamentos Web

Instrucciones: Une cada concepto de la **Columna A** con su definición correspondiente en la **Columna B**. Escribe la letra correcta en el espacio provisto.

- | | | |
|---|----------|--|
| 1. Internet | () | a. Conjunto de reglas que definen cómo se comunican los dispositivos en una red. |
| 2. Protocolo | () | b. Sistema que traduce nombres de dominio a direcciones IP. |
| 3. TCP/IP | () | c. Software que solicita y muestra contenido web (ej. navegador). |
| 4. DNS | () | d. Protocolo seguro para transferir datos web de forma cifrada. |
| 5. HTTPS | () | e. Enfoque de diseño que hace que un sitio web se adapte a diferentes tamaños de pantalla. |
| 6. Cliente Web | () | f. Red global de redes de computadoras interconectadas. |
| 7. Servidor Web | () | g. Lenguaje de programación principal para añadir interactividad en el navegador. |
| 8. HTML, CSS, JavaScript (lado del cliente) | () | h. Computadora o software que almacena y entrega páginas web. |
| 9. Python, Node.js, (lado del servidor), SQL (BD) | () | i. Protocolo fundamental para la comunicación en Internet. |
| 10. Diseño Responsivo | () | j. Lenguaje de programación que se ejecuta en el servidor para generar contenido dinámico. |

Pregunta 1:

¿Cuál opción resume mejor cómo funciona la comunicación entre cliente y servidor en la web?

- ☐ El servidor envía datos directamente al DNS sin interacción del cliente.
- ☐ El cliente solicita recursos mediante HTTP(S) y el servidor responde con los datos.
- ☐ El protocolo TCP/IP envía imágenes sin usar dominios ni direcciones IP.
- ☐ El diseño responsivo controla la seguridad de la transmisión de datos.

Pregunta 2:

¿Cuál de los siguientes aspectos representa una buena práctica en desarrollo web?

- ☐ Ignorar la accesibilidad para acelerar la entrega del proyecto.
- ☐ Usar HTTP en lugar de HTTPS para ahorrar costos de servidor.
- ☐ Diseñar páginas que se adapten a diferentes dispositivos y tamaños de pantalla.
- ☐ Depender solo de DNS sin configurar correctamente las rutas IP.



1. Etiquetas básicas

- <div> → contenedor genérico para agrupar elementos.
- <h1> → títulos o encabezados (h1 es el más grande, h6 el más pequeño).
- <p> → párrafos de texto.
- <a> → enlaces (anchor).
- → insertar imágenes.
- <form> → crear formularios para enviar datos.

```
<ul>
  <li>Manzana</li>
  <li>Plátano</li>
  <li>Naranja</li>
</ul>
```

```
<ol>
  <li>Paso 1</li>
  <li>Paso 2</li>
</ol>
```

```
<table border="1">
  <tr>
    <th>Nombre</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Ana</td>
    <td>20</td>
  </tr>
</table>
```

2. Listas y tablas

- → lista desordenada (con viñetas).
- → lista ordenada (con números).
- <table> → tablas de datos.

3. Atributos

- class → agrupar elementos para aplicar estilos.
- id → identificador único.
- href → destino de un enlace.
- src → fuente de imagen o recurso.

4. Etiquetas de imágenes y

- → imágenes.
- <audio> → audio.
- <video> → video.

```
<div>
  <h1>Título principal</h1>
  <p>Este es un párrafo.</p>
  <a href="https://ejemplo.com">Ir a Ejemplo</a>
  
  <form>
    <input type="text" placeholder="Escribe tu nombre">
    <button>Enviar</button>
  </form>
</div>
```

```
<p class="texto-importante">Texto con clase</p>
<p id="parrafo1">Texto con ID</p>
<a href="pagina.html">Ir a página</a>

```

```

<audio controls>
  <source src="musica.mp3" type="audio/mpeg">
</audio>
<video controls width="300">
  <source src="video.mp4" type="video/mp4">
</video>
```

multimedia

5. Etiquetas de formularios y tablas

- Formularios: <form>, <input>, <textarea>, <select>, <button>.
- Tablas: <table>, <tr> (fila), <td> (celda), <th> (encabezado).

