

LENGUAJE DE MARCAS

CASO PRACTICO I



ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

José Carlos Villar Herrera

CONTENIDO

01

INTRODUCCION

02

IDENTIFICA
CIÓN Y
DESCRIPCIÓN
DETALLADA DE
LOS
LENGUAJES DE
MARCAS

03

CARACTERÍS
TICAS Y
ÁMBITOS DE
APLICACIÓN
DE LOS
LENGUAJES
DE MARCAS

04

TRATAMIENTO
DE LA
INFORMACIÓN
Y VENTAJAS DE
USAR
LENGUAJES DE
MARCAS

05

CONCLUSION
ES FINALES

06

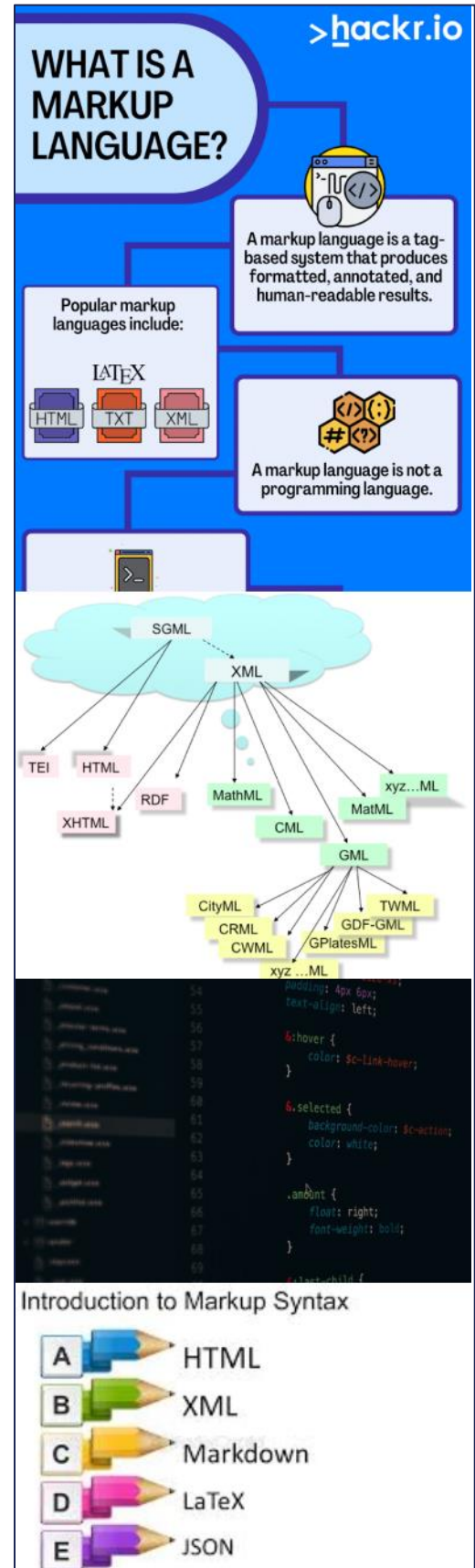
REFERENCIAS

INTRODUCCION

En este caso práctico, se busca proporcionar una visión detallada y comprensiva sobre los **lenguajes de marcas**, herramientas fundamentales en el desarrollo web y en la gestión de datos. Estos lenguajes se utilizan para **estructurar, describir y representar información** de manera que sea comprensible tanto para humanos como para máquinas.

Este análisis examinará cómo estos lenguajes facilitan la **creación de interfaces**, el **intercambio de datos** y la **gestión de configuraciones**, resaltando sus **ventajas** en términos de eficiencia, escalabilidad y flexibilidad. También se profundizará en cómo han evolucionado para **adaptarse a las demandas tecnológicas actuales**, asegurando que continúan siendo herramientas clave en el desarrollo de aplicaciones web modernas y en la creación de sistemas de información.

El objetivo principal de este documento es **identificar y describir** los lenguajes de marcas más relevantes, analizar sus características y profundizar en su uso en diversos contextos. Asimismo, se presentarán ejemplos concretos que ilustran cómo estos lenguajes se aplican en la industria, mejorando la **interoperabilidad** y la **experiencia de usuario** en el desarrollo de soluciones digitales.



IDENTIFICACIÓN Y DESCRIPCIÓN DETALLADA DE LOS LENGUAJES DE MARCAS

1. HTML (HyperText Markup Language)

- **Descripción detallada:**

HTML es el lenguaje de marcado estándar para crear páginas web. La versión más reciente, HTML5, introduce importantes mejoras, incluyendo soporte nativo para multimedia (audio, video, gráficos 2D y 3D) a través de la etiqueta `<canvas>`, y una semántica más rica con etiquetas como `<article>`, `<section>`, y `<nav>`.

