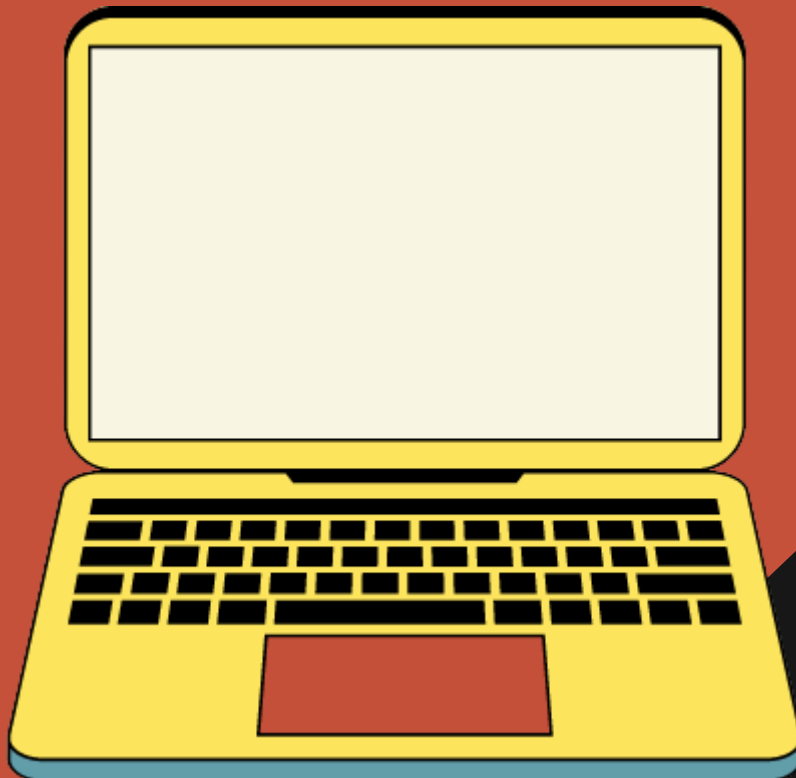


# Práctica 2

## Grupo 9

{  
Lenguajes  
de Marcas  
}



## Contenido

INTRODUCCIÓN .....	3
EDITORES DE TEXTO .....	4
CREACIÓN .....	7
VALIDACIÓN .....	15
PUBLICACIÓN .....	20
DOCUMENTO XML .....	22
CONCLUSIÓN .....	23

## INTRODUCCIÓN

En esta práctica, se abordarán dos aspectos esenciales del mundo digital: la creación y gestión de contenido web utilizando HTML 5, así como la representación estructurada de datos a través de documentos XML.

En primer lugar, se destaca la importancia de elegir un editor de texto adecuado, como Notepad++, que nos permitirá tener un control más completo sobre el código que crean.

La práctica se enfoca en la creación de páginas web utilizando HTML 5 como lenguaje de marcado donde incorporaremos formularios en el diseño. Y una posterior validación de estas páginas es un paso crucial para garantizar que cumplan con las especificaciones de HTML 5.

En paralelo, se aborda la creación de un documento XML. Donde deberemos estructurar la información de manera que sea legible tanto para humanos como para máquinas, utilizando etiquetas XML.

En resumen, esta práctica combina la creación de contenido web y la representación de datos, preparando a los estudiantes para enfrentar desafíos del mundo real en la creación y gestión de contenido digital. La elección de un editor adecuado, la creación de páginas web, la validación y la representación estructurada de datos en XML son habilidades fundamentales en un mundo cada vez más digital.

## EDITORES DE TEXTO

Antes de ponernos a trabajar en nuestra página web, necesitamos un editor de texto donde poder editar y visualizar nuestro código. La elección ha sido Notepad++, pues su sencillez permitirá tener mucho más control sobre lo que vayamos programando para nuestra página web.

Para comenzar con la instalación tenemos que ir a la página web para obtener el software gratuito <https://notepad-plus-plus.org>. Una vez ahí, en la parte derecha, hacemos clic en Download y seleccionamos la versión que más se nos ajuste. Recomendando no usar la última versión pues todavía se sigue actualizando y puede tener “bugs”. Por lo que, descargamos la versión 8.5.6.



Current Version 8.5.7

Home

Download

News

Online Help

Resources

RSS

Donate

Author

### Downloads

Download Notepad++ v8.5.7 (Vulnerability fixes)

Download Notepad++ v8.5.6

Download Notepad++ v8.5.5

Download Notepad++ v8.5.4

Download Notepad++ v8.5.3

Y faltaría descargar el ejecutable. (recomendado, hará que la instalación sea más sencilla). Si vuestro SO es de arquitectura 32 bits, asegurarse de seleccionar bien su versión.

Download 64-bit x64

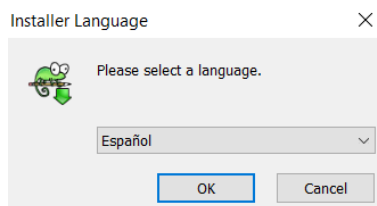


- [Installer](#) | [GPG Signature](#)
- [Portable \(zip\)](#) | [GPG Signature](#)
- [Portable \(7z\)](#) | [GPG Signature](#)
- [Mini-portable \(7z\)](#) | [GPG Signature](#)

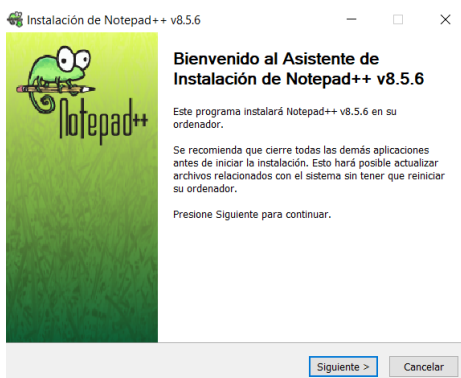
A continuación, comenzamos con la instalación del programa.

