

Hito Grupal

# Lenguaje de marcas

Eloy Pérez  
Natalia Ramírez  
Carla Nebreda  
Sergio Maeso

Enlace de la página en Google FireBase: <https://proyecto-1-eloy6lega.firebaseio.com>

# Índice

1.	FASE 1: Investigación teórica	Página 1
	- XML	Página 1
	- JSON	Página 4
2.	FASE 2: Proyecto web	Página 6
3.	División del trabajo	Página 16
4.	Webgrafía	Página 16

# FASE 1: Investigación teórica

## Lenguaje XML

### Historia

XML es el descendiente de un lenguaje de marcas que tuvo su origen en los años 70, llamado GML (General Markup Language), que se necesitaba para almacenar grandes cantidades de información por parte de IBM. De GML también procede otro lenguaje llamado SGML (Standard General Markup Language) , que se encargaba de estandarizar el GML.

Más tarde apareció un lenguaje a manos de un usuario de internet. Este era el HTML (Hypertext Markup Language), que se utilizaba para la web. HTML vio una rápida popularización. Una organización llamada W3C trató de establecer estándares y normas para HTML, pero este lenguaje de marcas todavía presentaba algunos inconvenientes.

Por ello, W3C comenzó el desarrollo de XML (Extensible Markup Language). En este lenguaje están trabajando expertos. Su objetivo es el de solucionar problemas de HTML al tratar la información. Algunos de estos problemas que HTML presentaba eran, por ejemplo, el contenido siendo mezclado con los estilos que se quieren aplicar; no permite compartir información con todos los dispositivos, como portátiles y móviles; la presentación en pantalla depende del visor que se utilice.

XML propuso soluciones como mezclar elementos de diferentes lenguajes (es decir, que sean extensibles), crear analizadores simples, remarcar que el documento no tenga errores de sintaxis, y dejar de lado características de SGML que permitían facilitar la escritura manual del documento y sustituirlas por metodologías que permitan a los programas automáticos interpretar lo que se incluye en el documento.

### Normas y sintaxis

Las normas de uso y sintaxis para que un documento XML se considere bien formado son:

- Todos los elementos XML deben tener una etiqueta de cierre. No se puede omitir dicha etiqueta.
- Las etiquetas de apertura y las de cierre son sensibles a minúsculas y también a mayúsculas. Las etiquetas <Etiqueta> y <etiqueta> son diferentes entre sí.
- La anidación de los elementos XML debe ser correcta.

- Debe haber un elemento raíz para todos los documentos XML. El elemento raíz es aquel que contiene todos los otros elementos, que pueden contener otros elementos hijos (los elementos hijo deben estar correctamente anidados).
- Los valores de los atributos siempre deben ir entre comillas ( " ).
- Los caracteres que se permiten usar para los elementos son las letras minúsculas, mayúsculas, números, puntos y guiones bajos y medios.
- También está permitido el uso de los dos puntos ( : ), pero su uso está reservado para la definición de los espacios de nombres.
- El primer carácter es obligatorio que sea una letra o un guión bajo.
- Las letras no inglesas se permiten pero no es recomendable usarlas, ya que podrían generar errores.

A continuación un ejemplo de un documento XML:

```
<?xml version="1.0" encoding="utf-8"?>
<items>
  <item id="0001" type="donut">
    <name>Cake</name>
    <ppu>0.55</ppu>
    <batters>
      <batter id="1001">Regular</batter>
      <batter id="1002">Chocolate</batter>
      <batter id="1003">Blueberry</batter>
      <batter id="1003">Devil's Food</batter>
    </batters>
    <topping id="5001">None</topping>
    <topping id="5002">Glazed</topping>
    <topping id="5005">Sugar</topping>
    <topping id="5007">Powdered Sugar</topping>
    <topping id="5006">Chocolate with Sprinkles</topping>
    <topping id="5003">Chocolate</topping>
    <topping id="5004">Maple</topping>
  </item>
  <item id="0002" type="donut">
    <name>Raised</name>
    <ppu>0.55</ppu>
    <batters>
      <batter id="1001">Regular</batter>
    </batters>
    <topping id="5001">None</topping>
    <topping id="5002">Glazed</topping>
    <topping id="5005">Sugar</topping>
    <topping id="5003">Chocolate</topping>
    <topping id="5004">Maple</topping>
  </item>
</items>
```