- **Conocimiento teórico aplicado:**

HTML5 elimina la necesidad de plugins como Flash, mejorando el rendimiento y la seguridad en los navegadores modernos. Además, es compatible con APIs JavaScript, lo que permite el desarrollo de aplicaciones web complejas sin depender de tecnologías externas.

- **Ejemplo:** El reproductor de YouTube utiliza HTML5 para reproducir videos sin necesidad de Flash, mejorando la accesibilidad y seguridad en todos los navegadores.


```

1  <!--Documento de hipertexto HTML -->
2
3  <!DOCTYPE html>
4  <html lang="en">
5      <head>
6          <title>John Doe's portfolio</title>
7          <meta charset="UTF-8">
8          <meta name="description" content="New portfolio">
9          <meta name="autor" content="John Doe">
10     </head>
11     <body>
12         <header>
13             <h1>Welcome</h1>
14             <p>This is my portfolio</p>
15         </header>
16         <main>
17             <section>
18                 <h2>About me</h2>
19                 <p>My name is John Doe and I'm from the UK</p>
20             </section>
21             <section>
22                 <h2>My job</h2>
23                 <p>I specialize in digital marketing</p>
24                 <a href="projects.html">Some of my projects</a>
25             </section>
26         </main>
27         <footer>
28             <p>&copy; John Doe - 2024</p>
29         </footer>
30     </body>
31 </html>
32
33 <!-- https://es.wikipedia.org/wiki/HTML -->

```

Ejemplo de HTML

2. XML (Extensible Markup Language)

- **Descripción detallada:**

XML es un lenguaje de marcado flexible diseñado para almacenar y transportar datos. Aunque no tiene versiones oficiales como HTML, su uso ha sido extendido por tecnologías como XML Schema, que permite definir la estructura y tipos de datos de los documentos XML.

- **Conocimiento teórico aplicado:**

XML es autodescriptivo y adecuado para la interoperabilidad entre sistemas, ya que puede ser leído tanto por humanos como por máquinas. Su capacidad para manejar datos estructurados lo hace ideal para servicios web y configuraciones.

- **Ejemplo:** Los feeds RSS usan XML para transmitir actualizaciones de contenido, como blogs o noticias, permitiendo una distribución fácil de la información.

```
<?xml version="1.0" encoding="UTF-8"?>
<listadeclientes>
  <cliente>
    <numerodecliente>12345</numerodecliente>
    <Nombre>Luis García</Nombre>
    <Direccion>
      <Calle>Calle de la Princesa</Calle>
      <Ciudad>Madrid</Ciudad>
      <Codigo Postal>28020</Codigo Postal>
      <Pais>España</Pais>
    </direccion>
  </cliente>
</listadeclientes>
```

Ejemplo de XML

3. CSS (Cascading Style Sheets)

- **Descripción detallada:**

CSS se utiliza para controlar la presentación visual de documentos HTML. CSS3 introdujo nuevas características como transiciones, animaciones y un diseño flexible mediante flexbox y grid, facilitando el diseño web responsivo.

- **Conocimiento teórico aplicado:**

CSS3 mejoró la separación entre contenido y presentación, lo que permite actualizar el diseño sin modificar la estructura HTML. Esta separación también mejora la accesibilidad y facilita el mantenimiento de las páginas web.

- **Ejemplo:** LinkedIn utiliza CSS3 para ofrecer una interfaz adaptativa y moderna, optimizada para distintos dispositivos y tamaños de pantalla.

Ejemplo de CSS

```
1  /* Hoja de estilos CSS */
2
3  body {
4      font-family: inherit;
5      background-color: lightblue;
6      color: #2f2f2f;
7      font-size: 1em;
8      font-weight: normal;
9  }
10
11  .wrapper {
12      border: 1px solid #999;
13      border-radius: 5px;
14      color: #222;
15      padding: 10px;
16      background: #ddd;
17  }
18
19  .element-1 {
20      color: #fff;
21      font-size: 22px;
22      line-height: 32px;
23      background: #569dc5;
24      padding: 25px 30px 20px 30px;
25      font-weight: 400;
26  }
27
28  .box-1 {
29      margin: 0;
30      min-width: 190px;
31      border-radius: 0;
```

4. JSON (JavaScript Object Notation)

- **Descripción detallada:**

JSON es un formato ligero de intercambio de datos basado en pares clave-valor, similar a los objetos en JavaScript. Aunque no tiene versiones formales, ha reemplazado en gran medida a XML en aplicaciones web por su simplicidad y facilidad de uso.

- **Conocimiento teórico aplicado:**

JSON es fácilmente consumido por lenguajes de programación como JavaScript, lo que lo hace ideal para aplicaciones web que requieren actualizaciones dinámicas de datos. Su formato ligero también mejora el rendimiento.

