Universidad de Alcalá

Arquitectura y Diseño de Sistemas Web y C/S

Práctica 3:

HTML 5 y CSS3. JavaScript y jQuery

Grupo 9

Universidad de Alcalá

Contenido

Introducción

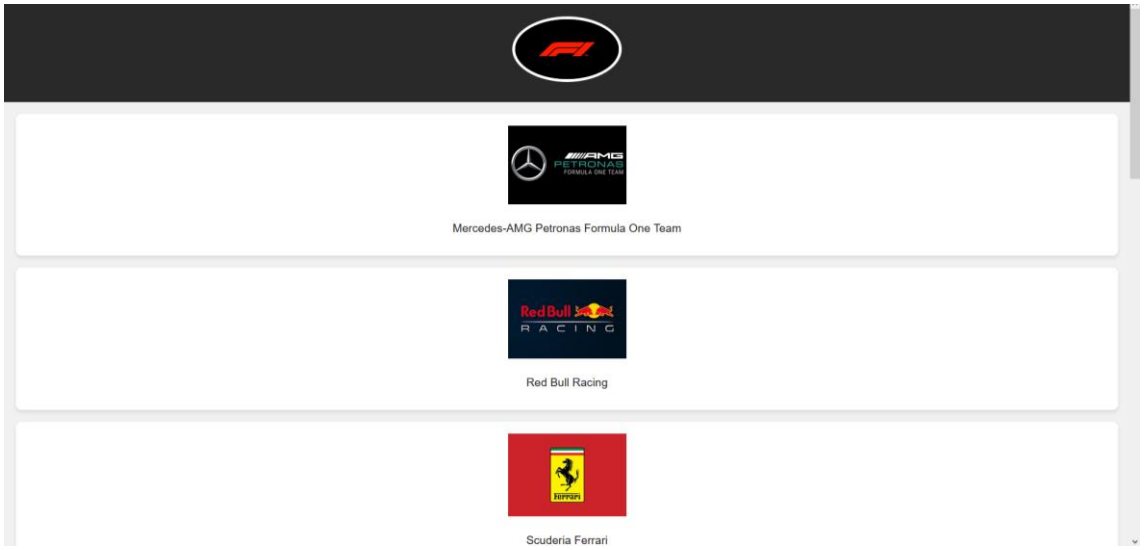
HTML 5

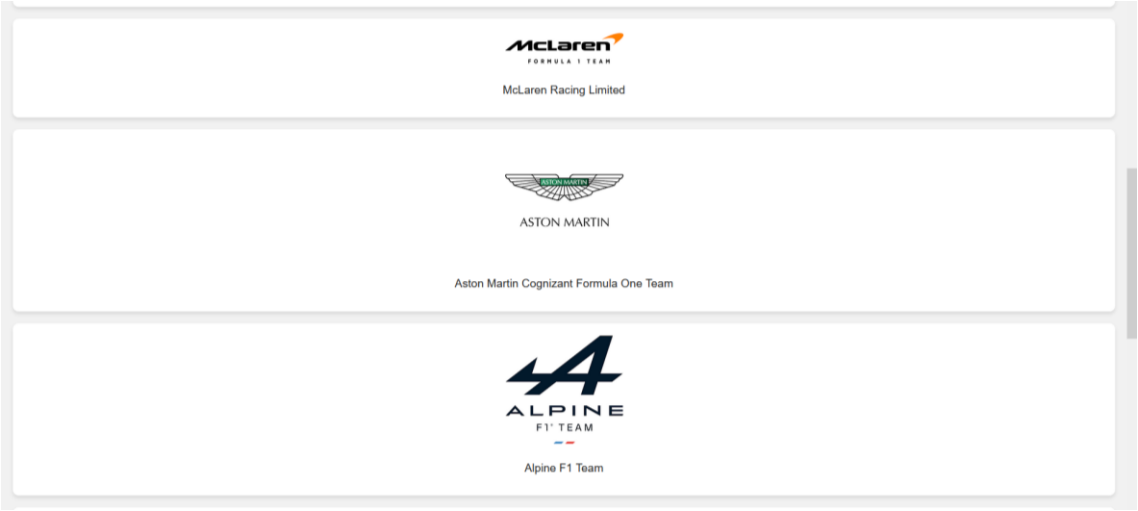
Aplicación de CSS en una web

Partiendo de la aplicación que ya diseñamos en la practica 2, vamos a crear una versión dos añadiendo css.