Otro ejemplo que muestra un archivo XML con una sintaxis bien formada. El nombre del archivo es colores.xml:

```
<mezclas>
  <mezclasDeColores1>
    <mezcla>rojo y amarillo</mezcla>
    <resultado mezcla1='Naranja'></resultado>
  </mezclasDeColores1>
  <mezclasDeColores2>
    <mezcla>azul y amarillo</mezcla>
    <resultado mezcla1='Verde'></resultado>
  </mezclasDeColores2>
  <mezclasDeColores3>
    <mezcla>blanco y negro</mezcla>
    <resultado mezcla1='Gris'></resultado>
  </mezclasDeColores3>
</mezclas>
```

### Aplicaciones del XML en el ámbito empresarial

El XML (Extensible Markup Language) es un lenguaje de marcado ampliamente utilizado en el ámbito empresarial para intercambiar información electrónica entre aplicaciones y sistemas. Algunos de los usos más comunes de XML en el ámbito empresarial incluyen:

1. Intercambio de datos: XML se utiliza para intercambiar datos entre diferentes aplicaciones y sistemas, permitiendo la integración de sistemas y la automatización de procesos.
2. Documentación electrónica: XML se utiliza para almacenar y transmitir documentos electrónicos, como facturas, órdenes de compra, etc.
3. Comercio electrónico: XML se utiliza para intercambiar información entre tiendas en línea y sistemas de pago, así como para la integración de diferentes aplicaciones de comercio electrónico.
4. Búsqueda y recuperación de información: XML permite la indexación y la recuperación de información en grandes volúmenes de datos, lo que resulta útil en el contexto de la gestión de documentos y la recuperación de información.
5. Descripción de productos: XML se utiliza para describir productos y servicios, lo que facilita la comparación y la selección de proveedores y productos.

En resumen, el XML es una tecnología muy versátil y ampliamente utilizada en el ámbito empresarial, que permite la integración de sistemas y la automatización de procesos, lo que a su vez mejora la eficiencia y reduce los costos.

## Lenguaje JSON

### Historia

Historia de JSON: JSON, acrónimo de JavaScript Object Notation, es un formato de datos de transferencia ligero, compacto, de estándar abierto e independiente que se utiliza para que los sistemas puedan intercambiar datos entre sí.

Cuando JSON surgió había la necesidad de crear un protocolo entre el servidor y el navegador en tiempo real, sin estado y sin el uso de plug-ins o el software que pueda ofrecer un tercero.

Douglas Crockford es el primer promotor de JSON como formato, y creó el acrónimo desde su empresa, State Software. Poco a poco JSON ha ido reemplazando XML como formato preferido a la hora de transferir datos en la web, aunque en la actualidad ambos se siguen utilizando.

La popularidad de JSON llevó a su estandarización por varios organismos internacionales, como ISO/IEC 21778:2017 y RFC 8259.

### Normas y sintaxis

Las normas de uso y de sintaxis de los documentos JSON consiste en:

- Todos los datos del archivo deben estar contenidos en llaves ( `{}` ) si la representación es de un objeto y en corchetes si este es un arreglo.
- No se permiten comillas simples ( ``` ).
- La llave en cada JSON debe ser única y debe situarse entre comillas dobles ( `"` ).
- Los números no pueden estar entre comillas dobles, para evitar que se les considere cadenas de caracteres.
- Los datos que sean NULL no se pueden colocar entre comillas dobles tampoco, por el mismo motivo.
- Los valores booleanos solo admiten verdadero y falso.
- Cada par llave:valor debe terminar en coma, excepto el último elemento.
- Un solo objeto que se encuentre en un arreglo también debe terminar con una coma.

