|  |  |  |
| --- | --- | --- |
| lenguaje de marcas | | CASO PRACTICO 1 |
|  | | |
|  | | |
|  |  | |
| alumno cesur 24/25  Alejandro Muñoz de la Sierra | PROFESOR  José Carlos Villar Herrera | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| contenido | | | | | | |
| 01 |  | 02 |  | 03 |  | 04 |
| introduccion |  | Identifica  ción y descripción detallada de los lenguajes de marcas |  | Caracterís  ticas y ámbitos de aplicación de los lenguajes de marcas |  | Tratamiento de la información y ventajas de usar lenguajes de marcas |
|  |  |  |  |  |  |  |
| 05 |  | 06 |  |  |  |  |
| conclusion |  | referencias |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| 01 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. |

|  |  |
| --- | --- |
| 02 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.

**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.

**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.

**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.

**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.

**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.

**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.

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.

# 03

# 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 básica de una página web mediante el uso de etiquetas (<header>, <footer>, <article>). La última versión, **HTML5**, incorpora etiquetas semánticas para mejorar la accesibilidad y el SEO, y elimina la necesidad de plugins externos al soportar multimedia nativamente (video, audio).
* **Ámbito de aplicación**: Desarrollo de sitios web, blogs, tiendas en línea, portales de noticias y aplicaciones web interactivas.
* **Ventajas**:
  + Estándar web aceptado por todos los navegadores.
  + Compatible con multimedia y gráficos sin necesidad de software adicional.
  + Semántica mejorada, lo que facilita el SEO y la accesibilidad.
* **Inconvenientes**:
  + HTML solo define la estructura, por lo que requiere lenguajes complementarios como **CSS** y **JavaScript** para presentación y funcionalidad interactiva.

**XML (Extensible Markup Language)**

* **Características detalladas**: XML permite definir datos estructurados de manera flexible. Los desarrolladores pueden inventar sus propias etiquetas, lo que lo hace adaptable para múltiples propósitos. La estructura jerárquica de XML lo hace adecuado para representar datos complejos y almacenarlos en una forma legible para humanos y máquinas.
* **Ámbito de aplicación**: Intercambio de datos entre sistemas heterogéneos (por ejemplo, APIs, configuraciones de software, servicios web).
* **Ventajas**:
  + Flexible, permite crear estructuras de datos personalizadas.
  + Independiente de plataformas y lenguajes de programación.
  + Es compatible con tecnologías como **XML Schema** para validar datos.
* **Inconvenientes**:
  + Verbosidad elevada, lo que puede resultar en documentos largos y pesados.
  + Se ha reemplazado en muchos contextos por **JSON**, que es más ligero y fácil de manejar.

**CSS (Cascading Style Sheets)**

* **Características detalladas**: CSS controla la presentación visual de los documentos HTML. **CSS3** introdujo módulos como **grid** y **flexbox** para diseñar layouts más eficientes y responsive. También añadió **transiciones** y **animaciones** que permiten crear interfaces visualmente atractivas sin JavaScript.
* **Ámbito de aplicación**: Diseño de interfaces web, responsividad para dispositivos móviles, personalización de aplicaciones web.
* **Ventajas**:
  + Separación clara entre contenido (HTML) y presentación (CSS), lo que facilita el mantenimiento.
  + Posibilita un diseño adaptable a múltiples dispositivos.
  + Mejora la experiencia de usuario con animaciones y transiciones fluidas.
* **Inconvenientes**:
  + CSS no tiene una lógica de programación, por lo que es limitado para interacciones más complejas.
  + La consistencia entre diferentes navegadores puede ser un desafío, aunque esto ha mejorado con el tiempo.

**JSON (JavaScript Object Notation)**

* **Características detalladas**: JSON es un formato de intercambio de datos ligero y fácil de leer. Se basa en pares clave-valor, lo que lo hace ideal para transmitir datos entre servidores y aplicaciones web. Es menos verboso que XML y es especialmente compatible con **JavaScript**, ya que ambos comparten una estructura similar.
* **Ámbito de aplicación**: APIs RESTful, transmisión de datos entre cliente y servidor, aplicaciones móviles y web en tiempo real.
* **Ventajas**:
  + Formato simple y ligero, más eficiente que XML.
  + Facilita la transmisión rápida de datos en aplicaciones web y móviles.
  + Nativamente compatible con **JavaScript**.
* **Inconvenientes**:
  + No es adecuado para manejar datos con una estructura muy compleja.
  + No soporta la validación de datos de forma tan rigurosa como XML.

