

# HITO 1 - 2o TRIMESTRE LM

---

» XML Y JSON



# PARTE 1

## 1. INFORMACIÓN SOBRE XML

XML, siglas en inglés de eXtensible Markup Language, traducido como 'Lenguaje de Marcado Extensible' o 'Lenguaje de Marcas Extensible', es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

XML proviene de un lenguaje inventado por IBM en los años 1970, llamado GML (Generalized Markup Language), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información y compartirla en otros SO y plataformas. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (Standard Generalized Markup Language), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información.

Debe seguir estas reglas al crear la sintaxis XML:

- Todos los elementos XML deben tener una etiqueta de cierre.
- Las etiquetas XML son sensibles a mayúsculas y minúsculas.
- Todos los elementos XML deben estar debidamente anidados.
- Todos los documentos XML debe tener un elemento raíz.
- Los valores de atributo siempre deben estar entre comillas.

## 2. INFORMACIÓN DE JSON

JSON (acrónimo de JavaScript Object Notation, 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que resulta mucho más sencillo escribir un analizador sintáctico (parser) para él.<sup>1</sup> En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, algo que (debido a la ubicuidad de JavaScript en casi cualquier navegador web) ha sido fundamental para que haya sido aceptado por parte de la comunidad de desarrolladores AJAX.

Reglas de sintaxis JSON que hay que conocer:

- Todos los datos del archivo deben estar rodeados de llaves `{}` si quieres representar un objeto y entre corchetes cuadrados si es un arreglo `[]`.
- Las comillas simples no están permitidas.
- La llave en cada JSON debe ser única y debe estar entre comillas dobles.
- Los números no deben ir entre comillas dobles, de lo contrario se tratarán como cadenas de texto.
- El tipo de dato `null` no debe ir entre comillas dobles.

- Los valores booleanos solo pueden ser verdaderos o falsos.
- Cada par llave:valor debe terminar con una coma, excepto el último elemento
- Un solo objeto dentro de un arreglo debe terminar también con una coma.

## PARTE 2

En esta parte de la fase uno te explicaré las partes del código que he realizado para llevar a cabo mi página web.

Empezaré explicándote la parte del código de la pagina principal, aportando capturas de ello.

### 1-ESTRUCTURA.HTML

#### 1. Head y conexiones

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
7      <link rel="stylesheet" href="estilo.css">
8      <title>Tu Blog de Programación</title>
9  </head>

```

Empezamos con este código el cual se encargará del empleo de bootstrap para que la página pueda quedar bien estructurada según mis necesidades.

Por otro lado tenemos la conexión al css para hacer todo más bonito a gusto personal.

Y ya por último el título de la página entre las llaves <title></title>.

```

12  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
13      <a class="navbar-brand" href="#">Inicio</a>
14      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
15          <span class="navbar-toggler-icon"></span>
16      </button>
17      <div class="collapse navbar-collapse" id="navbarNav">
18          <ul class="navbar-nav">
19              <li class="nav-item">
20                  <a class="nav-link" href="xml.html">XML</a>
21              </li>
22              <li class="nav-item">
23                  <a class="nav-link" href="json.html">JSON</a>
24              </li>
25              <li class="nav-item">
26                  <a class="nav-link" href="contacto.html">Contacto</a>
27              </li>
28          </ul>
29      </div>
30  </nav>

```

#### 2. Navbar

Este código se encarga de hacer una barra de navegación en la parte superior de la página web. Haremos unas conexiones a otras páginas mediante el empleo de los href, como podemos observar repetido en 3 ocasiones.



Esta sería la barra, a continuación, te enseñare el css correspondiente a ella, que en este caso sería únicamente para ajustarla.

