

Desarrollo del Caso Práctico: Creación de una Página Web para Receta de Cocina

Introducción

Este documento describe el proceso de creación de una página web basada en el caso práctico presentado. La tarea consiste en desarrollar una página web para mostrar una receta de "pisto manchego", siguiendo los requisitos técnicos de HTML y CSS. Además, se profundiza en conceptos teóricos como la estructura de los documentos HTML y las diferencias entre HTML y XHTML. Este análisis es fundamental para entender las mejores prácticas en el diseño y desarrollo web.

Desarrollo de la Página Web

Estructura HTML

La página web será desarrollada utilizando HTML5, la última versión del estándar HTML, que ofrece múltiples ventajas en comparación con versiones anteriores, como nuevas etiquetas semánticas, soporte mejorado para multimedia y mayor compatibilidad con dispositivos móviles.

A continuación, se presenta el código estructurado de la página:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Receta de Pisto Manchego</title>
    <link rel="stylesheet" href="styles.css">
    <style>
        <link rel="stylesheet" href="styles.css"></link>
    </style>
</head>
<body>
    <header>
        <h1>Receta de Pisto Manchego</h1>
    </header>
    <main>
        <section>
            </img>
        </section>
        <section>
            <h2>Ingredientes</h2>
```

```

<h3><b>Para 4 personas:</b></h3>
<ul>
    <li>Tomates hermosos .....<b>4 ud (1kg aprox)</b></li>
    <li>Cebolla picada .....<b>2 ud (250g aprox)</b></li>
    <li>Pimiento verde .....<b>2 ud (200g aprox)</b></li>
    <li>Pimiento rojo .....<b>2 ud (200g aprox)</b></li>
    <li>Diene de ajo .....<b>2 ud</b></li>
    <li>Calabacín .....<b>1 ud (300g aprox)</b></li>
    <li>Berenjena .....<b>1 ud (300g aprox)</b></li>
    <li>Aceite de oliva virgen extra</li>
    <li>Pimienta negra molida</li>
    <li>Sal al gusto</li>
</ul>
</section>
<section>
    <h2>Preparación</h2>
<p><b>Dificultad:</b> Fácil<br>
<b>Tiempo total:</b> 2 h 5 m<br>
<b>Elaboración:</b> 20 m<br>
<b>Cocción:</b> 1 h 45 m</p>
    <ol>
        <li>Calentar agua en una cacerola amplia y llevar a ebullición.</li>
        <li>Retirar la parte dura de los tomates y practicar un corte de cruz en la base.</li>
        <li>Introducir en el agua hirviendo <b>durante 15-20 segundos.</b></li>
        <li>Sacar e introducir en un cuenco de agua helada.</li>
        <li>Pelar y triturar los tomates.</li>
        <li>Pelar los dientes de ajo y la cebolla y picar finamente.</li>
        <li>Lavar bien el resto de verdura y picar en trozos pequeños.</li>
        <li>Reservar las verduras separadas unas de otras para ir añadiendo por fases.</li>
        <li>Calentar una cantidad generosa de aceite de oliva virgen extra en una cazuela.</li>
    </ol>
</section>

```

```

            <li>Pochar el ajo y la cebolla picadas <b>durante 15 minutos a fuego suave.</b></li>
            <li>Agregar los pimientos y <b>pochar durante 15 minutos más.</b></li>
            <li>Incorporar la berenjena y el calabacín, removiendo ocasionalmente.</li>
            <li>Por último, añadir los tomates troceados, sazonar con sal y pimienta al gusto.</li>
            <li>Cocinar a fuego lento, tapado <b>durante una hora y media como mínimo.</b></li>
            <li>Transcurrido ese tiempo, retirar la tapadera de la cazuela y subir un poco el fuego.</li>
            <li>Cocer durante 15 minutos más o hasta que el agua del tomate se haya evaporado.</li>
            <li><b>Asegurarse de que no quede resto de agua de las verduras</b>, pero tener cuidado de que quede jugoso.</li>
            <li>Servir caliente, acompañado de pan o como guarnición.</li>
        </ol>
    </section>
    <section>
        </img>
    </section>
    <section>
        <h2>¿Te ha gustado esta receta?</h2>
        <form>
            <label for="feedback">Tu opinión:</label><br />
            <textarea id="feedback" rows="4" cols="50"></textarea><br />
            <button type="submit">Enviar</button>
        </form>
    </section><br>
    <section>
        <button onclick="window.print()">Imprimir esta receta</button>
    </section>