Un ejemplo de archivo JSON bien formado es el siguiente. Tenemos un archivo llamado usuarios.json:

```
[
  {
    "codigo" : 1,
    "usuario" : "manolo",
    "passwords" : "manolito26"
  },
  {
    "codigo" : 2,
    "usuario" : "pepe",
    "passwords" : "pepito82"
  },
  {
    "codigo" : 3,
    "usuario" : "paco",
    "passwords" : "paquito105"
  }
]
```

### **Aplicaciones del JSON en el ámbito empresarial**

JSON (JavaScript Object Notation) es un formato de intercambio de datos que se ha vuelto muy popular en el ámbito empresarial debido a su facilidad de uso y leibilidad. Algunas aplicaciones comunes de JSON en el ámbito empresarial:

1. Intercambio de datos entre aplicaciones: JSON es un formato estándar para el intercambio de datos entre aplicaciones, lo que facilita la integración entre sistemas.
2. API web: Muchas API utilizan JSON para recibir y enviar datos, lo que permite a las aplicaciones recibir y enviar información en tiempo real.
3. Almacenamiento de datos: JSON es un formato de almacenamiento de datos popular que permite a las aplicaciones almacenar y recuperar información de manera sencilla.
4. Análisis de datos: JSON es compatible con muchos lenguajes de programación y herramientas de análisis de datos, lo que lo hace ideal para el procesamiento de grandes cantidades de datos.

5. Presentación de datos: JSON es un formato de datos fácil de leer y escribir, lo que lo hace ideal para la presentación de datos en aplicaciones web y móviles.

En resumen, JSON es un formato de intercambio de datos flexible y eficiente que se ha vuelto muy popular en el ámbito empresarial y se utiliza en una variedad de aplicaciones, desde la integración de sistemas hasta el análisis de datos y la presentación de información.

## FASE 2: Proyecto web

### Justifica el proyecto y la metodología de desarrollo

Nuestro cliente nos solicita un proyecto web desarrollado a partir de tecnologías web con contenidos académicos para que sus trabajadores puedan aprender a almacenar y gestionar datos usando XML y JSON.

La página se encuentra subida en un repositorio de GitHub, y puede visualizarse en el siguiente enlace: <https://natalia-ramirez-benito.github.io/TechHub>

Para el desarrollo de este proyecto hemos creado una página index a modo de página de inicio, hemos añadido un header personalizado para poder acceder a los contenidos de JSON, XML, Contacto dónde hemos creado un pequeño formulario y el logo te redirige a la página de inicio.

Antes de entrar al body, en el head hemos utilizado el quick start de Bootstrap para nuestro CSS y en el body para nuestro paquete de JavaScript (incluido Popper para colocar menús desplegables, ventanas emergentes e información sobre herramientas) antes del cierre `</body>`. Además hemos añadido otro link para poder utilizar las fuentes proporcionadas por Google.

Estos serían los elementos insertados en el `<head>`:

```
index.html > html > body#custom > div.header-custom > div.container > header.d-flex.flex-wrap.align-items-center.justify-content-center.justify-content-mo
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">
6   <meta name="keywords" content="json, xml, tecnologia, html">
7   <meta name="autor" content="Natalia Ramírez, Carla Nebreda, Eloy Pérez, Sergio Maeso">
8   <title>Tech Hub</title>
9   <link rel="shortcut icon" href="img/logo.png">
10
11   <!-- Custom styles for this template -->
12   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="s
13   <link href="https://fonts.googleapis.com/css?family=Playfair&#43;Display:700,900&amp;display=swap" rel="stylesheet">
14   <link rel="stylesheet" type="text/css" href="estilos/custom.css">
15 </head>
16
```



Y los elementos mencionados anteriormente en el <body>:

```
<script type="module" src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384">
</body>
</html>
```



XML JSON Contacto

Además hemos añadido el logo diseñado como icono de la pestaña:



En la parte central hemos creado una flexbox dónde colocaremos una imagen con una pequeña descripción de la web, una barra de navegación lateral con los elementos:

- Categorías
  - Tecnología
  - Lenguajes de programación
  - Frameworks
- Últimos posts
  - XML
  - JSON
  - HTML
  - XSL
- Síguenos en redes
  - Twitter
  - Twitch
  - Instagram
  - Youtube

#### Navegación

- ▼ Categorías
  - Tecnología
  - Lenguajes de programación
  - Frameworks
- > Últimos posts