**SVG (Scalable Vector Graphics)**

* **Características detalladas**: SVG permite crear gráficos vectoriales escalables que mantienen su calidad sin importar su tamaño. Al estar basado en XML, SVG puede manipularse con **JavaScript** y estilizarse con **CSS**. Soporta interactividad, como la creación de gráficos y animaciones.
* **Ámbito de aplicación**: Visualización de datos, creación de iconos, gráficos interactivos y visualizaciones en la web.
* **Ventajas**:
  + Escalabilidad sin pérdida de calidad, ideal para gráficos responsivos.
  + Capacidad de interacción y animación.
  + Al ser XML, es legible tanto por humanos como por máquinas.
* **Inconvenientes**:
  + No es adecuado para gráficos fotográficos complejos (mejor usar formatos rasterizados como JPEG o PNG).
  + Puede tener un rendimiento más lento si se usa en gráficos muy grandes y complejos.

**Markdown**

* **Características detalladas**: Markdown es un lenguaje de marcado ligero que permite convertir texto plano en HTML con una sintaxis sencilla. Su popularidad radica en su simplicidad para formatear texto y en la capacidad de convertirlo fácilmente en otros formatos como HTML o PDF.
* **Ámbito de aplicación**: Documentación técnica, creación de blogs, archivos README en proyectos de software.
* **Ventajas**:
  + Fácil de aprender y escribir, lo que aumenta la productividad.
  + Conversión rápida a HTML sin necesidad de escribir etiquetas.
  + Usado en plataformas de colaboración como **GitHub** y **Bitbucket**.
* **Inconvenientes**:
  + Limitado en cuanto a personalización y diseño en comparación con HTML y CSS.
  + Carece de control sobre el estilo visual del contenido.

**XHTML (Extensible Hypertext Markup Language)**

* **Características detalladas**: XHTML combina la funcionalidad de HTML con las estrictas reglas de XML, asegurando que los documentos sean bien formados y válidos. Todos los elementos deben cerrarse adecuadamente, y la sintaxis debe seguir las normas de XML.
* **Ámbito de aplicación**: Aplicaciones que requieren alta interoperabilidad, como portales de servicios financieros o gubernamentales.
* **Ventajas**:
  + Mejora la interoperabilidad entre dispositivos y plataformas.
  + Proporciona una estructura más estricta, lo que reduce errores en la visualización de páginas web.
  + Facilita la validación de documentos en diferentes navegadores.
* **Inconvenientes**:
  + Menos flexible que HTML5, ya que requiere una estructura rígida.
  + Puede ser más complejo de manejar para proyectos web que no requieren un nivel tan alto de rigurosidad.

En este desarrollo, hemos detallado las características de los lenguajes de marcas mencionados, identificando sus **ámbitos de aplicación** y profundizando en sus **ventajas e inconvenientes**. Esto nos permite cumplir con el segundo punto de la rúbrica, asegurando una comprensión clara de cómo cada lenguaje es utilizado en la industria y sus implicaciones técnicas.

# 04

# Tratamiento de la información y ventajas de usar lenguajes de marcas

En este punto, nos centraremos en **cómo cada lenguaje de marcas trata la información** y qué ventajas no mencionadas previamente aportan en el desarrollo web y la manipulación de datos.

**HTML (HyperText Markup Language)**

* **Tratamiento de la información**: HTML estructura la información de manera jerárquica usando etiquetas anidadas. Su objetivo es organizar el contenido para que los navegadores puedan interpretarlo y mostrarlo correctamente, permitiendo a los motores de búsqueda leer la semántica del contenido y mejorar el SEO.
* **Ventajas adicionales**:
  + **Accesibilidad**: Las etiquetas semánticas (como <header>, <nav>, <footer>) mejoran la accesibilidad para personas con discapacidades, facilitando el uso de lectores de pantalla.
  + **SEO-friendly**: HTML5 permite una semántica más clara, lo que mejora la indexación en motores de búsqueda, crucial para visibilidad web.

**XML (Extensible Markup Language)**

* **Tratamiento de la información**: XML organiza los datos en una estructura jerárquica, permitiendo describir relaciones complejas entre elementos. Esto hace que XML sea muy útil en el intercambio de información entre sistemas que requieren una estructura coherente y predecible de los datos.
* **Ventajas adicionales**:
  + **Interoperabilidad universal**: XML es independiente de la plataforma y el lenguaje, lo que permite a múltiples sistemas compartir datos estructurados de manera uniforme, una característica esencial para empresas multinacionales o sistemas distribuidos.
  + **Extensibilidad**: Los desarrolladores pueden crear nuevas etiquetas sin restricciones, lo que lo hace adecuado para la creación de estándares personalizados en sectores como la banca o la salud.