</main>
<footer>
    <p>&copy; 2025 Recetas Tradicionales</p>
</footer>
</body>
</html>
```

El código está estructurado siguiendo buenas prácticas, utilizando etiquetas semánticas como `<header>`, `<main>` y `<footer>`, las cuales mejoran la accesibilidad y el SEO. Se implementaron listas ordenadas y desordenadas para organizar los ingredientes y pasos de la receta, así como imágenes con dimensiones y estilos específicos para complementar el diseño visual.

Estilo CSS

Para la presentación de la página web, se diseñó una hoja de estilos que sigue principios de diseño limpio y profesional. Esto incluye una tipografía legible, colores contrastantes y una disposición centrada de los elementos. La hoja de estilos utiliza selectores básicos y avanzados para personalizar cada elemento, como `box-shadow` para realzar las imágenes y `counter-reset` para personalizar listas ordenadas.

```
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    margin: 0;
    padding: 0;
    background-color: #f9f9f9;
    color: #333;
    text-align: center;
}

header {
    background: #ff7043;
    color: #ffffff;
    text-align: center;
    padding: 1rem;
}

h1 {
    color: #ffffff;
}

main {
    padding: 2rem;
}

h2 {
    color: #ff7043;
}

ul, ol {
    margin: 1rem 0;
    padding-left: 1.5rem;
    text-align: left;
    display: inline-block;
}

img {
    display: block;
    margin: 2rem auto;
    border-radius: 8px;
    box-shadow: 0 8px 12px rgba(0, 0, 0, 0.2);
}
```

```
}

ol {
    counter-reset: step;
}

ol li {
    counter-increment: step;
    list-style-type: none;
    margin: 10px 0;
    position: relative;
    padding-left: 30px;
}

ol li::before {
    content: counter(step);
    position: absolute;
    left: 0;
    width: 20px;
    height: 20px;
    background: #ff7043;
    color: white;
    border-radius: 50%;
    text-align: center;
    line-height: 20px;
}

button {
    background-color: #ff7043;
    border: none;
    padding: 10px 20px;
    color: white;
    cursor: pointer;
    border-radius: 5px;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #d65c33;
}