```

1  body {
2    padding-top: 56px;
3  }
4

```

### 3. Resto del body

```

32  <div class="container-fluid">
33    <div class="row">
34      <div class="col-lg-3">
35        <br>
36        <div class="sidebar">
37          <h4>Categorías</h4>
38          <br>
39          1. Lenguajes de Programación:
40          <br>
41          <li>Python, JavaScript, Java, C++, Swift, Ruby</li>
42
43          2. Desarrollo Web:
44
45          <li>HTML/CSS, Frontend (React, Angular), Backend (Django, Spring) </li>
46          <br>
47          <h4>Últimos Posts</h4>
48          <br>
49          1. "Guía completa en Python"
50          <br>
51          <li>Una introducción paso a paso para aquellos que recién comienzan con Python.</li>
52          <br>
53          2. "Cómo optimizar el rendimiento en aplicaciones React"
54          <br>
55          <li>Consejos y buenas prácticas para mejorar el rendimiento en proyectos React.</li>
56          <br>
57          <h4>Síguenos en Redes Sociales</h4>
58          <br>
59          <div class="social-icons">
60            <a href="https://www.facebook.com/?locale=es_ES" class="social-icon"></a>
61            <a href="https://twitter.com/?lang=es" class="social-icon"></a>
62            <a href="https://www.instagram.com/" class="social-icon"></a>
63            <a href="https://www.tiktok.com/es/" class="social-icon"></a>
64          </div>
65        </div>
66      </div>
67    </div>
68  </div>
69

```

Aquí he realizado un container el cual voy a dividir posteriormente en dos grandes `<div class="col">`, uno de 3 y otro de 9, con la finalidad de dividir mi página en dos columnas principales, siendo una más grande que la otra.

En el primer `col=3` hemos realizado una sidebar, la cual constaría de 3 apartados principales los cuales van entre `<h4></h4>`, y la información que quisiera que llevara delante un punto, la he metido entre las llaves `<li></li>`.

Finalmente, en el último `<h4>` el cual va de redes sociales, he querido meter los iconos de las 4 más usadas para de esta manera mediante un href se pueda acceder directamente a dichas aplicaciones y así obtener información sobre dicha página en ellas. El `img src`, metiendo el png de la imagen es lo que se encarga de que sea visible.

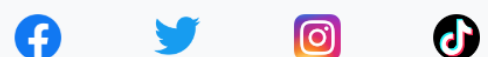
#### Categorías

1. Lenguajes de Programación:
  - Python, JavaScript, Java, C++, Swift, Ruby
2. Desarrollo Web:
  - HTML/CSS, Frontend (React, Angular), Backend (Django, Spring)

#### Últimos Posts

1. "Guía completa en Python"
  - Una introducción paso a paso para aquellos que recién comienzan con Python.
2. "Cómo optimizar el rendimiento en aplicaciones React"
  - Consejos y buenas prácticas para mejorar el rendimiento en proyectos React.

#### Síguenos en Redes Sociales



Por aquí tendríamos el css que le da forma y personaliza la sidebar que se puede ver arriba.

Tenemos por ejemplo el color del fondo con ese background-color, los márgenes por arriba, abajo, lateralmente, con esos margin.

Por ejemplo en el sidebar h4, modificamos lo que esta entre los <h4></h4>, y en este caso le metemos una línea de color negro y un padding a gusto personal.

```
6  .sidebar {
7      background-color: #ffffff;
8      border-radius: 8px;
9      padding: 20px;
10     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
11 }
12
13 .sidebar h4 {
14     border-bottom: 2px solid #000000;
15     padding-bottom: 10px;
16 }
17
18 .sidebar a {
19     display: block;
20     color: #1479e6;
21     margin-bottom: 10px;
22     text-decoration: none;
23     transition: color 0.3s;
24 }
25
26 .sidebar a:hover {
27     color: #0056b3;
28 }
29
30 .sidebar p {
31     margin-bottom: 20px;
32 }
33
34 .sidebar .social-icons {
35     display: flex;
36     justify-content: space-between;
37 }
38
39 .sidebar .social-icons a {
40     color: #007bff;
41     font-size: 20px;
42     transition: color 0.3s;
43 }
44
45 .sidebar .social-icons a:hover {
46     color: #0056b3;
47 }
```

```

72 <div class="col-lg-9">
73 <div class="content">
74 <br>
75 <div class="row">
76 <div class="col-md-6">
77 <div class="card mb-4">
78 <div class="card-body">
79 <h2 class="card-title">XML</h2>
80 
81 <p class="card-text">
82 Bienvenidos a la mejor pagina de información sobre XML, aqui encontrarás todo lo que necesites y más.
83 </p>
84 <div class="button-borders">
85 <a href="xml.html" class="primary-button">Más sobre xml</a>
86 </div>
87 </div>
88 </div>
89 </div>
90 <div class="col-md-6">
91 <div class="card mb-4">
92 <div class="card-body">
93 <h2 class="card-title">JSON</h2>
94 
95 <p class="card-text">
96 Bienvenidos a la mejor pagina de información sobre JSON, aqui encontrarás todo lo que necesites y más.
97 </p>
98 <div class="button-borders">
99 <a href="json.html" class="primary-button">Más sobre json</a>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>

```

Por otro lado, tenemos la otra columna de tamaño 9, dentro de la cual seguiremos jugando con las columnas y los tamaños. Como vemos dentro de la columna grande de tamaño 9, la hemos dividido en otras dos columnas de 6, en la que cada una haremos dos card mb-4, para que no ocupe la columna entera. Meteremos un titulo en cara card con el card-title entre `<h2></h2>`.

Para complementar el card he metido en la parte de abajo un button con un href a las demás páginas, en el card de xml para la pagina donde viene la información de xml y en la card de json para la página que tiene información sobre json.

## XML



Bienvenidos a la mejor pagina de información sobre XML, aqui encontrarás todo lo que necesites y más.

[Más sobre xml](#)

## JSON



Bienvenidos a la mejor pagina de información sobre JSON, aqui encontrarás todo lo que necesites y más.

[Más sobre json](#)