Basta con hacer doble clic en el ejecutable y comenzaremos con la misma:

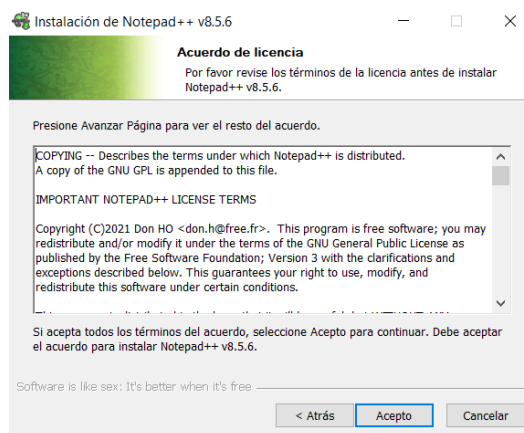
1. Primero seleccionamos el idioma de instalación.



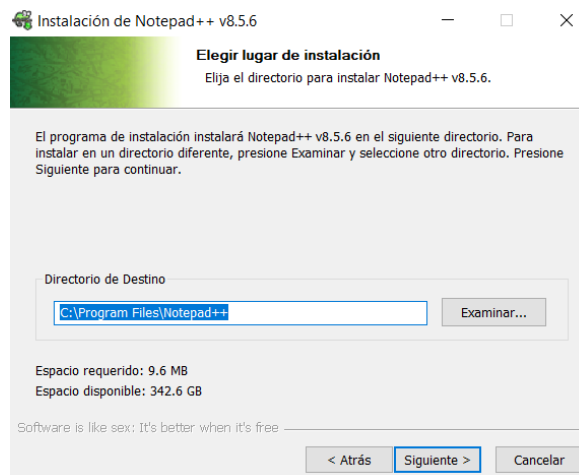
2. Hacemos clic en Siguiente.



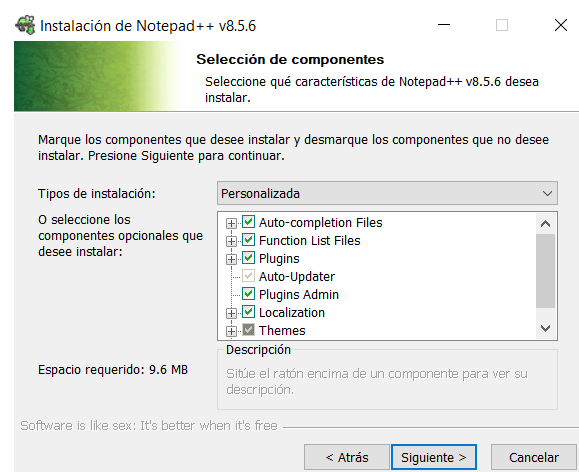
3. Aceptamos los términos de la licencia.



4. Seleccionamos la ruta donde se instalará el editor, nosotros lo dejaremos en la que muestra por defecto.



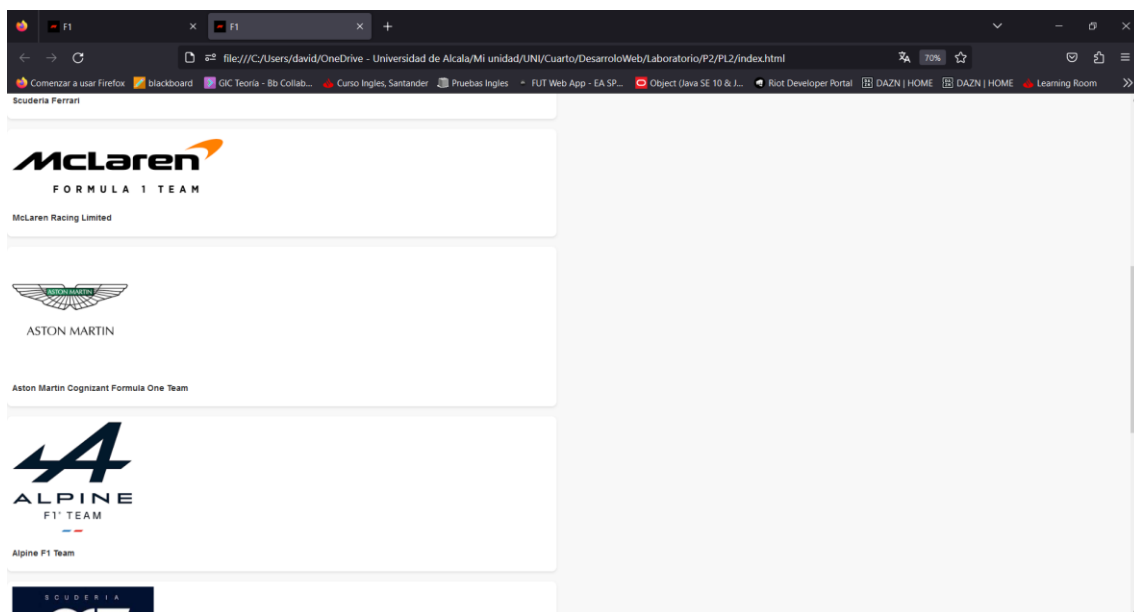
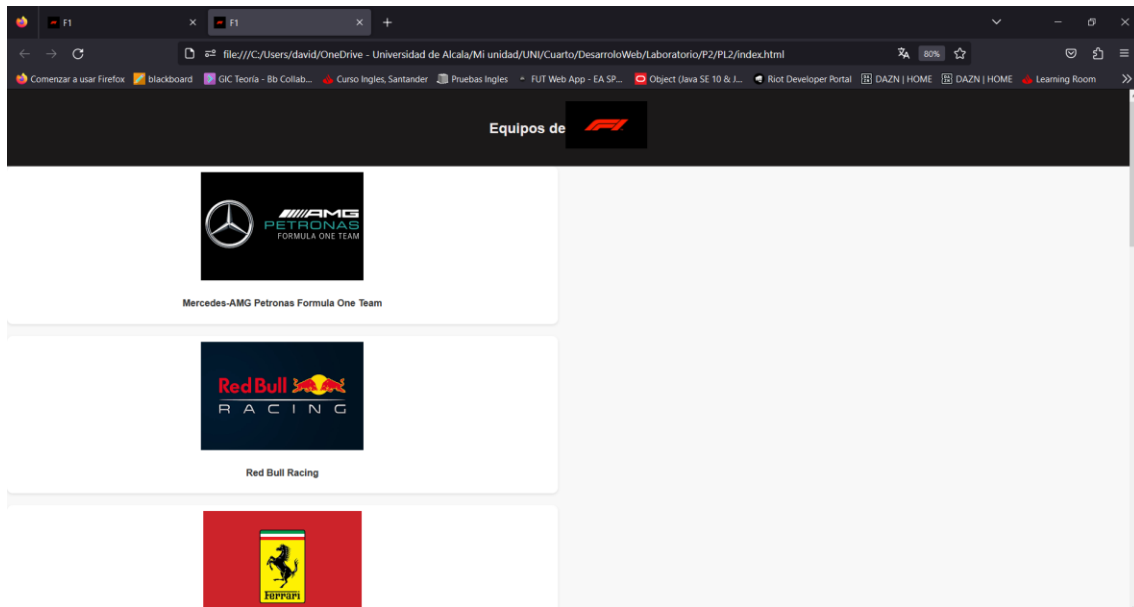
## 5. Seleccionamos todos los componentes adicionales.