footer {
    text-align: center;
    background: #333;
    color: #fff;
    padding: 0.5rem 0;
}
```

Análisis Teórico

Diferencias entre HTML y XHTML

HTML (HyperText Markup Language) y XHTML (Extensible HyperText Markup Language) son lenguajes de marcado esenciales para la creación de páginas web. A continuación, se presentan las diferencias principales con un mayor detalle:

1. Estricta adherencia a las reglas de sintaxis:

- En XHTML, todas las etiquetas deben cerrarse correctamente, incluyendo aquellas consideradas "vacías" en HTML, como `` y `
`. Esto asegura que el código sea consistente y compatible con más aplicaciones. (Ej: ``)
- HTML, en cambio, permite omitir el cierre de etiquetas en ciertas situaciones, lo que puede resultar en código menos predecible.

2. Uso de atributos y nombres en minúsculas:

- XHTML requiere que todos los nombres de etiquetas y atributos estén en minúsculas. Por ejemplo, se debe escribir `<input type="text">` en lugar de `<INPUT TYPE="text">`. Esta norma promueve la uniformidad en la escritura del código.
- HTML permite tanto mayúsculas como minúsculas, pero esto puede dar lugar a inconsistencias dentro del mismo documento.

3. Validación y compatibilidad:

- XHTML, al ser un subconjunto de XML, debe cumplir con las reglas estrictas de este estándar. Esto significa que cada elemento debe estar correctamente anidado y que el código debe validarse con herramientas como validadores XML.
- En HTML, los navegadores son más flexibles y "perdonan" muchos errores de sintaxis, pero esto puede causar problemas de representación en navegadores más antiguos o con configuraciones específicas.

4. Procesamiento como XML:

- Los documentos XHTML pueden ser procesados por herramientas XML, lo que permite su integración con aplicaciones más avanzadas, como sistemas de gestión de contenido o análisis de datos estructurados.
- HTML no tiene esta ventaja y se limita a ser interpretado directamente por los navegadores.

5. Capacidades de extensión:

- XHTML es más adecuado para la extensión mediante otros lenguajes basados en XML, como MathML o SVG, lo que facilita la inclusión de contenido complejo como fórmulas matemáticas o gráficos vectoriales.
- HTML no ofrece esta flexibilidad de manera nativa, lo que limita su uso en ciertos contextos.

6. Evolución y versiones:

- XHTML representa un intento de hacer el lenguaje HTML más riguroso, pero la aparición de HTML5 ha vuelto a posicionar al HTML como el estándar más usado debido a sus características modernas y su flexibilidad.

En resumen, XHTML es ideal para proyectos que requieren un alto nivel de precisión, interoperabilidad y compatibilidad con sistemas XML. HTML, por su parte, sigue siendo la

opción preferida para desarrollos rápidos y flexibles, especialmente en el contexto de la web moderna.

Herramientas Utilizadas

El desarrollo de esta página web se realizó utilizando herramientas modernas que optimizan el flujo de trabajo y aseguran un diseño eficiente y profesional. Estas son las herramientas clave empleadas y su impacto específico en este proyecto:

1. Visual Studio Code (<https://code.visualstudio.com/>)

- **Razones de uso:** Este editor es uno de los más versátiles y ampliamente utilizados para el desarrollo web. Ofrece soporte integrado para HTML, CSS y JavaScript, además de extensiones como Live Server, que permite previsualizar cambios en tiempo real. Recientemente además ha agregado Python.
- **Ventajas:**
 - Autocompletado inteligente y resaltado de sintaxis que ayudaron a reducir errores al escribir código.
 - Extensiones como **Prettier** facilitaron el formateo del código, garantizando consistencia en su estructura.
 - La extensión Live Server permitió identificar rápidamente errores visuales, como márgenes y alineaciones incorrectas, durante el desarrollo.

2. Navegadores Web (Google Chrome, Mozilla Firefox y Edge)

- **Razones de uso:** Estos navegadores garantizaron que el diseño fuese funcional en diferentes entornos, permitiendo pruebas multiplataforma.
- **Funciones clave utilizadas:**
 - Inspección del DOM para verificar la correcta aplicación de los estilos y detectar problemas con selectores CSS.
 - La consola de JavaScript ayudó a identificar advertencias relacionadas con imágenes y atributos faltantes, que se corrigieron para asegurar compatibilidad.

3. MDN Web Docs (<https://developer.mozilla.org/es/>) y W3Schools

(<https://www.w3schools.com/>)

- **Razones de uso:** Estas plataformas fueron esenciales para profundizar en conceptos clave de HTML5 y CSS3.
- **Ventajas:**
 - Ejemplos prácticos ayudaron a implementar estilos más avanzados, como `box-shadow` y `counter-reset`, para mejorar la presentación de listas y elementos visuales.
 - Información actualizada aseguró que el código cumpliera con los estándares web modernos.

4. Google Drive (<https://drive.google.com/>) y Tiiny.host (<https://tiiny.host/>)

- **Razones de uso:**
 - Google Drive facilitó el almacenamiento y la distribución de archivos del proyecto.
 - Tiiny.host permitió alojar la página web y realizar pruebas en dispositivos móviles.
- **Ventajas:**
 - Estas herramientas facilitaron el acceso rápido y fácil al proyecto desde múltiples dispositivos.
 - Las pruebas realizadas en diferentes pantallas permitieron ajustar elementos visuales, como el tamaño de las imágenes.