```

<div class="row">
  <div class="col-md-12">
    <div class="card">
      <div class="card-body">
        <h2 class="card-title">Más Información</h2>
        <p class="card-text">
          Explora estos recursos para practicar y obtener información educativa adicional:
        </p>
        <ul>
          <li>
            <a href="https://leetcode.com/" target="_blank" rel="noopener noreferrer">LeetCode - Desafíos de Programación</a>
          </li>
          <li>
            <a href="https://developer.mozilla.org/" target="_blank" rel="noopener noreferrer">MDN Web Docs - Documentación Web</a>
          </li>
          <li>
            <a href="https://github.com/" target="_blank" rel="noopener noreferrer">GitHub - Proyectos Open Source</a>
          </li>
        </ul>
      </div>
    </div>
  </div>

```

Luego para finalizar esta página tenemos un último card, debajo de estos dos, que este es de 12 para que ocupe toda la columna entera.

En este he querido meter enlaces a páginas web donde puedas entrenar tu conocimiento sobre estos lenguajes y puedas ver información adicional que te pueda servir útil. Como los demás, se realiza mediante un href y metiendo la ruta de la página web.

Más sobre xml

Más sobre json

## Más Información

Explora estos recursos para practicar y obtener información educativa adicional:

- [LeetCode - Desafíos de Programación](#)
- [MDN Web Docs - Documentación Web](#)
- [GitHub - Proyectos Open Source](#)

Por último, le he metido un footer, que suele ser empleado en cualquier página web.

```

135 <footer class="footer">
136   <p>&copy; 2024 Mi Blog de Programación</p>
137 </footer>

```

## 2-XML.HTML

En este archivo tenemos el navbar igual y el sidebar igual. Lo único diferente es que en el div class="col-9" hemos metido únicamente una card donde tenemos información sobre este lenguaje.

```
75 <div class="col-lg-9">
76 <div class="content">
77 <div class="card mx-auto mt-5" style="max-width: 500px;">
78 <div class="card-body text-center">
79 <h2 class="card-title">XML</h2>
80 <p class="card-text">
81 XML proviene de un lenguaje inventado por IBM en los años 1970, llamado GML (Generalized Markup Language), que surgió por la necesidad que tenía la emp
82 <br>
83 Debe seguir estas reglas al crear la sintaxis XML:
84 <br>
85 <li>Los elementos XML deben tener una etiqueta de cierre.</li>
86 <br>
87 <li>Las etiquetas XML son sensibles a mayúsculas y minúsculas.</li>
88 <br>
89 <li>Todos los elementos XML deben estar debidamente anidados.</li>
90 <br>
91 <li>Todos los documentos XML debe tener un elemento raíz.</li>
92 <br>
93 <li>Los valores de atributo siempre deben estar entre comillas.</li>
94 </p>
95 </div>
96 <a href="https://www.w3.org/XML/" class="card-link">Más sobre XML en W3C</a>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
```

Con un enlace abajo para que me lleve a una página con mucha más información sobre XML.

## XML

XML proviene de un lenguaje inventado por IBM en los años 1970, llamado GML (Generalized Markup Language), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información y compartirla en otros SO y plataformas. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (Standard Generalized Markup Language), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información.

Debe seguir estas reglas al crear la sintaxis XML:

- Los elementos XML deben tener una etiqueta de cierre.
- Las etiquetas XML son sensibles a mayúsculas y minúsculas.
- Todos los elementos XML deben estar debidamente anidados.
- Todos los documentos XML debe tener un elemento raíz.
- Los valores de atributo siempre deben estar entre comillas.

[Más sobre XML en W3C](https://www.w3.org/XML/)



### 3-JSON.HTML

Esta parte es prácticamente igual, lo único que cambia es el contenido de la card y el enlace de abajo que lleva a otra página, en este caso de JSON.