Una vez terminada la instalación, ejecutamos la aplicación y podremos proseguir con la realización de esta práctica.

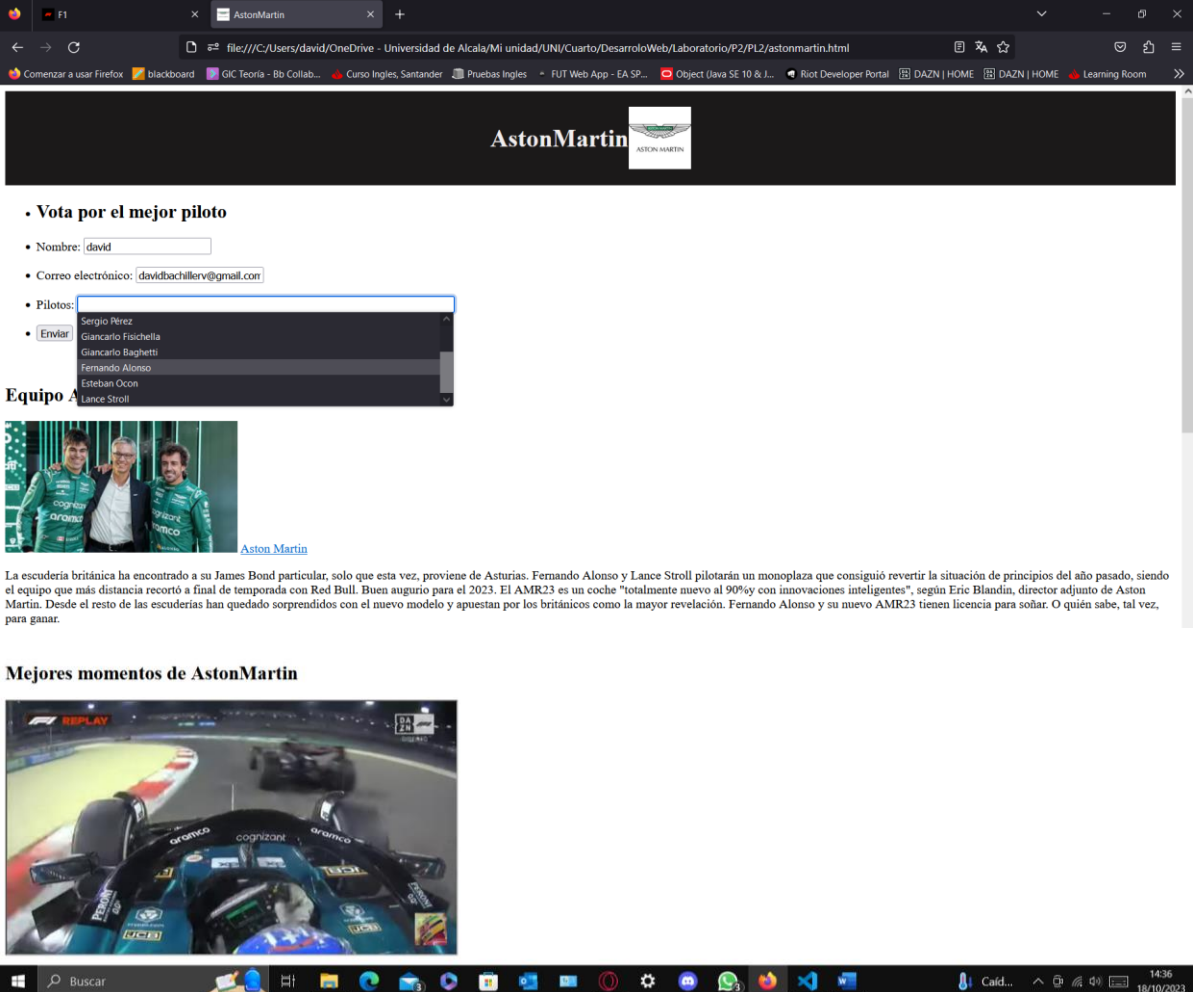
# CREACIÓN

Para la creación de nuestra página web hemos elegido temática la F1, donde tendremos una página principal con todas las escuderías que actualmente ahí en la F1.



Donde sí vamos bajando vamos a poder encontrar los diferentes equipos y vamos a poder seleccionar el equipo que más nos guste o del que queramos conocer más información.

La estructura de cada equipo es similar, tenemos una cabecera con el nombre del equipo luego un formulario para seleccionar los mejores pilotos de la historia de ese equipo, también información adicional del equipo y un video con los mejores momentos del equipo.



