

Actividad: Explorando los Lenguajes de Marcas (RA1)

Objetivo: Comprender el uso, la estructura y la importancia de los lenguajes de marcas, aplicando estos conocimientos de manera reflexiva y práctica en escenarios reales.

Parte 1: Conceptos Generales

Situación:

Estás en una **entrevista de trabajo** para un puesto de desarrollador web. El entrevistador te pide que expliques qué es un **lenguaje de marcas** y por qué es importante en el desarrollo web.

Instrucciones:

1. **Explica** con tus propias palabras qué es un **lenguaje de marcas** y cómo se diferencia de otros tipos de lenguajes como los de programación.

El lenguaje de marcas es un lenguaje mas visual o orientado ha hacer que añadan listas

2. Imagina que estás explicando esto a alguien que no tiene conocimientos técnicos. ¿Cómo harías que lo entienda fácilmente?

Le explicaría un código para que vea como se aplica el lenguaje de marcas

Punto de reflexión:

- Piensa en un caso práctico en tu vida diaria donde hayas interactuado con un lenguaje de marcas (por ejemplo, al rellenar un formulario en línea). **Descríbelo** en detalle.

Para ir a moodle por ejemplo al interactuar con google ya estoy interactuando con lenguaje de marcas



Parte 2: Clasificación de Lenguajes

Situación:

Te han asignado la tarea de crear una **guía rápida** para nuevos desarrolladores que acaban de unirse a tu equipo sobre los diferentes tipos de lenguajes de marcas.

Instrucciones:

1. **Clasifica** los lenguajes de marcas en **lenguajes de presentación** (como HTML) y **lenguajes descriptivos** (como XML o JSON).
2. Da un ejemplo de cada tipo y **explica sus diferencias** de manera clara.

Lenguajes utilizados en la presentación

Los lenguajes de presentación se utilizan para mostrar el contenido. Se enfocan en la forma y el aspecto de la presentación.

HTML crea la base de una página web, que incluye elementos como párrafos, encabezados, enlaces, imágenes y otros.

Al controlar el diseño y el formato de las páginas web, CSS permite la aplicación de estilos como colores, fuentes y márgenes, entre otros.

TeX: se utiliza para crear documentos de alta calidad tipográfica, especialmente en matemáticas y ciencias.

lenguajes descriptivos

Propósito: La estructura y el significado del contenido se describen utilizando lenguajes descriptivos, independientemente de su presentación visual. Se benefician cuando se transfieren datos entre sistemas.

XML, JSON y YAML, por ejemplo.

Punto de reflexión:

- Reflexiona sobre cuál de los dos tipos de lenguaje es **más importante** para un desarrollador web y **por qué**. ¿Crees que alguno es más relevante para ciertos proyectos?

Opino que los mejores son los de presentación ya que una web necesitas que se vea de forma presentable y no solo un pegote de texto



Parte 3: Herramientas de Edición

Situación:

Como **líder técnico**, uno de tus compañeros te pide que le recomiendes herramientas para trabajar con **XML** y **JSON**.

Instrucciones:

1. **Recomienda** dos herramientas que consideres útiles para trabajar con **XML** y **JSON**, explicando por qué son efectivas.
Recomendaría visual studio code ya que es muy cómodo de usar y fácil de usar, por otro lado Net beans es otra buena opción para usar ya que tiene una interfaz cómoda aunque un poco lisa de usar si eres nuevo
2. Crea un pequeño fragmento de **código XML** que represente un libro y **valida su estructura** utilizando una herramienta en línea (como https://www.w3schools.com/xml/xml_validator.asp o <https://validator.w3.org/>).
3. Describe el **proceso de validación**. ¿Tuviste algún problema al intentar validarlo? ¿Cómo lo solucionaste?