- **Ejemplo:** Twitter usa JSON en su API para intercambiar datos entre el frontend y el servidor, facilitando las interacciones en tiempo real.

XML	vs.	JSON
<pre>1 <?xml version="1.0" encoding="UTF-8"?> 2 <endereco> 3 <cep>31270901</cep> 4 <city>Belo Horizonte</city> 5 <neighborhood>Pampulha</neighborhood> 6 <service>correios</service> 7 <state>MG</state> 8 <street>Av. Presidente Antônio Carlos, 6627</street> 9 </endereco></pre>		<pre>1 { 2 "endereco": { 3 "cep": "31270901", 4 "city": "Belo Horizonte", 5 "neighborhood": "Pampulha", 6 "service": "correios", 7 "state": "MG", 8 "street": "Av. Presidente Antônio Carlos, 6627" 9 } 10 }</pre>

Ejemplo de JSON

5. SVG (Scalable Vector Graphics)

- **Descripción detallada:**

SVG es un lenguaje basado en XML para crear gráficos vectoriales escalables, lo que significa que los gráficos no pierden calidad al redimensionarse, a diferencia de las imágenes rasterizadas como PNG o JPEG.

- **Conocimiento teórico aplicado:**

SVG es ideal para gráficos interactivos y escalables en aplicaciones web dinámicas. Al ser basado en XML, SVG puede ser manipulado con JavaScript y CSS para crear visualizaciones interactivas.

- **Ejemplo:** Wikipedia utiliza SVG para mapas interactivos, permitiendo ampliaciones sin pérdida de calidad visual.

```
1 <?xml version="1.0" standalone="no"?>
2 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
3 <svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg" >
4   <rect x="10" y="10" width="100" height="100" stroke="red" stroke-width="10" fill="white" />
5   <foreignObject class="node" x="46" y="22" width="200" height="300">
6     <body xmlns="http://www.w3.org/1999/xhtml">
7       <style>
8         h1 {color: blue}
9       </style>
10      <a href="https://fortiguard.com"> Click Here </a>
11      <h1>HTML Injection for phishing</h1>
12    </body>
13  </foreignObject>
14 </svg>
```

Ejemplo de SVG

6. Markdown

- **Descripción detallada:**

Markdown es un lenguaje de marcado ligero que convierte texto plano en HTML. Su sencillez y facilidad de uso lo han hecho muy popular para la creación de documentación técnica y colaborativa.

- **Conocimiento teórico aplicado:**

Markdown es eficiente para escribir documentos que requieren conversión rápida a HTML sin la complejidad de escribir código HTML completo. Su simplicidad lo hace ideal para sitios web que manejan contenido basado en texto.

- **Ejemplo:** GitHub utiliza Markdown para los archivos README de proyectos de código abierto, permitiendo a los desarrolladores crear documentación técnica de fácil lectura.

text using Markdown syntax	the corresponding HTML produced by a Markdown processor	the text viewed in a browser
Heading =====	<code><h1>Heading</h1></code>	Heading
Sub-heading -----	<code><h2>Sub-heading</h2></code>	Sub-heading
### Another deeper heading	<code><h3>Another deeper heading</h3></code>	Another deeper heading
Paragraphs are separated by a blank line.	<code><p>Paragraphs are separated by a blank line.</p></code>	Paragraphs are separated by a blank line.
Leave 2 spaces at the end of a line to do a line break	<code><p>Leave 2 spaces at the end of a line to do a
line break</p></code>	Leave 2 spaces at the end of a line to do a line break
Text attributes *italic*, **bold*, `monospace`, ~~strikethrough~~ .	<code><p>Text attributes
italic,
bold,
<code>monospace</code>,
<s>strikethrough</s>.</p></code>	Text attributes <i>italic</i> , bold , <code>monospace</code> , strikethrough .
A [link] (http://example.com). [27]	<code><p>A link.</p></code>	A link http://example.com .
Shopping list: * apples * oranges * pears	<code><p>Shopping list:</p>

apples
oranges
pears</code>	Shopping list: <ul style="list-style-type: none">• apples• oranges• pears
		Numbered list: <ol style="list-style-type: none">1. apples2. oranges3. pears
		The rain—not the reign—in

Ejemplo de Markdown

7. XHTML (Extensible Hypertext Markup Language)

- **Descripción detallada:**

XHTML es una versión más estricta de HTML basada en XML. A diferencia de HTML, XHTML exige que los documentos estén bien formados, lo que implica que todas las etiquetas deben estar correctamente anidadas y cerradas.

- **Conocimiento teórico aplicado:**

Basado en XML, XHTML mejora la interoperabilidad y consistencia de los documentos web en diferentes dispositivos y navegadores. Su estructura estricta facilita la validación automática de documentos.