**AstonMartin**

• Vota por el mejor piloto


• Nombre:

• Correo electrónico:

• Pilotos:


•

**Equipo Aston Martin**

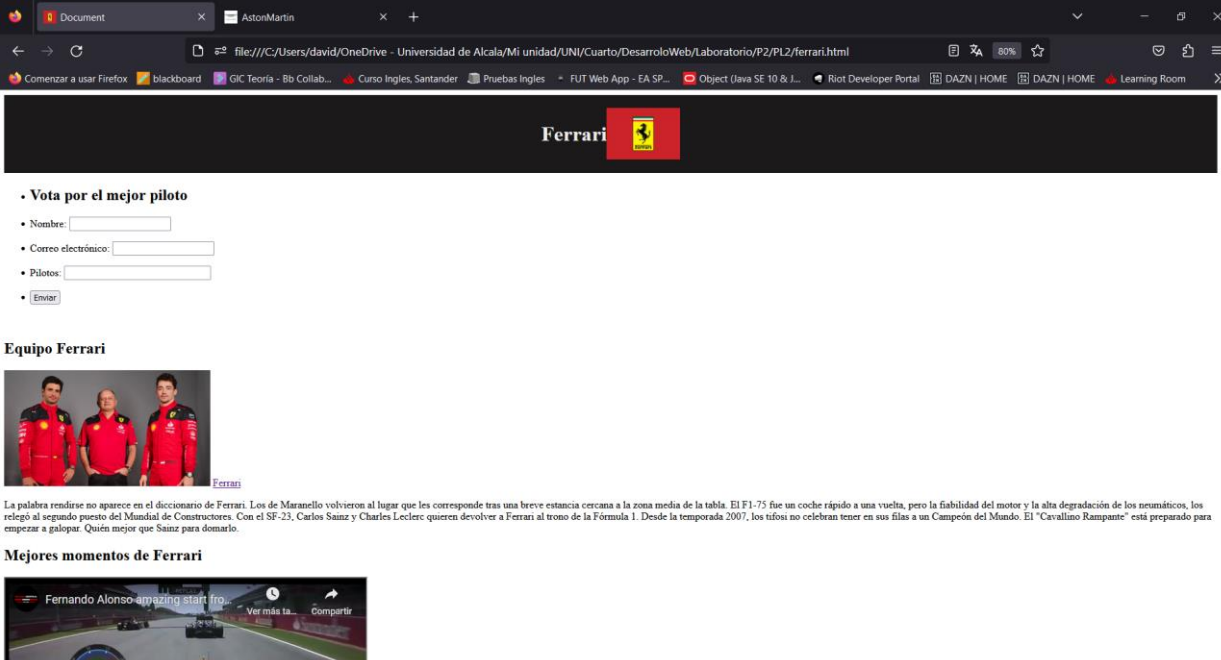


La escudería británica ha encontrado a su James Bond particular, solo que esta vez, proviene de Asturias. Fernando Alonso y Lance Stroll pilotarán un monoplaza que consiguió revertir la situación de principios del año pasado, siendo el equipo que más distancia recorrió a final de temporada con Red Bull. Buen augurio para el 2023. El AMR23 es un coche "totalmente nuevo al 90% con innovaciones inteligentes", según Eric Blandin, director adjunto de Aston Martin. Desde el resto de las escuderías han quedado sorprendidos con el nuevo modelo y apuestan por los británicos como la mayor revelación. Fernando Alonso y su nuevo AMR23 tienen licencia para soñar. O quién sabe, tal vez, para ganar.

**Mejores momentos de AstonMartin**



La estructura de los 9 equipos restantes será similar solo cambiarán los pilotos, los team manager, la información, un video de mejores momentos de cada escudería, así como el logo y el nombre. Veamos el ejemplo de otra escudería:



**Ferrari**

• Vota por el mejor piloto


• Nombre:

• Correo electrónico:

• Pilotos:


•

**Equipo Ferrari**



La palabra renderise no aparece en el diccionario de Ferrari. Los de Maranello volvieron al lugar que les corresponde tras una breve estancia cercana a la zona media de la tabla. El F1-75 fue un coche rápido a una vuelta, pero la fiabilidad del motor y la alta degradación de los neumáticos, los relegó al segundo puesto del Mundial de Constructores. Con el SF-23, Carlos Sainz y Charles Leclerc quieren devolver a Ferrari al trono de la Fórmula 1. Desde la temporada 2007, los tifosi no celebran tener en sus filas a un Campeón del Mundo. El "Cavallino Rampante" está preparado para empezar a galopar. Quéda mejor que Sainz para domarlo.

**Mejores momentos de Ferrari**





Vista como es la parte “bonita” de la web. Vamos a adentrarnos en el código que hay detrás de esta.

Algo que tiene que estar presente en toda página web es la siguiente estructura:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" href="Imagenes/fl.jpg">
  <title>F1</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f8f8f8;
    }

    section {
      background-color: #1b1919;
      color: #f2eeee;
      text-align: center;
      padding: 20px 0;
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .team {
      width: 48%;
      margin-bottom: 20px;
      background-color: #fff;
      padding: 10px;
      border-radius: 10px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
      text-align: center;
    }

    ul {
      list-style: none;
      padding: 0;
    }

    li {
      margin: 10px 0;
    }

    a {
      text-decoration: none;
      color: #333;
      font-weight: bold;
    }

    a:hover {
      color: #ff6600;
    }

    img {
      max-width: 100%;
      height: auto;
      margin-bottom: 10px;
    }
  </style>
</head>
```

- <!DOCTYPE html>: Esta es una declaración doctype que se utiliza para especificar la versión de HTML que se está utilizando en la página. En este caso, se refiere a HTML5, la última versión de HTML en el momento de su publicación. HTML5 es ampliamente compatible y recomendado para la creación de páginas web modernas.

- `<html>`: El elemento `<html>` es el elemento raíz de una página HTML. En él, se definen los elementos y estructura de toda la página. El atributo `lang` se utiliza para especificar el idioma principal de la página. En este caso, `lang="en"` indica que el idioma principal de la página es el inglés.
- Encabezado HTML (`<head>`):  
En esta sección, se especifican las meta etiquetas que proporcionan información sobre la página, como el juego de caracteres utilizado y la escala de la vista en dispositivos móviles. También se define el ícono de la pestaña del navegador y el título de la página.
- Estilos CSS (`<style>`):  
Aquí se definen los estilos de la página utilizando CSS (Cascading Style Sheets). Esto incluye la apariencia de los elementos, como colores de fondo, fuentes y sombras. Los estilos se aplican a toda la página. Hemos usado esto para que la página web sea más amigable y atractiva a la vista.

Lo que ya es particular del índice viene en el `body`:

```
<body>
  <section>
    <h1>Equipos de</h1>
    
  </section>

  <div class="team">
    
    <ul>
      <li><a href="mercedes.html">Mercedes-AMG Petronas Formula One Team</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="redbull.html">Red Bull Racing</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="ferrari.html">Scuderia Ferrari</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="mclaren.html">McLaren Racing Limited</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="astonmartin.html">Aston Martin Cognizant Formula One Team</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="alpine.html">Alpine F1 Team</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="alphatauri.html">Scuderia AlphaTauri Honda</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="haas.html">Haas F1 Team</a></li>
    </ul>
  </div>
  <div class="team">
    
    <ul>
      <li><a href="sauber.html">Alfa Romeo Racing Orlen</a></li>
    </ul>
  </div>
  <div class="team">

```

- Cuerpo HTML (<body>):  
El cuerpo de la página contiene el contenido visible para los visitantes. Está dividido en dos secciones:
  - Sección de Encabezado (<section>):

En esta sección, se coloca un título "Equipos de F1" junto con una imagen del logo de Fórmula 1. Esta área se utiliza comúnmente para presentar el tema principal de la página.

- Equipos de Fórmula 1 (<div class="team">):

A partir de aquí, se encuentran las tarjetas de los equipos de Fórmula 1. Cada tarjeta (<div>) representa un equipo y contiene la siguiente información:

- Imagen del equipo (<img>): Se muestra el logo del equipo.
- Lista sin viñetas (<ul>): Se utiliza para crear una lista de elementos sin viñetas.
- Elemento de lista (<li>): En este elemento, se muestra el nombre del equipo como un enlace (<a>) a una página web relacionada con ese equipo. Los enlaces se pueden abrir haciendo clic en el nombre del equipo.
- Imágenes (<img>):  
Se utilizan etiquetas de imagen para mostrar los logotipos de los equipos de Fórmula 1. Cada imagen se encuentra dentro de su tarjeta de equipo correspondiente. Cabe añadir que, el atributo src es imprescindible. Pues, se indica la ubicación de la imagen que queremos mostrar.

Y como el código es similar en todas las páginas HTML restantes, voy a explicar una y generalizar para las demás. Hay que añadir que la estructura base es la misma en todos los documentos HTML, es decir, el body, head, div, style, ...

```

<body>
  <header>
    <h1>AstonMartin</h1>
    
  </header>
  <section>
    <form action="index.html" method="post">
      <ul>
        <li>
          <h2>Vota por el mejor piloto</h2>
        </li>
        <li>
          <p><label for="name">Nombre:</label>
            <input type="text" id="name" name="user_name" required>
          </li>
        <li>
          <p><label for="mail">Correo electrónico:</label>
            <input type="email" id="mail" name="user_mail" required ></p>
          </li>
        <li>
          <p><label for="msg">Pilotos:</label>
            <input type="text" name="msg" id="msg" list="pilotos" required size="64" pattern="(Fernando Alonso|Lewis Hamilton|Max Verstappen|Daniel Ricciardo|Sergio Pérez|Nigel Mansell|Giancarlo Fisichella|Giancarlo Baghetti|Fernando Alonso|Esteban Ocon|Lance Stroll)">
            <datalist id="pilotos">
              <option value="Nigel Mansell"></option>
              <option value="Sergio Pérez"></option>
              <option value="Giancarlo Fisichella"></option>
              <option value="Giancarlo Baghetti"></option>
              <option value="Fernando Alonso"></option>
              <option value="Esteban Ocon"></option>
              <option value="Lance Stroll"></option>
            </datalist>
          </li>
        <li>
          <p><button type="submit">Enviar</button>
        </li>
      </ul>
    </form>
  </section>
  <section class="Aston">
    <h2>Equipo AstonMartin</h2>
    