6. Etiquetas de división y especiales

- <div> → contenedor genérico.
- → contenedor en línea (para resaltar texto).
-
 → salto de línea.
- <hr> → línea horizontal.
- → espacio en blanco.

```
<form action="enviar.php" method="post">
  <label>Nombre:</label>
  <input type="text" name="nombre">
  <textarea name="mensaje"></textarea>
  <select name="opcion">
    <option value="1">Opción 1</option>
  </select>
  <button type="submit">Enviar</button>
</form>
```

```
<div>
  <p>Texto normal <span style="color:red;">Texto en rojo</span></p>
  <hr>
  <p>Línea de abajo<br>con salto</p>
</div>
```



03

¿Qué es CSS?

Dando Estilo y Vida a las Páginas Web

Definición

CSS, o Cascading Style Sheets, es un lenguaje utilizado para dar estilo y diseño a las páginas web. Junto con HTML, es un componente esencial en la construcción de experiencias web visualmente atractivas y agradables.

¿Qué usos tiene?:

- **Función Principal:** CSS se enfoca en el aspecto visual y la presentación de una página web. Permite controlar colores, tipografías, diseños y espaciado.
- **Independencia de Estilo:** Separa el estilo del contenido, facilitando los cambios y actualizaciones. Sus Cambios de diseño no afectan la estructura del contenido.



Sintaxis y Reglas

Reglas de Estilo

Cada regla CSS consta de un selector y un conjunto de propiedades.

- El **selector** permite identificar al elemento que vamos a modificar
- Las **propiedades** definen cómo se verá el elemento seleccionado.

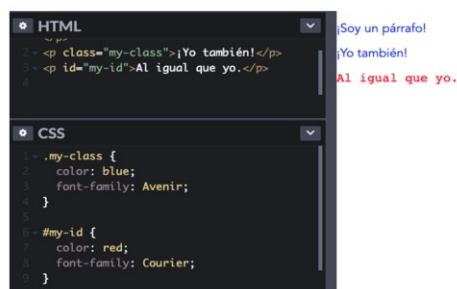
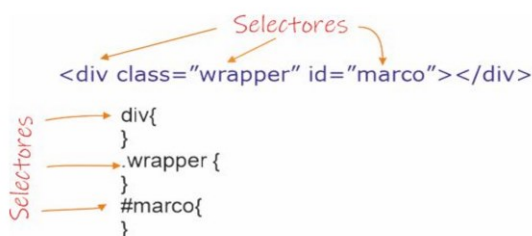


Propiedades y Valores

Hay diferentes tipos de propiedades como **color**, **font-size**, **margin**, **etc**, a los que se les puede **asignar valores** para cada propiedad, lo que nos permite determinar el aspecto deseado.



Aplicación de Estilos



Selección de Elementos

Nosotros podemos seleccionar los elementos HTML que queremos darle estilos mediante **etiquetas**, **clases** o **IDS**.

Clases en CSS

La **clase** en CSS es una propiedad de las etiquetas que nos permiten **referenciar y agrupar** elementos para **editar sus propiedades** usando el **selector de clase**

ID en CSS

El **ID** es similar a las **clases** en **css**, pero este se utiliza más que nada para **identificar un solo elemento**, mientras que una **clase** se puede usar para **agrupar a más de uno**

Unidades y Medidas

Unidades Absolutas

- **px** Píxeles: fijan el tamaño exacto.
- **pt** Puntos: utilizado en impresión.

Unidades Relativas

- **%** Porcentaje: relativo al tamaño del elemento contenedor.
- **em** y **rem**: Relativos al tamaño de fuente base.

Tabla de equivalencia Unidad Medida CSS

Píxeles	Em	%
8px	0.5em	50%
9px	0.55em	55%
10px	0.625em	62.5%
11px	0.7em	70%
12px	0.75em	75%
13px	0.8em	80%
14px	0.875em	87.5%
15px	0.95em	95%
16px	1em	100%

Letra de 20px

Letra de 20px

Letra de 40px



Diseño Responsivo



Media Queries

Cambios de estilo según las características del dispositivo. **Diseño responsivo** para adaptarse a diferentes pantallas.

Mobile-First

Es un enfoque de desarrollo y diseño web que se enfoca en la priorización de **Diseñar primero para dispositivos móviles**, luego para pantallas más grandes.

Conclusión

CSS es el lenguaje que **da vida y estilo a las páginas web**. Al permitir el control detallado de la presentación, transforma las páginas ordinarias en experiencias visuales cautivadoras y coherentes.