Síguenos en redes



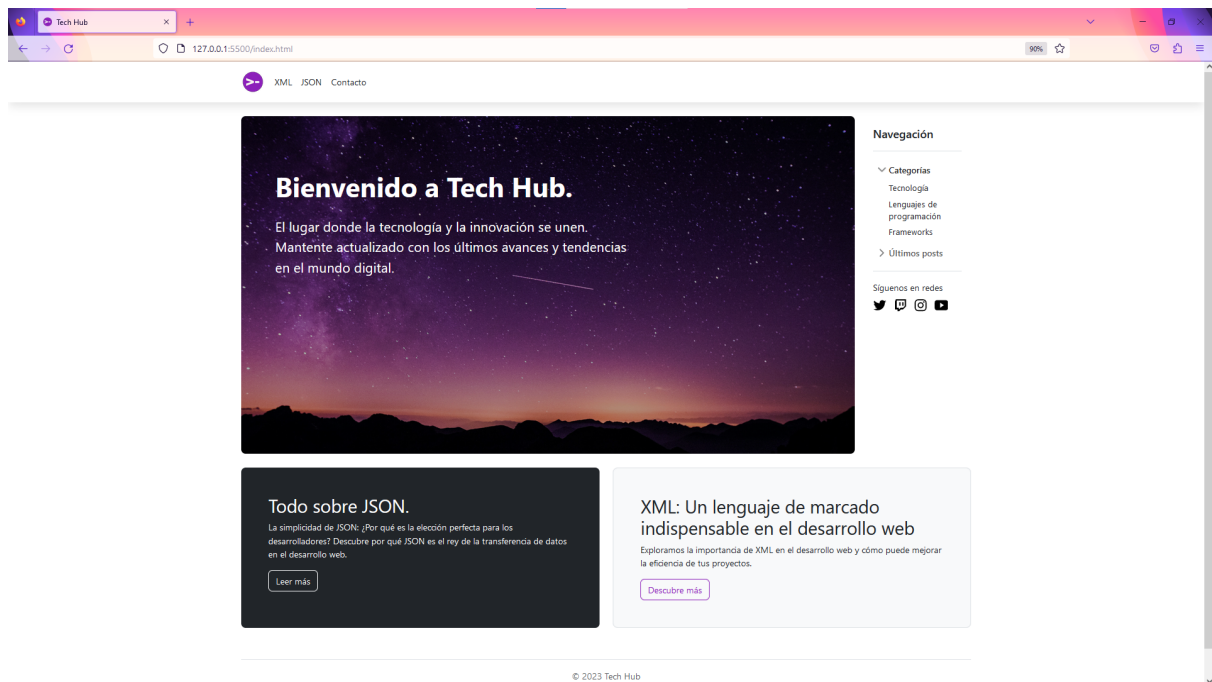
La parte de Categorías la mostraremos abierta, Últimos Post cerrado y en la parte de Síguenos en redes utilizaremos los iconos de las correspondientes redes sociales. Además añadiremos que al pinchar en XML y JSON nos lleve correspondientemente a las páginas sobre XML y JSON.

Para los iconos hemos tenido que implementar un módulo de la página web [ionicons](#) para poder utilizar logos más atractivos que los que ofrece Bootstrap.

Debajo de estos dos elementos crearemos dos tarjetas con un título sobre JSON y el otro sobre XML con un pequeño resumen del contenido y un botón que redirige a cada página para seguir leyendo sobre los temas expuestos.

Por último creamos un footer con el nombre centrado de la empresa.

El producto final se vería así:




Para la página de XML hemos utilizado la misma estructura que en index.html pero en vez de las tarjetas hemos organizado la información de la primera fase sobre XML y lo mismo con la página sobre JSON. Además a ambos les hemos añadido una cabecera antes de la explicación sobre cada uno.

Así se vería cada una de las páginas nombradas anteriormente:

- JSON:



- XML:


[XML](#)
[JSON](#)
[Contacto](#)

# XML

Un lenguaje de marcado imprescindible para mejorar la organización y presentación de datos.

## Historia

XML es el descendiente de un lenguaje de marcas que tuvo su origen en los años 70, llamado GML (General Markup Language), que se necesitaba para almacenar grandes cantidades de información por parte de IBM. De GML también procede otro lenguaje llamado SGML (Standard General Markup Language), que se encargaba de estandarizar el GML.