    <a href="https://especiales.marca.com/motor/formula1/escuderias/2023/astonmartin.html">Aston Martin</a>
    <p>La escudería británica ha encontrado a su James Bond particular, solo que esta vez, proviene de Aston Martin. El AMR23 es un coche "totalmente nuevo al 90%y con innovaciones inteligentes", según Eric Blandin, director de desarrollo.
  </section>
  <section>
    <h2>Mejores momentos de AstonMartin</h2>
    <iframe width="560" height="315" src="https://www.youtube.com/embed/TPz8121LNv4?si=DU-TuYXSHyuZ1LLB" title="Mejores momentos de AstonMartin" allowfullscreen>
  </section>
</body>
</html>

```

- Formulario (<form>):

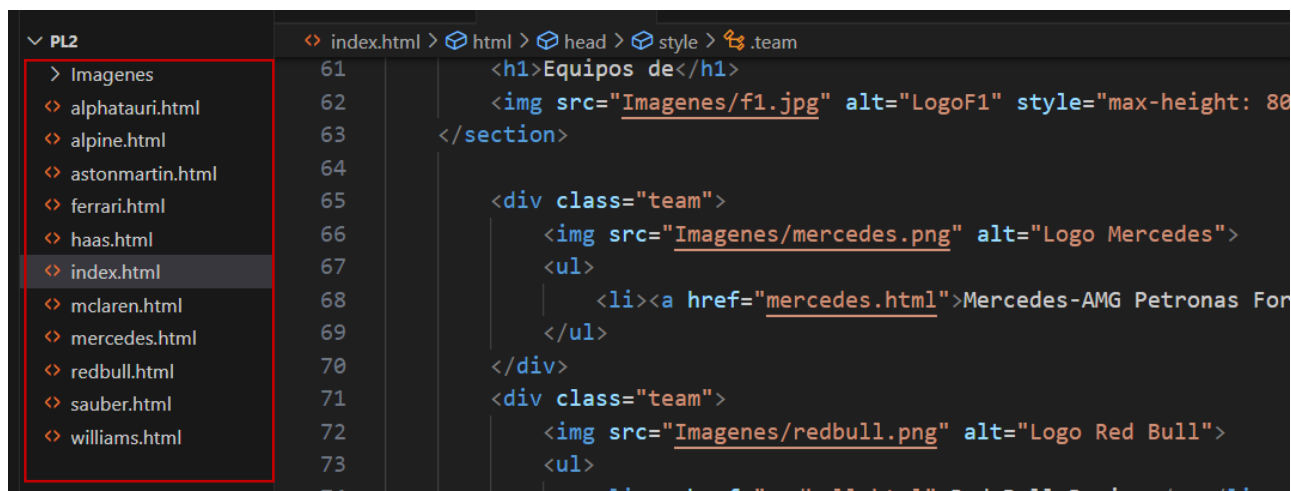
Se ha creado un formulario para permitir a los usuarios votar por el mejor piloto. Hay que añadir la función del atributo *action*, sirve para indicar que, una vez finalizado el formulario se redirigirá a la página indicada. Y el atributo *method*, en este caso, se ha establecido como "post", lo que significa que los datos se enviarán en el cuerpo de la solicitud HTTP. Los elementos del formulario son los siguientes:

- <label>: Etiquetas para los campos del formulario que indican lo que se debe ingresar.
- <input>: Campos de entrada de datos que permiten a los usuarios escribir su nombre y correo electrónico. El atributo *required* indica que estos campos son obligatorios.

- `<datalist>`: Se utiliza para proporcionar opciones predefinidas para el campo de selección de pilotos. Los usuarios pueden seleccionar un piloto de la lista desplegable o escribirlo manualmente.
  - `<button>`: Un botón "Enviar" que los usuarios pueden hacer clic para enviar el formulario. Una vez hecho esto le redirige de manera automática al index.
- Sección de Aston Martin (`<section class="nombre_escuderia">`):  
En esta sección, se presenta información sobre el equipo de Fórmula 1 Aston Martin. Contiene los siguientes elementos:
  - `<img>`: Muestra una imagen del equipo.
  - `<a>`: Un enlace que lleva a una página externa con información adicional sobre Aston Martin.
  - `<p>`: Un párrafo de texto que describe la situación del equipo y las expectativas para la temporada.
- Videos de YouTube (`<iframe>`):  
Se ha incorporado un video de YouTube en la página utilizando un elemento `<iframe>`. Este elemento permite incrustar contenido de video en la página web. Los atributos `width` y `height` establecen las dimensiones del reproductor de video.

## VALIDACIÓN

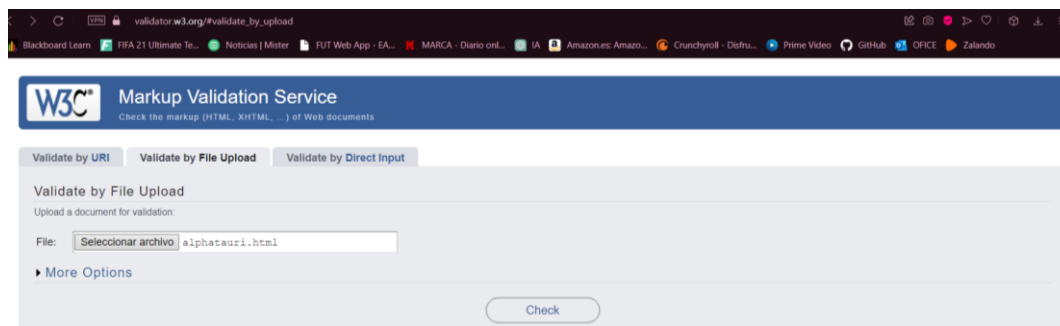
En este apartado trataremos de validar las diferentes páginas HTML de nuestra web.



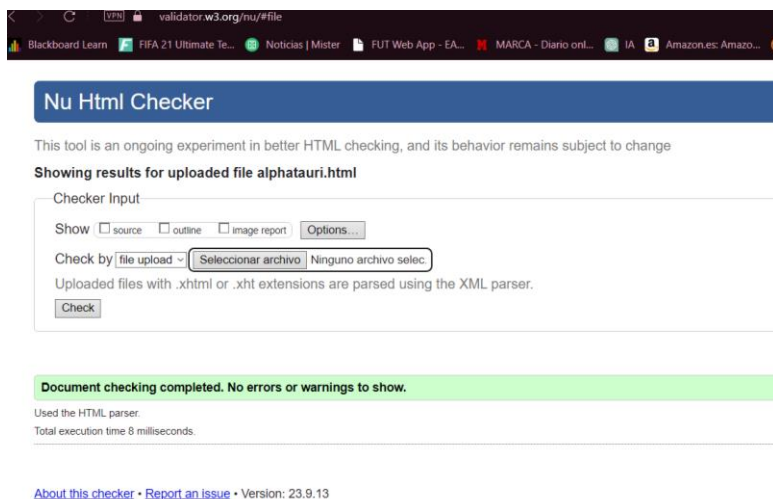
Esta es la estructura de nuestro código donde podemos ver las 10 clases de las correspondientes escuderías y a clase index que es donde se aloja la pagina principal de nuestra web explicada en la sección de creación.

Para validar nuestras paginas usaremos la herramienta que nos proporciona la web <http://validator.w3.org/>, vamos a seguir la estructura de la imagen por lo que empezaremos validando la página alphatauri y así hasta williams.

### Alphatauri.html



Seleccionamos la opción de **Validate by File Upload** y le pasamos el archivo y le damos a la opción de check.



**Nu Html Checker**

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for uploaded file alphatauri.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by   Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

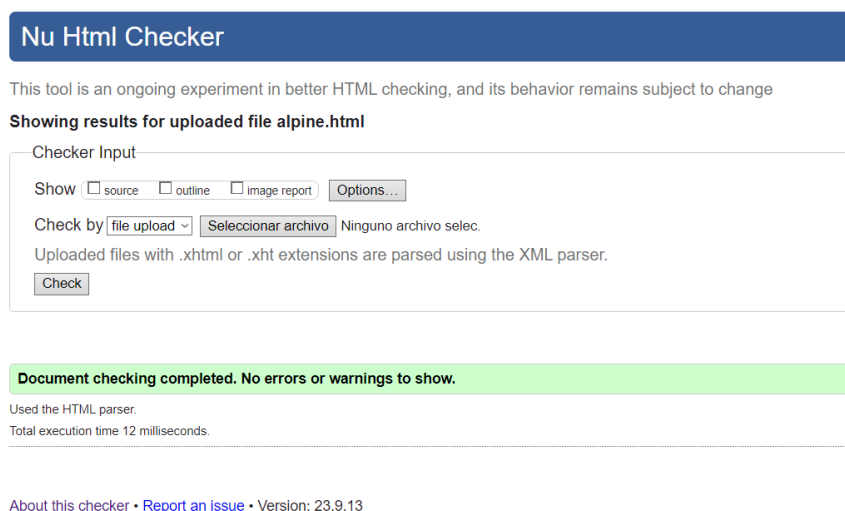
Used the HTML parser.  
Total execution time 8 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.9.13

Como podemos ver la pagina ha sido validada correctamente.

Procedemos a hacer lo mismo con las demás páginas.

## Alpine.html



**Nu Html Checker**

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for uploaded file alpine.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by   Ninguno archivo selec.

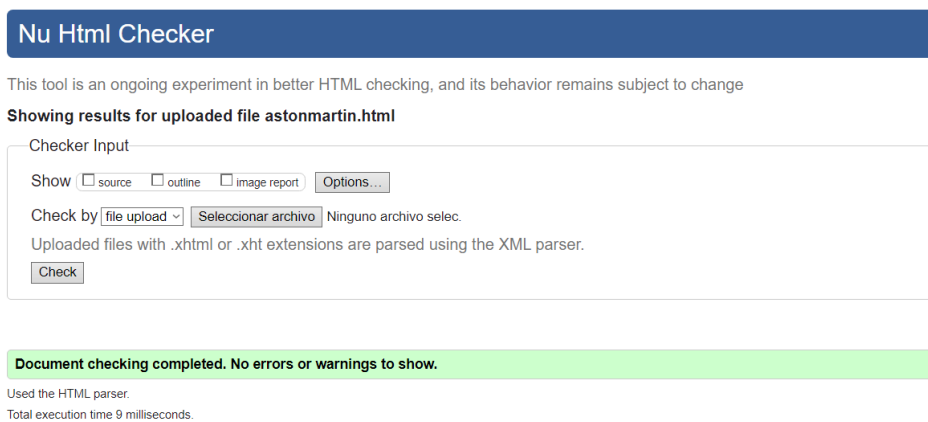
Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.  
Total execution time 12 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.9.13

## AstonMartin.html



**Nu Html Checker**

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for uploaded file astonmartin.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by   Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.  
Total execution time 9 milliseconds.



## Ferrari.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for uploaded file ferrari.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 13 milliseconds.

## Haas.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for haas.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 11 milliseconds.

## McLarem.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for uploaded file mclaren.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 14 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.8.18

## Mercedes.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for uploaded file mercedes.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 12 milliseconds.

## RedBull.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for uploaded file redbull.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 16 milliseconds.

## Sauber.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for sauber.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 9 milliseconds.

## Williams.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for williams.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.

Total execution time 13 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.9.13

## Index.html

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for index.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.

Total execution time 10 milliseconds.

Una vez que hemos validado todas las paginas nos disponemos a la publicación de la web en el servidor apache-tomcat.

## PUBLICACIÓN

Una vez creadas y validadas todas las páginas HTML, procedemos a su publicación.

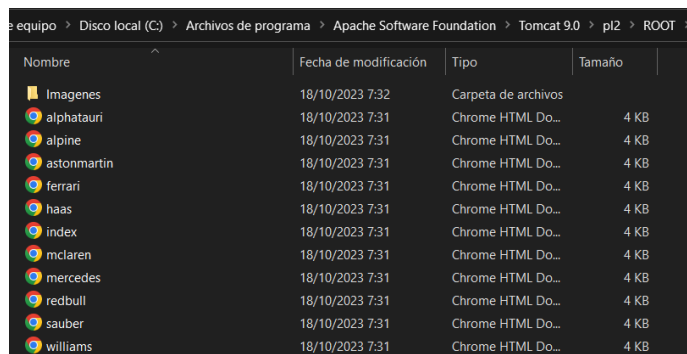
Para poner en funcionamiento la web en Tomcat, lo que hemos hecho nosotros es crear un nuevo servicio dentro del archivo `server.xml` (puede ser encontrado en la carpeta donde se haya instalado tomcat), quedando el siguiente fragmento de código:

```
<Service name="Practica2">
  <Connector port="8082" protocol = "HTTP/1.1" />
  <Engine name = "Practica2" defaultHost = "localhost">
    <Host name = "localhost" appBase = "pl2"></Host>
  </Engine>
</Service>
```

Aquí podemos ver que podremos acceder a la web accediendo a la dirección `localhost:8082`, entrando al `index.html` que es a donde accede Tomcat por defecto. Una vez que añadimos este servicio se nos creará una carpeta llamada `pl2` en la carpeta donde se encuentra Tomcat, tal y como se aprecia en la siguiente imagen:

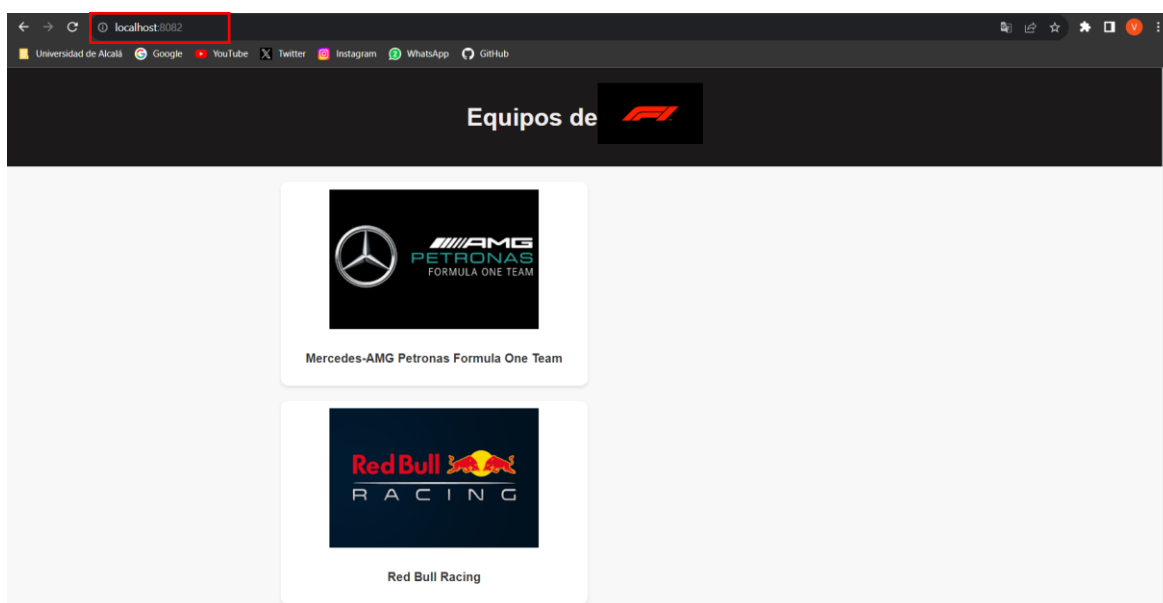
bin	28/09/2023 16:33	Carpeta de archivos	
conf	17/10/2023 11:43	Carpeta de archivos	
lib	28/09/2023 16:33	Carpeta de archivos	
logs	18/10/2023 7:33	Carpeta de archivos	
pl2	17/10/2023 11:43	Carpeta de archivos	
temp	28/09/2023 16:33	Carpeta de archivos	
vhost1	28/09/2023 23:56	Carpeta de archivos	
vhost2	02/10/2023 12:39	Carpeta de archivos	
vhost3	02/10/2023 12:39	Carpeta de archivos	
vhost4	02/10/2023 12:39	Carpeta de archivos	
webapps	28/09/2023 16:33	Carpeta de archivos	
work	17/10/2023 11:43	Carpeta de archivos	
LICENSE	23/08/2023 23:59	Archivo	57 KB
NOTICE	23/08/2023 23:59	Archivo	3 KB
RELEASE-NOTES	23/08/2023 23:59	Archivo	7 KB
tomcat	23/08/2023 23:59	Icono	22 KB
Uninstall	23/08/2023 23:59	Aplicación	86 KB

Una vez que tengamos la carpeta, simplemente añadimos todos los archivos de nuestra web, de manera que los tengamos tal y como se muestra a continuación:



Nombre	Fecha de modificación	Tipo	Tamaño
Imagenes	18/10/2023 7:32	Carpeta de archivos	
alphatauri	18/10/2023 7:31	Chrome HTML Do...	4 KB
alpine	18/10/2023 7:31	Chrome HTML Do...	4 KB
astonmartin	18/10/2023 7:31	Chrome HTML Do...	4 KB
ferrari	18/10/2023 7:31	Chrome HTML Do...	4 KB
haas	18/10/2023 7:31	Chrome HTML Do...	4 KB
index	18/10/2023 7:31	Chrome HTML Do...	4 KB
mclaren	18/10/2023 7:31	Chrome HTML Do...	4 KB
mercedes	18/10/2023 7:31	Chrome HTML Do...	4 KB
redbull	18/10/2023 7:31	Chrome HTML Do...	4 KB
sauber	18/10/2023 7:31	Chrome HTML Do...	4 KB
williams	18/10/2023 7:31	Chrome HTML Do...	4 KB

Una vez que tengamos los archivos metidos ahí, volvemos a arrancar Tomcat, y si introducidos *localhost:8082* nos abrirá el *index.html* tal y como se muestra en la siguiente imagen:



## DOCUMENTO XML

En este ejercicio se pedía hacer un documento XML que pudiera almacenar una carta de un restaurante. La carta se muestra a continuación:

Nombre	Precio	Descripción	Calorías
Puré de Patata	5,95€	Puré de patata roja con picatostes	350
Macarrones con queso	4,95€	Macarrones con queso parmesano	450
Garbanzos con chorizo	6,95€	Garbanzos cocidos con chorizo de Atienza	950
Judías verdes	3,95	Judías verdes cocidas de la variedad "Blanca de España"	350
Sopa de pescado	5,95€	Sopa de morralla de pescado	350
Merluza a la plancha	6,95€	Rodaja de merluza a la plancha con aceite de oliva	400

Por tanto, tendríamos que crear una estructura *Plato* que a su vez tuviera varias subcategorías dentro de la misma, las cuales son *Nombre*, *Precio*, *Descripción* y *Calorías*.

Una vez que tenemos la estructura clara, repliqué el contenido de la tabla siguiendo la estructura mencionada con anterioridad, y nuestro documento XML es el siguiente:

```
<?xml version = "1.0" encoding = "UTF-8"?>

<Plato>
  <Nombre>Pure de Patata</Nombre>
  <Precio>5.95</Precio>
  <Descripcion>Pure de patata roja con picatostes</Descripcion>
  <Calorias>350</Calorias>
</Plato>

<Plato>
  <Nombre>Macarrones con queso</Nombre>
  <Precio>4.95</Precio>
  <Descripcion>Macarrones con queso parmesano</Descripcion>
  <Calorias>450</Calorias>
</Plato>

<Plato>
  <Nombre>Garbanzos con chorizo</Nombre>
  <Precio>6.95</Precio>
  <Descripcion>Garbanzos cocidos con chorizo de Atienza</Descripcion>
  <Calorias>950</Calorias>
</Plato>

<Plato>
  <Nombre>Judias verdes</Nombre>
  <Precio>3.95</Precio>
  <Descripcion>Judias verdes cocidas de la variedad "Blanca de
Espana"</Descripcion>
  <Calorias>350</Calorias>
</Plato>
```

```
<Plato>
  <Nombre>Sopa de pescado</Nombre>
  <Precio>5.95</Precio>
  <Descripcion>Sopa de morralla de pescado</Descripcion>
  <Calorias>350</Calorias>
</Plato>

<Plato>
  <Nombre>Merluza a la plancha</Nombre>
  <Precio>6.95</Precio>
  <Descripcion>Rodaja de merluza a la plancha con aceite de
oliva</Descripcion>
  <Calorias>400</Calorias>
</Plato>
```

## CONCLUSIÓN

En esta práctica, se han abordado aspectos esenciales de la creación y gestión de contenido web utilizando HTML5, así como la representación estructurada de datos a través de documentos XML. Los objetivos planteados en la introducción, que incluyen la elección de un editor de texto adecuado, la creación de páginas web, su validación, la publicación en un servidor Apache-Tomcat y la creación de un documento XML para representar una carta de restaurante, se han cumplido con éxito. Estos aspectos son fundamentales para la creación y gestión de contenido digital.