www.w3schools.com dice

No errors found

Aceptar

Daniel García Brun

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Daniel</to>
  <from>Jani</from>
  <heading>Hola</heading>
  <body>Wow</body>
</note>
```

Daniel García Brun

Punto de reflexión:

- ¿Por qué crees que es importante **validar la estructura** de un documento XML antes de usarlo en una aplicación web?

Para que no colapse ni de errores la web



Parte 4: Espacios de Nombres en XML

Situación:

Estás integrando dos sistemas diferentes que usan XML, pero ambos utilizan la etiqueta <precio> en diferentes contextos (hardware y software).

Instrucciones:

1. **Explica** qué son los **espacios de nombres en XML** y cómo te ayudarían a resolver el problema (Buscar en la teoría).

Sirve para que no colapsen las etiquetas entre si y de esta forma funcione

2. Crea un documento XML con información sobre **tres libros** y sus autores, utilizando **espacios de nombres** para diferenciar dos categorías (por ejemplo, ciencia ficción y ciencia).

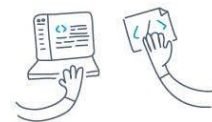
```
<?xml version="1.0" encoding="UTF-8"?>
<libros xmlns:cf="cienciaficción" xmlns:f="fantasia" xmlns:so="Space Opera">
  <cf:libro>
    <cf:titulo>Proyecto Hail Mary</cf:titulo>
    <cf:autor>Andy Weir</cf:autor>
  </cf:libro>
  <f:libro>
    <f:titulo>El camino de los reyes</f:titulo>
    <f:autor>Brandon Sanderson</f:autor>
  </f:libro>
  <so:libro>
    <so:titulo>El juego de ender</so:titulo>
    <so:autor>Orson Scott Card</so:autor>
  </so:libro>
</libros>
```

Daniel García Brun

Punto de reflexión:

- ¿Qué ventajas ofrecen los **espacios de nombres** al trabajar con grandes volúmenes de datos en XML?

Para que el código se vea de forma más clara y fácil de leer



Parte 5: Introducción a JSON

Situación:

Un cliente te ha pedido que **conviertas su sistema de XML a JSON** para hacer que su sitio web sea más ligero y rápido.

Instrucciones:

1. **Explica** en tus propias palabras las ventajas de JSON sobre XML en el desarrollo de aplicaciones web.

JSON es la mejor opción ya que es lo favorito de los programadores al ser compatible con java script

2. Convierte el documento XML que creaste en la **Parte 4** a **formato JSON**.



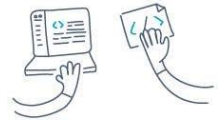
```
{
  "libros": [
    {
      "cf:libro": {
        "cf:titulo": "Proyecto Hail Mary",
        "cf:autor": "Andy Weir"
      }
    },
    {
      "f:libro": {
        "f:titulo": "El camino de los reyes",
        "f:autor": "Brandon Sanderson"
      }
    },
    {
      "so:libro": {
        "so:titulo": "El juego de ender",
        "so:autor": "Orson Scott Card"
      }
    }
  ]
}
```

Daniel Garcia Brun

Punto de reflexión:

- ¿Qué formato te parece **más fácil de trabajar**, JSON o XML, y por qué?

Opimo que JSON ya que por ejemplo las variables no has de ir a declararlas y tal las puedes crear como en HTML



Parte 6: Comparación entre XML y JSON

Situación:

En una reunión de equipo, surge un debate sobre si usar **XML** o **JSON** para desarrollar una nueva API.

Instrucciones:

1. **Compara las estructuras** de tu documento en XML y JSON (creados en la Parte 5).
2. Responde: ¿Cuál de los dos te parece más **legible** y **fácil de entender**?

Opino que los dos por igual JSON y XML me parecen sencillos de leer

3. Enumera tres **ventajas** de JSON sobre XML.

JSON crea archivos mas reducidos de tamaño, aparte de adaptarse de forma nativa con JavaScript y tiene una sintaxis más simple y fácil de leer

Punto de reflexión:

- ¿En qué situaciones crees que sería más útil usar **XML** y en cuáles **JSON**? (Aunque todavía no tienes mucha experiencia, trata de desarrollar una respuesta basándote en lo que has aprendido en clase y la opinión que has formado al respecto.)

Xml es más para representar datos y estructuras complejas, por otro lado JSON es perfecto para intercambiar datos de apps webs y apis

Parte 7: YAML para Configuraciones

Situación:

Estás configurando un servidor para un cliente y descubres que el archivo de configuración está escrito en **YAML** en lugar de XML o JSON.

Instrucciones:

1. **Explica** por qué **YAML** es una opción popular para los archivos de configuración, comparado con XML o JSON.

Yaml se creó después y tuvieron en cuenta y cuidado para que fuese más fácil de leer y entender a diferencia de sus competidores

2. Da un ejemplo de situación donde crees que **YAML** sería más útil que XML o JSON.

A la hora de leerlo es mas legible y mas simple, aparte de ser archivos más compactos

Punto de reflexión:

- ¿**YAML** te gusta cómo formato?

Creo que es un lenguaje más o menos fácil de entender, pero opino que json es más fácil de entender



Parte 8: Creación de una Página Web Básica en HTML

Situación:

Estás empezando en el mundo del desarrollo web y te han pedido que crees tu **primera página HTML**. Este ejercicio te ayudará a comprender la estructura básica de una página web.

Instrucciones:

1. Crea un **documento HTML** muy básico utilizando las etiquetas `<html>`, `<head>`, `<body>`, `<h1>`, y `<p>`.
2. Escribe un **título** para la página (utilizando la etiqueta `<title>`) y un **encabezado** (`<h1>`) que diga "Mi Primera Página Web".
3. En el **cuerpo** de la página (`<body>`), añade un párrafo (`<p>`) que describa algo sobre ti o tu día favorito.