Más tarde apareció un lenguaje a manos de un usuario de internet. Este era el HTML (Hypertext Markup Language), que se utilizaba para la web. HTML vio una rápida popularización. Una organización llamada W3C trató de establecer estándares y normas para HTML, pero este lenguaje de marcas todavía presentaba algunos inconvenientes.

Por ello, W3C comenzó el desarrollo de XML (Extensible Markup Language). En este lenguaje están trabajando expertos. Su objetivo es el de solucionar problemas de HTML al tratar la información. Algunos de estos problemas que HTML presentaba eran, por ejemplo, el contenido siendo mezclado con los estilos que se quieren aplicar; no permite compartir información con todos los dispositivos, como portátiles y móviles; la presentación en pantalla depende del visor que se utilice.

XML propuso soluciones como mezclar elementos de diferentes lenguajes (es decir, que sean extensibles), crear analizadores simples, remarcar que el documento no tenga errores de sintaxis, y dejar de lado características de SGML que permitían facilitar la escritura manual del documento y sustituirlas por metodologías que permitan a los programas automáticos interpretar lo que se incluye en el documento.

## Normas y sintaxis





Las normas de uso y sintaxis para que un documento XML se considere bien formado son:

- Todos los elementos XML deben tener una etiqueta de cierre. No se puede omitir dicha etiqueta.
- Las etiquetas de apertura y las de cierre son sensibles a minúsculas y también a mayúsculas. Las etiquetas y son diferentes entre sí.
- La anidación de los elementos XML debe ser correcta.


### Navegación

- ✓ Contenido
  - Historia
  - Normas y sintaxis
  - Ámbito empresarial
- > Categorías
- > Últimos posts

Síguenos en redes

En la página de contacto hemos creado un pequeño formulario en el que le pedimos al usuario que nos de su email, su nombre y que nos cuente sus dudas o sugerencias para el blog. Además hemos añadido una pequeña cabecera introduciendo el apartado con una imagen de fondo.


[XML](#)
[JSON](#)
[Contacto](#)

# Contacta con nosotros

¿Tienes dudas?

Introduce tu email

Introduce tu nombre

¿Tienes dudas?

Escribe tus dudas o sugerencias para el blog.

Enviar

© 2023 Tech Hub

## Tecnologías utilizadas

Hemos creado un CSS único para todas las páginas llamado custom.css en dónde hemos manejado el responsive, algunas implementaciones más complejas de las flexbox, los botones con su hover y transición, el header, el hero-img y algunos elementos específicos como el tamaño máximo del ancho del elemento con clase main.

Hemos utilizado el framework de Bootstrap para implementar un sistema de rejilla flexible que, junto con el uso de media query, ha facilitado que el proyecto sea responsive adaptándose a las diferentes resoluciones de pantalla (mobile, tablet, laptop, desktop) en las que el proyecto será visualizado.

El flexbox administra rápidamente el diseño, la alineación y el tamaño de las columnas de cuadrícula, la navegación, los componentes y más con un conjunto completo de utilidades flexbox receptivas. Para el espacio entre elementos (justify-content) y el como se muestran los elementos (display). Para el justify-content le hemos dado el valor de "space-between" que sirve para distribuir elementos uniformemente estando el primer elemento al ras con el inicio y el último está al ras con el final.

```
5  /* =====
6  GENERAL
7  =====
8
9  .flexbox {
10     display: block;
11     justify-content: space-between;
12 }
```

La regla .main establece un max-width de 900px.

```
20  .main {
21     max-width: 900px;
22  }
23
```

La regla .container-narrow establece un max-width de 600px.

```
43  .container-narrow {
44     max-width: 600px;
45  }
```

La regla #custom .logo establece un width automático, una altura de 41 píxeles y un margin-right de 10 píxeles.

```
47 #custom .logo {  
48     width: auto;  
49     height: 41px;  
50     margin-right: 10px;  
51 }
```

La regla .btn-purple establece una transición de 0.4 segundos, un margin-top de 24 píxeles, un background-color y un border-color de púrpura y un ancho del 100%. Cuando se coloca el cursor encima de los botones de clase .btn-purple, se cambia el background-color y el border-color a púrpura y el color a blanco.