Esta miniaplicación se basa en un menú de los 10 equipos pertenecientes a la f1, y dentro de cada equipo podemos encontrar información de sus pilotos, hacer encuestas, saber datos sobre sus victorias, carreras, etc.

El menú principal:





Tenemos un menú desplegable en el que el usuario podrá elegir el equipo que mas le guste para poder consultar información.

Ya que lo importante de este apartado es la parte de hojas de estilo vamos a ir explicando el código, y la parte de html se podrá consultar en la carpeta **PL3_Aplicacion_de_CSS_en_una_web**

Ahora veremos el código css de esta parte, todo esta comentado para que se entienda que hemos hecho en cada parte:

```css
/* Estilos para el cuerpo del documento */
body {
    font-family: Arial, sans-serif; /* Fuente del texto */
    background-color: #f1f1f1; /* Color de fondo */
    margin: 0; /* Elimina el margen predeterminado del cuerpo */
    padding: 0; /* Elimina el relleno predeterminado del cuerpo */
}

/* Estilos para las secciones del documento */
section {
    text-align: center; /* Centra el contenido */
    padding: 30px 0; /* Aumenta el espacio en la parte superior e inferior */
    color: #fff; /* Color del texto */
    background-color: #292929; /* Fondo de la sección */
}

/* Estilos para los títulos de las secciones */
section h1 {
    font-size: 3em; /* Tamaño del texto */
    margin-bottom: 10px; /* Margen en la parte inferior */
}
```

```css
/* Estilos para las imágenes dentro de las secciones */
section img {
    max-width: 175px; /* Ancho máximo */
    height: auto; /* Altura automática */
    border-radius: 50%; /* Bordes redondeados */
    border: 5px solid #fff; /* Borde blanco */
}

/* Estilos para el contenedor de equipos */
.team {
    text-align: center; /* Centra el contenido */
    margin: 20px; /* Margen exterior */
    background-color: #fff; /* Color de fondo */
    border-radius: 10px; /* Bordes redondeados */
    padding: 20px; /* Relleno interno */
    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1); /* Sombra */
}

/* Estilos para las imágenes dentro del contenedor de equipos */
.team img {
    max-width: 200px; /* Ancho máximo */
    margin-bottom: 10px; /* Margen en la parte inferior */
}

/* Estilos para la lista de miembros del equipo */
.team ul {
    list-style: none; /* Elimina la viñeta de la lista */
    padding: 0; /* Elimina el relleno de la lista */
}

/* Estilos para los elementos de la lista de miembros del equipo */
.team ul li {
    font-size: 1.2em; /* Tamaño del texto */
    margin: 10px 0; /* Margen en la parte superior e inferior */
}

/* Estilos para los enlaces dentro de la lista de miembros del equipo */
.team ul li a {
    color: #292929; /* Color del texto */
    text-decoration: none; /* Elimina la decoración de enlace */
    transition: color 0.3s; /* Transición de color */
}
```