5. Validador W3C (World Wide Web Consortium) (<https://validator.w3.org/>)

- **Razones de uso:**

Esta herramienta asegura que el código cumple con los estándares HTML y CSS, mejorando la accesibilidad, el SEO y la compatibilidad entre navegadores.

- **Ventajas:**

- Detectó errores de sintaxis en las etiquetas `` y `
`, que se corrigieron para mejorar la compatibilidad.
- Proporcionó sugerencias valiosas para optimizar el código y garantizar su correcto funcionamiento en diferentes navegadores.

Conclusiones

El desarrollo de este caso práctico permitió aplicar y consolidar conocimientos esenciales para el diseño y la programación web. A continuación, se destacan las principales lecciones aprendidas y su importancia en proyectos futuros:

1. Consolidación de conceptos técnicos

- Este ejercicio reforzó la importancia de utilizar una estructura semántica en HTML5, mejorando tanto la accesibilidad como la optimización para motores de búsqueda (SEO).
- La comparación entre HTML y XHTML destacó cómo la adherencia a normas estrictas contribuye a un código más limpio, aunque en este proyecto se optó por la flexibilidad de HTML5 debido a sus ventajas prácticas.

2. Importancia de las hojas de estilo

- La aplicación de CSS no sólo transformó el contenido estático en una experiencia visualmente atractiva, sino que también permitió implementar un diseño responsive y centrado en el usuario.
- La exploración de propiedades avanzadas como `box-shadow` y `transition` enriqueció el diseño, haciéndolo más dinámico y moderno.

3. Desafíos superados durante el proyecto

- Uno de los retos fue garantizar la coherencia visual en diferentes navegadores y dispositivos. Este problema se resolvió mediante pruebas iterativas y ajustes en las hojas de estilo.
- Otro desafío fue optimizar el orden del código CSS para evitar conflictos entre estilos. La organización en bloques temáticos (ej. `header`, `main`, `footer`) facilitó esta tarea.

4. Valor de las herramientas modernas

- Herramientas como Visual Studio Code y el validador W3C optimizaron significativamente el desarrollo, garantizando que el código fuera funcional y estuviera alineado con los estándares actuales.
- La capacidad de depurar y probar en tiempo real permitió resolver problemas rápidamente, mejorando la eficiencia del flujo de trabajo.

5. Preparación para proyectos más complejos

- Este proyecto brindó una base sólida para abordar tareas más avanzadas, como la integración de JavaScript o el diseño de páginas completamente responsivas.
- Las habilidades adquiridas, como el uso de etiquetas semánticas y la estructuración eficiente del código, serán fundamentales para futuros desarrollos.

En conclusión, este ejercicio no solo cumplió con los requisitos técnicos y visuales, sino que también destacó la importancia de la planificación y la estructura en el diseño web. La experiencia adquirida en este proyecto resulta invaluable para el crecimiento profesional en el desarrollo de aplicaciones web.

Anexos

- Se incluyen capturas de pantalla del diseño final y un enlace a la página web funcional.

Receta de Pisto Manchego



Ingredientes

Para 4 personas:

• Tomates hermosos	4 ud (1kg aprox)
• Cebolla picada	2 ud (250g aprox)
• Pimiento verde	2 ud (200g aprox)
• Pimiento rojo	2 ud (200g aprox)
• Diente de ajo	2 ud
• Calabacín	1 ud (300g aprox)
• Berenjena	1 ud (300g aprox)
• Aceite de oliva virgen extra	
• Pimienta negra molida	
• Sal al gusto	

Preparación

Dificultad: Fácil
Tiempo total: 2 h 5 m
Elaboración: 20 m
Cocción: 1 h 45 m

- ① Calentar agua en una cacerola amplia y llevar a ebullición.
- ② Retirar la parte dura de los tomates y practicar un corte de cruz en la base.
- ③ Introducir en el agua hirviendo durante 15-20 segundos.
- ④ Sacar e introducir en un cuenco de agua helada.
- ⑤ Pelear y triturar los tomates.
- ⑥ Pelear los dientes de ajo y la cebolla y picar finamente.
- ⑦ Lavar bien el resto de verdura y picar en trozos pequeños.
- ⑧ Reservar las verduras separadas unas de otras para ir añadiendo por fases.
- ⑨ Calentar una cantidad generosa de aceite de oliva virgen extra en una cazuela.
- ⑩ Pochar el ajo y la cebolla picadas durante 15 minutos a fuego suave.
- ⑪ Agregar los pimientos y pochar durante 15 minutos más.
- ⑫ Incorporar la berenjena y el calabacín, removiendo ocasionalmente.
- ⑬ Por último, añadir los tomates troceados, sazonar con sal y pimienta al gusto.
- ⑭ Cocinar a fuego lento, tapado durante una hora y media como mínimo.
- ⑮ Transcurrido ese tiempo, retirar la tapadera de la cazuela y subir un poco el fuego.
- ⑯ Cocer durante 15 minutos más o hasta que el agua del tomate se haya evaporado.
- ⑰ Asegurarse de que no quede resto de agua de las verduras, pero tener cuidado de que quede jugoso.
- ⑱ Servir caliente, acompañado de pan o como guarnición.



¿Te ha gustado esta receta?

Tu opinión:

Enviar

Imprimir esta receta

- **Descargar el código HTML simple:**
https://drive.google.com/file/d/1hewYAVMQLjpobD3Mimp6ZMHGf0pMUpUq/view?usp=drive_link
- **Descargar el código CSS:**
https://drive.google.com/file/d/1S84MATqclpJCsvF1Jr0R_dWlewu0jMU1/view?usp=drive_link
- **Descargar el código HTML con el código CSS incorporado:**
https://drive.google.com/file/d/1oaYVffF5d8BM3vxxlh5eX--Fmu0tzpVw/view?usp=drive_link
- **Visualizar el código de manera funcional en página web:**
<https://violet-daryl-54.tiny.site>