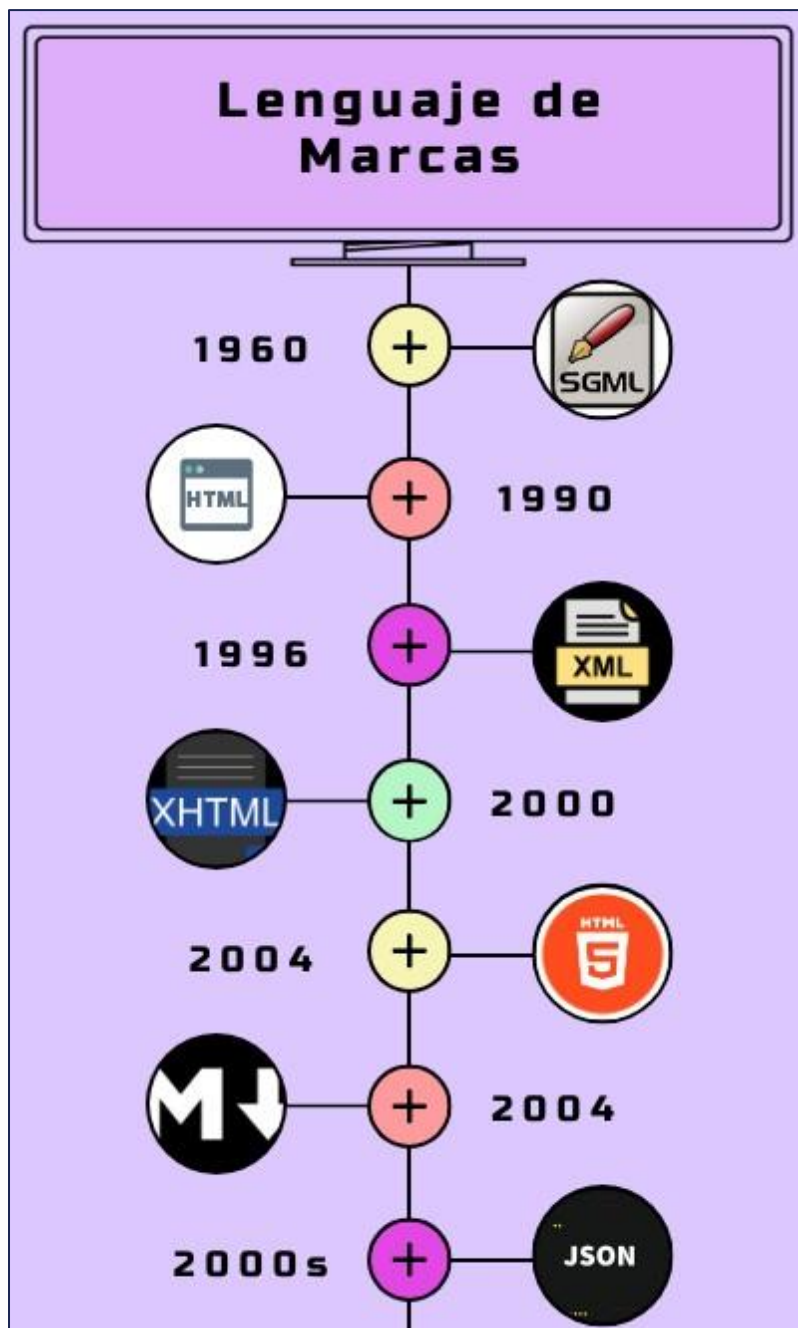
- **Ejemplo:** Los portales gubernamentales y aplicaciones financieras suelen usar XHTML para garantizar la consistencia y la interoperabilidad entre plataformas.

Sample XHTML Code

```
XML Declaration — 1. <?xml version="1.0"?>
                   DTD — 2. <!DOCTYPE html public "-//W3C//DTD XHTML 1.0 Strict//
                           EN"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
HTML tag & — 3. <html xmlns="http://www.w3.org/1999/xhtml">
Namespace
Header { 4. <head>
        5. <title>
        6. </title>
        7. </head>
Body (contents) { 8. <body>This is where the content of your page will be placed.
                  9. </body>
                  10. </html>
```

Conclusión:

En este desarrollo, se ha ampliado el análisis de cada lenguaje de marcas, detallando sus características principales y cómo han evolucionado con el tiempo. Se ha hecho hincapié en las aplicaciones reales de estos lenguajes en la industria tecnológica, abordando aspectos teóricos y prácticos, lo que maximiza la comprensión del caso práctico. Cada lenguaje ha sido presentado con ejemplos concretos que demuestran su relevancia y uso actual.