```
74 <div class="col-lg-9">
75 <div class="content">|
76 <div class="card mx-auto mt-5" style="max-width: 500px;">
77 <div class="card-body text-center">
78 <h2 class="card-title">JSON</h2>
79 <p class="card-text">
80 JSON (acrónimo de JavaScript Object Notation, 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata
81 <br>
82 Reglas de sintaxis JSON que hay que conocer:
83 <br>
84 <li>Todos los datos del archivo deben estar rodeados de llaves {} si quieres representar un objeto y entre corchetes cuadrados si es un arreglo [].</li>
85 <br>
86 <li>Las comillas simples no están permitidas.</li>
87 <br>
88 <li>La llave en cada JSON debe ser única y debe estar entre comillas dobles.</li>
89 <br>
90 <li>Los números no deben ir entre comillas dobles, de lo contrario se tratarán como cadenas de texto.</li>
91 <br>
92 <li>El tipo de dato null no debe ir entre comillas dobles.</li>
93
94 </p>
95 <a href="https://www.json.org/" class="card-link">Más sobre JSON en json.org</a>
96
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
```

## JSON

JSON (acrónimo de JavaScript Object Notation, 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

Reglas de sintaxis JSON que hay que conocer:

- Todos los datos del archivo deben estar rodeados de llaves {} si quieres representar un objeto y entre corchetes cuadrados si es un arreglo [].
- Las comillas simples no están permitidas.
- La llave en cada JSON debe ser única y debe estar entre comillas dobles.
- Los números no deben ir entre comillas dobles, de lo contrario se tratarán como cadenas de texto.
- El tipo de dato null no debe ir entre comillas dobles.

[Más sobre JSON en json.org](https://www.json.org/)

## 4-CONTACTO.HTML

Esta última página es la que llevará un formulario de registro para el usuario que quiera estar registrado en mi página web.

```
34 <div class="container">
35 <div class="heading">Sign In</div>
36 <form action="" class="form">
37 <input required="" class="input" type="email" name="email" id="email" placeholder="E-mail">
38 <input required="" class="input" type="password" name="password" id="password" placeholder="Password">
39 <span class="forgot-password"><a href="#">Forgot Password ?</a></span>
40 <input class="login-button" type="submit" value="Sign In">
41
42 </form>
43 <div class="social-account-container">
44 <span class="title">Or Sign in with</span>
45 <div class="social-accounts">
46 <button class="social-button google">
47 <svg class="svg" xmlns="http://www.w3.org/2000/svg" height="1em" viewBox="0 0 488 512">
48 <path d="M488 261.8c488 403.3 391.1 504 504 504 0 393.2 0 2565110.8 8 248 8c66.8 0 123 24.5 166.3 64.91-67.5 64.9c258.5 52.6 94.3 116.6 94.3 256c0 86.5 69.
49 </svg></button>
50 <button class="social-button apple">
51 <svg class="svg" xmlns="http://www.w3.org/2000/svg" height="1em" viewBox="0 0 384 512">
52 <path d="M318.7 268.7c-.2-36.7 16.4-64.4 50-84.8 18.8-26.9 47.2-41.7-84.7-44.6-35.5-2.8 74.3 20.7-88.5 20.7-15 0-49.4-19.7-76.4-19.7c63.3 141.2 4 184.8 4 273.5q0 39.
53 </svg>
54 </button>
55 <button class="social-button twitter">
56 <svg class="svg" xmlns="http://www.w3.org/2000/svg" height="1em" viewBox="0 0 512 512">
57 <path d="M318.7 268.7c-.2-36.7 16.4-64.4 50-84.8 18.8-26.9 47.2-41.7-84.7-44.6-35.5-2.8 74.3 20.7-88.5 20.7-15 0-49.4-19.7-76.4-19.7c63.3 141.2 4 184.8 4 273.5q0 39.
58 </svg>
59 </button>
60 </div>
61 </div>
62 <span class="agreement"><a href="#">Learn user licence agreement</a></span>
63 </div>
```

Vamos a destacar lo mas importante de este código que pertenece al formulario.

Lo primero es el container donde ira el propio formulario, con el <div class "heading"> pondremos lo que es el título, básicamente lo que sale arriba de los campos del formulario.

Por otro lado, tenemos el form action para meter todos los campos que lleve dicho formulario, en este caso el e-mail y la contraseña.

Por último, el div class de abajo, sirve para hacer los logos de registrarse con cuentas de Google, por ejemplo.