La regla .btn-purple-inline establece un color y un border-color de púrpura. Cuando se coloca el cursor encima de los botones de clase .btn-purple-inline, el background-color y el border-color cambian a púrpura.

La regla .btn-purple:hover y .btn-purple-inline:hover se usa para seleccionar elementos cuando pasa el mouse sobre ellos con un background-color y un border-color púrpura.

```
63 .btn-purple {  
64     transition: 0.4s;  
65     margin-top: 24px;  
66     background-color: #8C20B9;  
67     border-color: #8C20B9;  
68     width: 100%;  
69 }  
70  
71 .btn-purple:hover {  
72     background-color: white;  
73     color: #8C20B9;  
74     border-color: #8C20B9;  
75 }  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

La regla .img-sample establece un margen de 36 píxeles en la parte superior y inferior y un display de bloque con un margin automático en los lados.

```
87 .img-sample {  
88     margin: 36px auto;  
89     display: block;  
90 }
```

Las reglas debajo de `/* HEADER */` establecen un box-shadow en el encabezado y cambian el color de los enlaces de navegación cuando se coloca el cursor sobre ellos.

```
92  /* =====
93  HEADER
94  ===== */
95
96  #custom .header-custom {
97      box-shadow: 0px 10px 15px -3px rgba(0,0,0,0.1);
98      transition: 0.3s;
99  }
100
101  #custom .nav-link:hover {
102      color: #8C20B9!important;
103  }
104
105  @media (min-width: 767px) {
106      #custom .btns-custom {
107          width: 30%;
108      }
109  }
110
```

Las reglas debajo de `/* HERO */` establecen el tamaño de la imagen del héroe, su posición y otras características de estilo. También establecen el tamaño y el estilo de las imágenes del blog.

```
115  .hero-img {
116      background-repeat: no-repeat;
117      background-position: center center;
118      background-size: cover;
119      position: relative;
120      overflow: hidden;
121      width: 90%;
122      height: 600px;
123      margin-right: 16px;
124  }
```

```
134  .hero-transparent {
135      background: rgba(00, 00, 00, 0.3);
136      position: absolute;
137      top: 0;
138      bottom: 0;
139      right: 0;
140      z-index: 0;
141      width: 100%;
142      height: 100%;
143  }
144
145  .hero-index {
146      position: relative;
147      z-index: 10;
148      color: white;
149  }
```

```

151 .hero-blog {
152     background-repeat: no-repeat;
153     background-position: center center;
154     background-size: cover;
155     position: relative;
156     overflow: hidden;
157     width: 100%;
158     min-height: 233px;
159 }
160
161 @media (max-width: 650px) {
162     .hero-blog {
163         padding: 32px !important;
164     }
165 }

```

```

166
167 .hero-blog p {
168     margin-bottom: 0;
169 }
170
171 .copy {
172     margin-top: 24px;
173 }
174

```

Las reglas debajo de /\* SIDE BAR \*/ establecen el tamaño y el estilo de la barra lateral y los iconos sociales, y también establecen el estilo de los botones de alternancia.

```

179 .nav-bar {
180     width: 190px;
181 }
182
183 @media (max-width: 650px) {
184     .nav-bar {
185         width: 100%;
186         padding: 0 !important;
187     }
188 }
189
190 .social-icon {
191     display: inline-block;
192     font-size: 24px;
193     padding-right: 8px;
194     transition: 0.3s;
195     color: black;
196 }
197
198 .social-icon:hover {
199     color: #8C20B9;
200 }

```

```

202 .btn-toggle {
203     padding: .25rem .5rem;
204     font-weight: 600;
205     color: rgba(0, 0, 0, .65);
206     background-color: transparent;
207     text-align: left;
208 }
209 .btn-toggle:hover,
210 .btn-toggle:focus {
211     color: rgba(0, 0, 0, .85);
212     background-color: #e1b4f7;
213 }
214
215 .btn-toggle::before {
216     width: 1.25em;
217     line-height: 0;
218     content: url("data:image/svg+xml,%3csvg xml
219     transition: transform .35s ease;
220     transform-origin: .5em 50%;
221 }

```