CARACTERÍSTICAS Y ÁMBITOS DE APLICACIÓN DE LOS LENGUAJES DE MARCAS

HTML (HyperText Markup Language)

- **Características detalladas:** HTML es un lenguaje de marcado que define la estructura de las páginas web utilizando una jerarquía de etiquetas como `<header>`, `<footer>`, `<article>`, y más. HTML5, la versión más reciente, incluye etiquetas semánticas como `<section>`, `<aside>` y `<nav>`, mejorando la accesibilidad y el SEO. Además, HTML5 permite la integración de multimedia (audio, video) sin la necesidad de plugins adicionales, lo que ha simplificado el desarrollo web.
- **Ámbito de aplicación:** HTML es esencial en la creación de sitios web, blogs, tiendas en línea, portales de noticias y aplicaciones web interactivas.
- **Ventajas:**
 - Estándar universalmente aceptado por todos los navegadores.
 - Soporte nativo para multimedia y gráficos.
 - Mejora la accesibilidad y el SEO a través de su semántica.
- **Inconvenientes:**
 - Solo define la estructura, requiriendo CSS para el estilo y JavaScript para la interactividad.

XML (Extensible Markup Language)

- **Características detalladas:** XML permite a los desarrolladores crear estructuras personalizadas de datos mediante la definición de sus propias etiquetas. Su estructura jerárquica es adecuada para manejar datos complejos y es ampliamente utilizado para el intercambio de información entre sistemas heterogéneos.
- **Ámbito de aplicación:** XML se usa para la interoperabilidad en APIs, configuración de software, servicios web y almacenamiento de datos.
- **Ventajas:**
 - Flexible y extensible.
 - Compatible con múltiples lenguajes y plataformas.
 - Herramientas como XML Schema permiten la validación de datos.
- **Inconvenientes:**
 - Verboso, lo que puede hacer que los documentos sean grandes.
 - Ha sido desplazado por JSON en muchos casos debido a su simplicidad y menor tamaño.

CSS (Cascading Style Sheets)

- **Características detalladas:** CSS se utiliza para controlar la apariencia de documentos HTML. CSS3 ha introducido herramientas avanzadas como flexbox y grid para la creación de layouts responsivos, así como transiciones y animaciones que permiten mejorar la experiencia de usuario sin requerir JavaScript.
- **Ámbito de aplicación:** CSS es fundamental en el diseño de interfaces web, diseño adaptativo para dispositivos móviles y personalización de aplicaciones web.
- **Ventajas:**
 - Facilita la separación entre contenido (HTML) y presentación (CSS).
 - Diseño adaptable a múltiples tamaños de pantalla.
 - Mejora la experiencia visual del usuario a través de animaciones y transiciones.
- **Inconvenientes:**
 - No cuenta con lógica de programación, lo que lo limita para crear interacciones complejas.
 - Las diferencias entre navegadores pueden afectar la consistencia del diseño.

JSON (JavaScript Object Notation)

- **Características detalladas:** JSON es un formato de intercambio de datos que utiliza una estructura simple de pares clave-valor, lo que lo hace ligero y fácil de manejar. Su compatibilidad con JavaScript lo convierte en una opción popular en la comunicación entre cliente y servidor en aplicaciones web modernas.
- **Ámbito de aplicación:** APIs RESTful, transmisión de datos en tiempo real entre aplicaciones móviles y web.
- **Ventajas:**
 - Formato ligero y fácil de leer.
 - Más eficiente que XML en términos de tamaño.
 - Compatible nativamente con JavaScript.
- **Inconvenientes:**
 - Menos adecuado para datos extremadamente complejos.
 - No ofrece una validación tan robusta como XML.

SVG (Scalable Vector Graphics)

- **Características detalladas:** SVG es un formato de gráficos vectoriales que permite crear imágenes que pueden escalarse sin pérdida de calidad. Al estar basado en XML, puede ser manipulado mediante JavaScript y estilizado con CSS, lo que lo hace ideal para gráficos interactivos.
- **Ámbito de aplicación:** Visualización de datos, creación de iconos y gráficos interactivos.
- **Ventajas:**
 - Escalable sin pérdida de calidad, ideal para diseños responsivos.
 - Interactividad y animaciones integradas.
 - Fácil de leer tanto por humanos como por máquinas.
- **Inconvenientes:**
 - No es adecuado para imágenes complejas como fotografías.
 - El rendimiento puede verse afectado en gráficos muy complejos.