Pregunta	Respuesta	Ejemplo de Código
1. ¿Qué es _____ ?	CSS (Cascading Style Sheets). Permite dar estilo, diseño, animaciones y adaptabilidad a las páginas _____.	
2. ¿Cómo vincular CSS a HTML?	Se puede hacer de _____ formas: • _____ • _____ • _____ externo.	En línea: Texto Interno: <style> p { color: blue; } </style> Externo: <link rel="stylesheet" href="estilos.css">
3. ¿Qué son los selectores básicos?	Son formas de aplicar estilos a elementos usando: • etiquetas • _____ • _____	Tag: p { color: green; } Class: .titulo { font-size: 24px; } ID: #principal { background: yellow; }
4. Propiedades básicas de CSS	Se usan para modificar: • color, • _____ • márgenes, • relleno y • fuente.	body { _____ : white; /* Color del texto */ _____ : black; /* Fondo */ _____ : 20px; /* Margen externo */ _____ : 10px; /* Espacio interno */ _____ : 18px; /* Tamaño de letra */ } .caja { width: 200px; padding: 20px; border: 5px solid red; margin: 10px; box-sizing: border-box; } .inline { display: inline; } .block { display: block; } .hidden { display: none; } .rel { position: relative; top: 10px; left: 20px; } .abs { position: absolute; top: 0; right: 0; } .fix { position: fixed; bottom: 0; right: 0; } .contenedor { display: flex; justify-content: center; /* Alineación horizontal */ align-items: center; /* Alineación vertical */ gap: 10px; }
5. ¿Qué es el Modelo de Caja (Box Model)?	Cada elemento es una caja formada por: content → padding → border → margin.	
6. Display y Posicionamiento	_____ controla el comportamiento del elemento, _____ controla su ubicación.	
7. ¿Qué es Flexbox?	Sistema de diseño flexible para alinear y distribuir elementos en un contenedor.	

8. ¿Qué es CSS Grid?	Sistema de diseño basado en cuadrículas.	<pre>.grid { display: grid; grid-template-columns: repeat(3, 1fr); /* 3 columnas iguales */ gap: 10px; }</pre>
9. ¿Qué es Responsive Design (Media Queries)?	Es la capacidad de adaptar el diseño a distintos tamaños de pantalla.	<pre>@_____ (max-width: 600px) { body { font-size: 14px; } }</pre>
10. ¿Qué son Transiciones y Animaciones?	Transiciones: cambios suaves. Animaciones: movimientos definidos con @keyframes.	Transición: <pre>.boton { background: blue; transition: background 0.5s; } .boton:hover { background: red; }</pre> Animación: <pre>@keyframes mover { 0% { transform: translateX(0); } 100% { transform: translateX(100px); } } .animado { animation: mover 2s infinite alternate; }</pre>
11. Efectos Visuales Avanzados	Incluyen sombras, filtros y desenfoques.	<pre>.sombra { box-shadow: 5px 5px 10px gray; text-shadow: 2px 2px 5px black; filter: blur(2px); }</pre>
12. ¿Qué son las Variables CSS (Custom Properties)?	Variables reutilizables para colores, tamaños o espacios.	<pre>:_____ { --color-principal: #3498db; --espaciado: 10px; } .caja { background: var(--color-principal); margin: var(--espaciado); }</pre>



¿Qué es JavaScript?

Añadiendo Interactividad y Dinamismo a la Web



Definición

JavaScript es un lenguaje de programación utilizado en desarrollo web para agregar interactividad y dinamismo a las páginas. Es un componente esencial para crear experiencias en línea ricas y atractivas de código.

Características Clave:

- **Lenguaje de Programación:** JavaScript es un lenguaje de alto nivel Utilizado para controlar el comportamiento de las páginas web en tiempo real.
- **Cliente y Servidor:** Ejecutado en el navegador del cliente (navegador web) y en el servidor (Node.js).
- **Interacción con el DOM:** Manipula el Document Object Model (DOM) para cambiar el contenido y la apariencia de una página sin necesidad de recargarla.



JavaScript transforma páginas web estáticas en aplicaciones dinámicas y atractivas. Su capacidad para agregar interactividad y personalización en tiempo real ha hecho que sea un lenguaje esencial en el desarrollo web moderno.

Aplicaciones de JavaScript



Sintaxis y Estructura



Conclusión

JavaScript transforma páginas web estáticas en aplicaciones dinámicas y atractivas. Su capacidad para agregar interactividad y personalización en tiempo real ha hecho que sea un lenguaje esencial en el desarrollo web moderno.