```

215 .btn-toggle::before {
216     width: 1.25em;
217     line-height: 0;
218     content: url("data:image/svg+xml,%3csvg xmlns='http://w
219     transition: transform .35s ease;
220     transform-origin: .5em 50%;
221 }
222
223 .btn-toggle[aria-expanded="true"] {
224     color: rgba(0, 0, 0, .85);
225 }
226 .btn-toggle[aria-expanded="true"]::before {
227     transform: rotate(90deg);
228 }
229
230 .btn-toggle-nav a {
231     padding: .1875rem .5rem;
232     margin-top: .125rem;
233     margin-left: 1.25rem;
234 }
235 .btn-toggle-nav a:hover,
236 .btn-toggle-nav a:focus {
237     background-color: #d2f4ea;
238 }
239
240 .scrollarea {
241     overflow-y: auto;
242 }

```

Para la parte del responsive hemos utilizado el responsive de bootstrap en el código html creando containers y container-custom además del row y el col correspondiente y, además, hemos ajustado en el css los elementos a las distintas resoluciones:

```

31 <div class="container container-custom">
32   <div class="flexbox">
33     <div class="p-5 mb-4 bg-light rounded-3 hero-img" style="background-image: url('./img/hero-img.jpg');">
34       <div class="container-fluid py-5 hero-index">
35         <h1 class="display-5 fw-bold">Bienvenido a Tech Hub.</h1>
36         <p class="col-md-8 fs-4 copy">El lugar donde la tecnología y la innovación se unen. <br>Mantente actualizado con los últimos avances y tendencias en el mundo digital.
37       </div><!--end of container-fluid-->
38     <div class="hero-transparent"></div>
39   </div><!--end of p-5 mb-4-->
40
41
42   <div class="flex-shrink-0 p-3 bg-white nav-bar">
43     <div href="#" class="d-flex align-items-center pb-3 mb-3 link-dark text-decoration-none border-bottom">
44       <span class="fs-5 fw-semibold">Navegación</span>
45     </div>

```



```
14  @media (min-width: 651px) {
15    .flexbox {
16      display: flex;
17    }
18  }
19
20  .main {
21    max-width: 900px;
22  }
23
24  @media (max-width: 992px) {
25    .container-custom {
26      max-width: none;
27      margin: 0 auto;
28    }
29  }
```

```
30
31  @media (min-width: 768px) {
32    .container-custom {
33      margin: 0 auto;
34    }
35  }
36
37  @media (max-width: 768px) {
38    .container-custom {
39      max-width: none;
40    }
41  }
```

# División del trabajo

**Eloy:** Búsqueda de parte de la información y utilización de la misma en el sitio web. Ligeras ayudas a Carla con el HTML y CSS. Creación del logo con Natalia

**Carla:** desarrollo y maquetación del sitio web en HTML y CSS de manera responsive, usando Media Querys y el grid system de Bootstrap.

**Sergio:** Documentación de la fase 1.

**Natalia:** documentación fase 2, creación logo con Eloy y subir el proyecto a repositorio Github.

## Webgrafía

Historia de XML

<https://desarrolloweb.com/articulos/450.php>

Introducción a XML

[https://lm-xml-apuntes.readthedocs.io/apuntes/10\\_introduccion\\_xml.html](https://lm-xml-apuntes.readthedocs.io/apuntes/10_introduccion_xml.html)

¿Qué es JSON y para qué sirve?

<https://marquesfernandes.com/es/tecnologia-es/what-and-json-and-for-that-serves/>

[https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=956:ique-es-y-para-que-sirve-json-especificacion-oficial-javascript-object-notation-diferencia-de-xml-cu01213f&catid=83&Itemid=212](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=956:ique-es-y-para-que-sirve-json-especificacion-oficial-javascript-object-notation-diferencia-de-xml-cu01213f&catid=83&Itemid=212)

Historia completa de JSON

<https://history-computer.com/json-complete-history/>

Normas de uso y sintaxis de XML

<https://www.ibm.com/docs/es/scbn?topic=syntax-xml-rules>

Normas de sintaxis de XML

<https://www.abrirllave.com/xml/normas-de-sintaxis-basicas.php>

Qué es un archivo JSON, normas de uso y sintaxis

<https://www.freecodecamp.org/espanol/news/que-es-un-archivo-json-ejemplo-de-codigo-en-javascript/>