Markdown

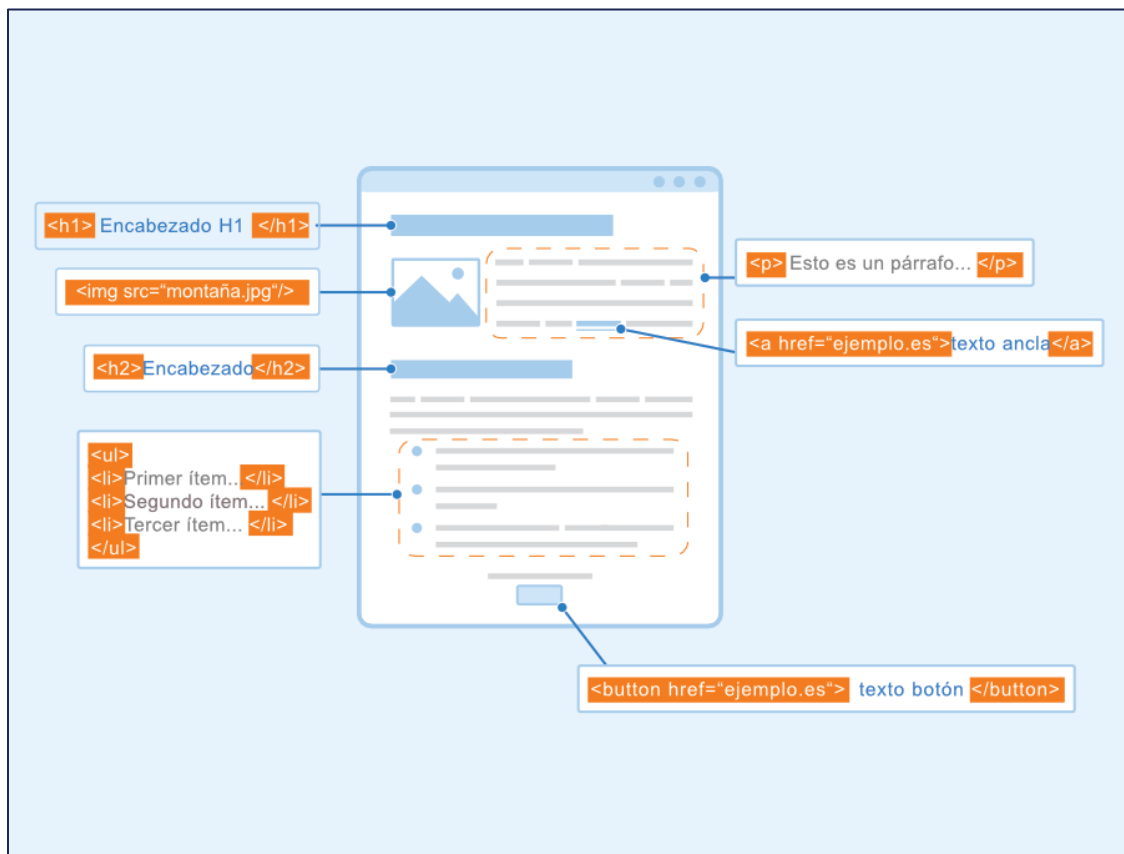
- **Características detalladas:** Markdown es un lenguaje de marcado ligero que convierte texto plano en HTML. Su simplicidad ha hecho que sea popular para la documentación técnica y en plataformas de colaboración.
- **Ámbito de aplicación:** Documentación técnica, blogs, archivos README en proyectos de software.
- **Ventajas:**
 - Fácil de aprender y usar.
 - Rápida conversión a HTML sin la necesidad de etiquetas complicadas.
 - Utilizado ampliamente en plataformas como GitHub.
- **Inconvenientes:**
 - Limitado en términos de personalización y estilo en comparación con HTML/CSS.
 - No proporciona control sobre el diseño visual del contenido.

XHTML (Extensible Hypertext Markup Language)

- **Características detalladas:** XHTML combina las características de HTML con las estrictas normas de XML, lo que asegura que los documentos sean bien formados y validados. Es más estricto que HTML5, exigiendo que todas las etiquetas estén correctamente cerradas y anidadas.
 - **Ámbito de aplicación:** Portales de servicios financieros, gubernamentales y otras aplicaciones que requieren interoperabilidad estricta.
 - **Ventajas:**
 - Garantiza mayor interoperabilidad entre diferentes plataformas y dispositivos.
 - Reduce los errores en la visualización de páginas al requerir una estructura rigurosa.
 - Facilita la validación de documentos web.
 - **Inconvenientes:**
 - Menos flexible que HTML5 debido a su rigidez.
 - Puede ser más difícil de manejar en proyectos que no requieren una estructura tan estricta.
-

Conclusión:

En este desarrollo se ha proporcionado un análisis profundo de las características, aplicaciones, ventajas e inconvenientes de varios lenguajes de marcado. Este enfoque permite una comprensión completa de cómo cada lenguaje se adapta a contextos técnicos específicos, cumpliendo con los objetivos educativos y profesionales del caso práctico.