```

index.html
Ejemplo web > index.html > html > body > div.titulo > h1.subtitulo
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>M4</title>
7      <link rel="stylesheet" href="./css/styles.css">
8  </head>
9  <body>
10     <h1 id="t1">Mi Primera Página Web</h1>
11     <div class="titulo">
12         <h1 class="subtitulo">Me llamo Daniel y creo que ya me conoces Roberto, vengo de smx en Monlau y ahora
13     </div>
14     <div class="titulo">
15         <h2 class="subtitulo">Subtitulo</h2>
16     </div>
17 </body>
18 </html>

```

Daniel García Brun

Punto de reflexión:

- ¿Has encontrado dificultad en crear tu primera página web? ¿Qué elementos crees que son esenciales para una página básica?

No me ha resultado difícil, opino que es fácil de crear y opino que lo más básico a de ser un header, body y footer aparte del css

Parte 9: Estructuración de Datos con XML

Situación:

Estás trabajando con un equipo de desarrolladores que necesita **almacenar datos** sobre una colección de libros. Te piden que crees un archivo XML para estructurar la información de manera clara.

Instrucciones:

1. Crea un archivo **XML** que incluya una lista de al menos **tres libros**, con la siguiente información para cada uno:
 - Título (`<titulo>`)
 - Autor (`<autor>`)



- Año de publicación (<año>)
2. Asegúrate de que el archivo siga las reglas de un **XML bien formado**, con todas las etiquetas correctamente abiertas y cerradas.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <libros xmlns:cf="cienciaficcion" xmlns:f="fantasia" xmlns:so="Space Opera">
3      <cf:libro>
4          <cf:titulo>El ritmo de la guerra</cf:titulo>
5          <cf:autor>Brandon Sanderson</cf:autor>
6          <cf:año_publicacion>17 de noviembre de 2020</cf:año_publicacion>
7      </cf:libro>
8      <cf:libro>
9          <cf:titulo>Rekoners calamity</cf:titulo>
10         <cf:autor>Brandon Sanderson</cf:autor>
11         <cf:año_publicacion>16 de febrero de 2016</cf:año_publicacion>
12     </cf:libro>
13     <cf:libro>
14         <cf:titulo>Yumi y el pintor de pesadillas</cf:titulo>
15         <cf:autor>Brandon Sanderson</cf:autor>
16         <cf:año_publicacion>1 de julio de 2023</cf:año_publicacion>
17     </cf:libro>
18 </libros>

```

Daniel García Brun

Task ID	Job ID	Source	Destination	Status	Start time	End time
1	1	This local mach...	DAM-30.monl...	1%	10/3/24 10:09...	Estimated time remaining: 1 hours and 3 minutes

Daniel García Brun

Punto de reflexión:

- ¿Te parece más fácil estructurar datos en XML o en otros formatos?
Explica tu experiencia.

Opino que xml tiene una estructura muy sencilla y es muy similar a la estructura de html



Parte 10: Creación y Validación de un Documento JSON

Situación:

En un proyecto de desarrollo web, necesitas intercambiar información entre el navegador y el servidor utilizando **JSON**. El primer paso es crear un documento básico de datos en formato JSON.