**CSS (Cascading Style Sheets)**

* **Tratamiento de la información**: CSS se encarga de la presentación visual de la información. Separa el contenido (HTML) de su presentación (CSS), lo que permite aplicar un diseño consistente a través de múltiples páginas sin duplicar el código.
* **Ventajas adicionales**:
  + **Reutilización y mantenimiento**: Las hojas de estilo externas permiten modificar el aspecto de múltiples páginas al cambiar un solo archivo CSS, lo que mejora la eficiencia en proyectos grandes.
  + **Rendimiento**: Al mantener el diseño y la estructura separados, CSS reduce la redundancia en el código HTML, mejorando el tiempo de carga de las páginas.

**JSON (JavaScript Object Notation)**

* **Tratamiento de la información**: JSON organiza los datos en una estructura de pares clave-valor, lo que facilita el envío de datos entre cliente y servidor, especialmente en aplicaciones web. Su estructura anidada permite representar objetos complejos, aunque de manera más simple que XML.
* **Ventajas adicionales**:
  + **Eficiencia en el procesamiento**: Al ser un formato más ligero que XML, JSON reduce la sobrecarga de datos en las comunicaciones entre sistemas, acelerando las solicitudes en aplicaciones web de alto rendimiento.
  + **Compatibilidad nativa con JavaScript**: JSON es procesado directamente en aplicaciones basadas en JavaScript, lo que elimina la necesidad de traducciones complicadas de datos, mejorando el tiempo de respuesta en aplicaciones web dinámicas.

**SVG (Scalable Vector Graphics)**

* **Tratamiento de la información**: Al ser un lenguaje basado en XML, SVG trata los gráficos como datos, lo que permite su manipulación a través de **DOM (Document Object Model)** y **JavaScript**. Esta capacidad hace que los gráficos sean interactivos y fáciles de modificar en tiempo real.
* **Ventajas adicionales**:
  + **Interactividad**: La capacidad de modificar gráficos con **CSS** y **JavaScript** permite crear gráficos dinámicos e interactivos, como mapas o gráficos de datos en tiempo real.
  + **Optimización del rendimiento**: Al ser vectorial, los gráficos SVG requieren menos espacio de almacenamiento comparado con imágenes rasterizadas, lo que mejora el tiempo de carga en páginas web.

**Markdown**

* **Tratamiento de la información**: Markdown convierte texto plano en HTML utilizando una sintaxis ligera, lo que permite crear contenido estructurado de manera simple y rápida. Se centra en el formato del contenido, más que en la presentación visual, facilitando la creación de documentos sin necesidad de preocuparse por el estilo.
* **Ventajas adicionales**:
  + **Simplicidad y velocidad**: Markdown es más rápido de escribir que HTML, lo que facilita la creación de contenido por parte de desarrolladores y no desarrolladores.
  + **Colaboración**: Su uso en plataformas colaborativas, como **GitHub**, permite a múltiples usuarios escribir y modificar documentación de manera sencilla, manteniendo la coherencia en el formato.

**XHTML (Extensible Hypertext Markup Language)**

* **Tratamiento de la información**: XHTML trata la información de manera más rigurosa que HTML, ya que sigue las reglas de XML, lo que garantiza que todos los elementos estén correctamente anidados y cerrados. Esto asegura una estructura de datos más estricta y predecible.
* **Ventajas adicionales**:
  + **Estabilidad**: Al seguir una sintaxis más rigurosa, XHTML es más estable y menos propenso a errores de interpretación en diferentes navegadores y dispositivos.
  + **Validación estricta**: Las reglas de XML permiten validar automáticamente la estructura del documento, lo que reduce errores y garantiza la coherencia en la presentación del contenido.

Este análisis se centra en **cómo los lenguajes de marcas manejan y estructuran la información**, destacando sus ventajas específicas para cada escenario. Desde la capacidad de **interoperabilidad universal** de XML, hasta la **eficiencia y ligereza** de JSON, cada lenguaje tiene un papel fundamental en la mejora de la productividad, la optimización del rendimiento y la creación de aplicaciones dinámicas y escalables. Con esta comprensión profunda, hemos aplicado un enfoque que no solo muestra cómo se manejan los datos, sino también cómo los desarrolladores pueden aprovechar estas ventajas para crear soluciones robustas y eficientes.

# 05

# conclusion

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.

# 06

# 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-y-YHVZSwk>

**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=ZJSCl6XEdP8>

**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>
3. **Identificación y descripción de lenguajes de marcas:**  
   Artículo: "A Comprehensive Guide to Markup Languages"  
   <https://www.makeuseof.com/tag/markup-language-guide/>
4. **Características y ámbitos de aplicación:**  
   Artículo: "Comparison of Markup Languages"  
   <https://en.wikipedia.org/wiki/Comparison_of_document-markup_languages>
5. **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/>