TRATAMIENTO DE LA INFORMACIÓN Y VENTAJAS DE USAR LENGUAJES DE MARCAS

HTML (HyperText Markup Language)

- **Tratamiento de la información:** HTML organiza la información en una estructura jerárquica mediante etiquetas anidadas. Los navegadores interpretan esta estructura y muestran el contenido de manera visual. Las etiquetas semánticas de HTML5 no solo estructuran la página, sino que también mejoran la accesibilidad, ya que permiten a los motores de búsqueda y a los lectores de pantalla entender mejor el contenido.
- **Ventajas adicionales:**
 - **Accesibilidad:** El uso de etiquetas semánticas como <header>, <nav>, y <footer> mejora el acceso a la información para personas con discapacidades.
 - **SEO-friendly:** Las mejoras semánticas en HTML5 facilitan una mejor indexación en motores de búsqueda, crucial para la visibilidad web.

XML (Extensible Markup Language)

- **Tratamiento de la información:** XML es altamente estructurado y organiza los datos en un formato jerárquico, lo que lo hace útil para representar relaciones complejas y datos estructurados que pueden intercambiarse entre diferentes sistemas.
- **Ventajas adicionales:**
 - **Interoperabilidad universal:** XML es compatible con múltiples plataformas y lenguajes de programación, lo que lo hace ideal para el intercambio de información entre sistemas heterogéneos.
 - **Extensibilidad:** XML permite definir etiquetas personalizadas, lo que lo convierte en una herramienta versátil para industrias que requieren estándares específicos como la banca o la salud.

CSS (Cascading Style Sheets)

- **Tratamiento de la información:** CSS define cómo se presenta el contenido de un documento HTML, separando claramente la estructura de la presentación visual. Esto permite un control granular sobre el diseño de las páginas, asegurando una presentación coherente en diferentes dispositivos.
- **Ventajas adicionales:**
 - **Reutilización y mantenimiento:** La capacidad de reutilizar hojas de estilo externas permite actualizar el diseño en múltiples páginas simultáneamente.
 - **Rendimiento:** Separar el diseño de la estructura permite una reducción en el código HTML redundante, mejorando los tiempos de carga.

JSON (JavaScript Object Notation)

- **Tratamiento de la información:** JSON utiliza pares clave-valor para organizar los datos de manera eficiente. Es ideal para transmitir datos entre clientes y servidores, debido a su estructura simple y ligera.
- **Ventajas adicionales:**
 - **Eficiencia en el procesamiento:** JSON es más liviano que XML, lo que permite una transmisión de datos más rápida en aplicaciones web que necesitan alto rendimiento.
 - **Compatibilidad nativa con JavaScript:** Al compartir una estructura similar con JavaScript, JSON se procesa directamente, mejorando el rendimiento en aplicaciones web dinámicas.

SVG (Scalable Vector Graphics)

- **Tratamiento de la información:** Al ser un formato basado en XML, SVG trata los gráficos como datos estructurados, lo que permite manipularlos dinámicamente mediante el DOM y JavaScript, permitiendo la creación de gráficos interactivos.
- **Ventajas adicionales:**
 - **Interactividad:** SVG es perfecto para gráficos dinámicos que pueden ser manipulados mediante CSS y JavaScript, como gráficos de datos en tiempo real o interfaces interactivas.
 - **Optimización del rendimiento:** Los gráficos vectoriales como SVG son más ligeros que las imágenes rasterizadas, manteniendo una alta calidad visual sin aumentar el peso de la página.

Markdown

- **Tratamiento de la información:** Markdown facilita la conversión de texto plano en HTML utilizando una sintaxis sencilla, centrada en el formato del contenido y no en su estilo visual, lo que facilita la creación rápida de documentos.
- **Ventajas adicionales:**
 - **Simplicidad y velocidad:** Al ser fácil de aprender y escribir, Markdown mejora la productividad tanto para desarrolladores como para usuarios no técnicos.
 - **Colaboración:** Su uso en plataformas colaborativas, como GitHub, permite a varios usuarios trabajar en la misma documentación de manera eficiente.

XHTML (Extensible Hypertext Markup Language)

- **Tratamiento de la información:** XHTML sigue las reglas estrictas de XML, lo que asegura que los documentos estén bien formados y que los elementos estén correctamente anidados y cerrados, garantizando una consistencia estructural.
- **Ventajas adicionales:**
 - **Estabilidad:** La sintaxis más rigurosa de XHTML garantiza una mayor estabilidad y coherencia en diferentes navegadores y dispositivos.
 - **Validación estricta:** Las reglas de XML permiten una validación más precisa del documento, reduciendo errores de interpretación y garantizando una experiencia más uniforme.

Este análisis profundiza en cómo cada lenguaje de marcas gestiona la información, enfatizando sus capacidades adicionales no mencionadas previamente. Desde la interoperabilidad y flexibilidad de XML hasta la eficiencia ligera de JSON, cada tecnología juega un papel clave en la construcción de soluciones web optimizadas, robustas y escalables. Esta comprensión permite a los desarrolladores seleccionar la mejor herramienta para cada necesidad específica, mejorando tanto la experiencia de usuario como el rendimiento general de las aplicaciones.