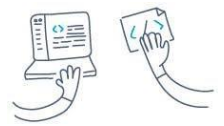
Instrucciones:

1. Crea un archivo **JSON** que contenga información sobre **tres personas**, con los siguientes detalles para cada una:
 - Nombre ("nombre")
 - Edad ("edad")
 - Ciudad ("ciudad")

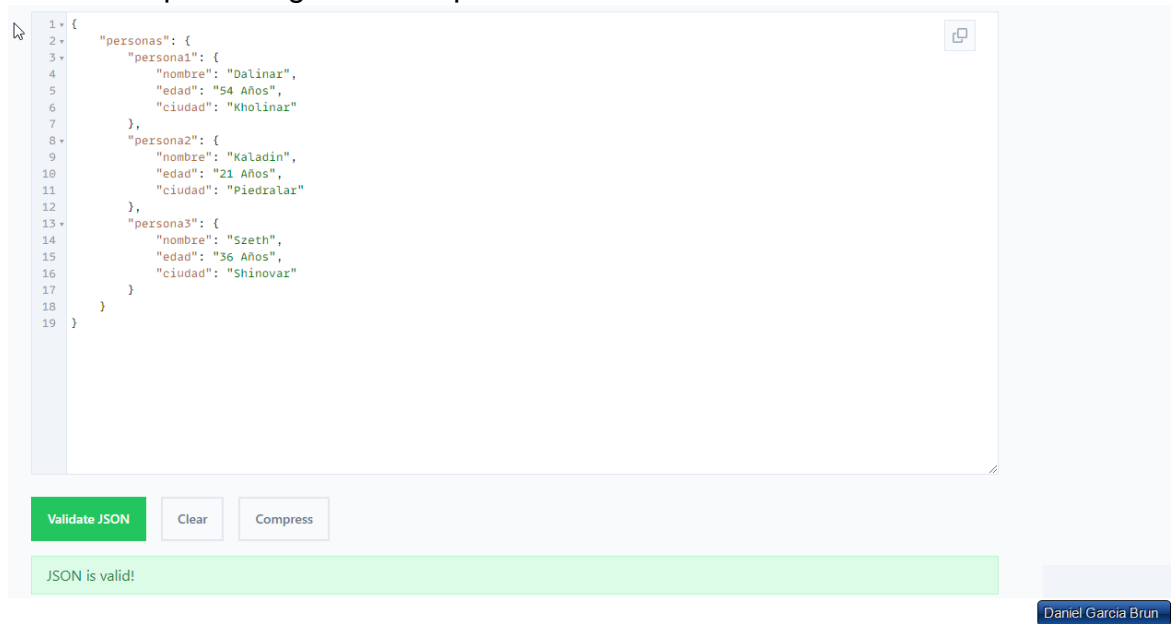
```
{
  "personas": {
    "persona1": {
      "nombre": "Dalinar",
      "edad": "54 Años",
      "ciudad": "Kholinar"
    },
    "persona2": {
      "nombre": "Kaladin",
      "edad": "21 Años",
      "ciudad": "Piedralar"
    },
    "persona3": {
      "nombre": "Szeth",
      "edad": "36 Años",
      "ciudad": "Shinovar"
    }
  }
}
```

Daniel García Brun

○



2. Valida el archivo JSON utilizando una herramienta en línea como **JSONLint** para asegurarte de que está bien formado.



The screenshot shows the JSONLint website interface. A text area contains a JSON object with three persons. Below the text area are buttons for 'Validate JSON', 'Clear', and 'Compress'. A green message bar at the bottom states 'JSON is valid!'. A small blue button with the name 'Daniel Garcia Brun' is in the bottom right corner.

```
1+ {
2+   "personas": {
3+     "persona1": {
4+       "nombre": "Dalinar",
5+       "edad": "54 Años",
6+       "ciudad": "Kholinar"
7+     },
8+     "persona2": {
9+       "nombre": "Kaladin",
10+      "edad": "21 Años",
11+      "ciudad": "Piedralar"
12+    },
13+    "persona3": {
14+      "nombre": "Szeth",
15+      "edad": "36 Años",
16+      "ciudad": "Shinovar"
17+    }
18+  }
19+ }
```

Validate JSON Clear Compress

JSON is valid!

Daniel Garcia Brun

Punto de reflexión:

- ¿Qué te ha parecido más sencillo: trabajar con JSON o XML? Explica por qué.

El mas sencillo me ha parecido xml debido a que ya estoy familiarizado con html