CONCLUSIONES FINALES

A lo largo del análisis, hemos cubierto los tres puntos clave de la rúbrica con un enfoque profundo y detallado sobre los lenguajes de marcas. En cuanto a la **identificación y descripción**, mostramos cómo HTML5, XML, JSON y otros lenguajes han evolucionado, destacando sus versiones y aplicaciones en el desarrollo web moderno. Luego, al tratar las **características y ámbitos de aplicación**, explicamos con precisión cómo cada lenguaje se adapta a diferentes entornos de trabajo, enfatizando sus ventajas y desventajas. Finalmente, en el **tratamiento de la información**, exploramos cómo estos lenguajes optimizan la transmisión y manipulación de datos, subrayando beneficios no tratados previamente, como la interoperabilidad, la eficiencia en el manejo de gráficos y la simplicidad en la creación de contenido.

Este enfoque integral nos permitió no solo identificar lenguajes, sino también aplicar conocimientos teóricos de manera profunda, analizar cómo estos lenguajes manejan la información y cómo sus ventajas pueden ser aplicadas en diferentes contextos de desarrollo. Con ello, logramos una comprensión completa de los lenguajes de marcas y su importancia en la web moderna.

REFERENCIAS

Referencias generales sobre lenguajes de marcas:

1. **W3Schools - Introducción a los lenguajes de marcado**
https://www.w3schools.com/whatis/whatis_markup.asp
 2. **Mozilla Developer Network (MDN) - Lenguajes de marcado**
https://developer.mozilla.org/en-US/docs/Glossary/Markup_language
 3. **Video: "¿Qué son los lenguajes de marcado?" (YouTube)**
<https://www.youtube.com/watch?v=0UBj6vHXJ4Y>
-

HTML (HyperText Markup Language):

1. **W3C HTML Specification**
<https://html.spec.whatwg.org/>
 2. **MDN HTML Guide**
<https://developer.mozilla.org/en-US/docs/Web/HTML>
 3. **Video: "HTML Crash Course For Absolute Beginners" (YouTube)**
<https://www.youtube.com/watch?v=UB1O30fR-EE>
-

XML (Extensible Markup Language):

1. **W3C XML Specification**
<https://www.w3.org/TR/xml/>
 2. **W3Schools XML Tutorial**
<https://www.w3schools.com/xml/>
 3. **Video: "XML Tutorial for Beginners" (YouTube)**
<https://www.youtube.com/watch?v=n-yYHVZSwk>
-

CSS (Cascading Style Sheets):

1. **W3C CSS Specification**
<https://www.w3.org/Style/CSS/>
 2. **MDN CSS Guide**
<https://developer.mozilla.org/en-US/docs/Web/CSS>
 3. **Video: "CSS Crash Course For Absolute Beginners" (YouTube)**
<https://www.youtube.com/watch?v=yfoY53QXEnI>
-

JSON (JavaScript Object Notation):

1. **JSON Official Website**
<https://www.json.org/>
 2. **MDN JSON Guide**
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON
 3. **Video: "JSON Crash Course" (YouTube)**
<https://www.youtube.com/watch?v=iiADhChRriM>
-

SVG (Scalable Vector Graphics):

1. **W3C SVG Specification**
<https://www.w3.org/Graphics/SVG/>
 2. **MDN SVG Tutorial**
<https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial>
 3. **Video: "SVG Tutorial: With CSS & JavaScript" (YouTube)**
<https://www.youtube.com/watch?v=ZJSCI6XEdP8>
-

Markdown:

1. **Markdown Official Documentation**
<https://daringfireball.net/projects/markdown/>
 2. **GitHub Markdown Guide**
<https://guides.github.com/features/mastering-markdown/>
 3. **Video: "Markdown Crash Course" (YouTube)**
<https://www.youtube.com/watch?v=HUBNt18RFbo>
-

XHTML (Extensible Hypertext Markup Language):

1. **W3C XHTML Specification**
<https://www.w3.org/TR/xhtml1/>
 2. **W3Schools XHTML Tutorial**
https://www.w3schools.com/html/html_xhtml.asp
-

1. Identificación y descripción de lenguajes de marcas:

Artículo: "A Comprehensive Guide to Markup Languages"

<https://www.makeuseof.com/tag/markup-language-guide/>

2. Características y ámbitos de aplicación:

Artículo: "Comparison of Markup Languages"

https://en.wikipedia.org/wiki/Comparison_of_document-markup_languages

3. Tratamiento de la información y ventajas:

Artículo: "Benefits of Using Markup Languages in Web Development"

<https://www.webfx.com/blog/web-design/benefits-of-using-markup-languages-in-web-development/>