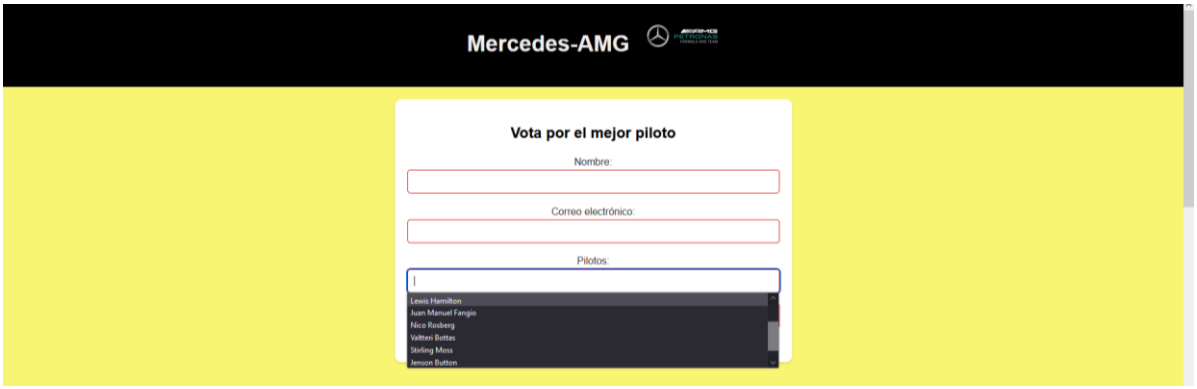
```
/* Estilos al pasar el ratón sobre los enlaces */
.team ul li a:hover {
    color: #f00; /* Cambia el color al pasar el ratón */
}
```

Ahora pasaremos a ver como es la interfaz de cada equipo ya que todos los equipos tienen la misma interfaz, solo vamos a mostrar dos para que se vea la estructura principal.

Ferrari.html

Todos los equipos siguen la misma estructura, vamos a ver un equipo mas para ver su funcionamiento:

MERCEDES.html



Ahora veremos lo mas destacado del código css, para entender como hemos diseñado esta interfaz. De igual manera que la anterior vamos a ir comentado todo línea por línea para que no quede muy largo.

```
/* Estilos para el body y fuentes */
body {
    font-family: Arial, sans-serif; /* Se establece la fuente para el cuerpo del documento, dando prioridad a Arial y luego a cualquier fuente sans-serif disponible */
    margin: 0; /* Se elimina el margen predeterminado del cuerpo del documento */
    padding: 0; /* Se elimina el relleno predeterminado del cuerpo del documento */
    background-color: #f7f571; /* Se establece el color de fondo del cuerpo del documento */
    text-align: center; /* Se centra el texto dentro del cuerpo del documento */
}

/* Estilos para el encabezado */

h1 {
    font-size: 36px; /* Se establece el tamaño de fuente para los elementos h1 */
    margin: 0; /* Se elimina el margen predeterminado de los elementos h1 */
    margin-right: 25px; /* Se establece un margen derecho de 25 píxeles para los elementos h1 */
}
header img {
    max-width: 100%; /* Se establece el ancho máximo de la imagen dentro del encabezado al 100% del contenedor */
    height: auto; /* Se permite que la altura de la imagen se ajuste automáticamente para mantener la proporción */
    max-height: 120px; /* Se establece la altura máxima de la imagen dentro del encabezado en 120 píxeles */
}
```

Todas las cabeceras son iguales luego solo mostraremos dos para que se vea la estructura pero todas son similares.

```
#headerFerrari {
    background-color: #cf0e0e; /* Se establece el color de fondo para el encabezado del equipo Ferrari */
    color: #ffffff; /* Se establece el color del texto para el encabezado del equipo Ferrari */
    text-align: center; /* Se centra el texto dentro del encabezado del equipo Ferrari */
    padding: 20px 0; /* Se establece un relleno de 20 píxeles en la parte superior e inferior del encabezado del equipo Ferrari y ningún relleno en los lados */
    display: flex; /* Se utiliza un modelo de caja flexible para organizar los elementos en el encabezado del equipo Ferrari */
    justify-content: center; /* Se centran los elementos horizontalmente dentro del encabezado del equipo Ferrari */
    align-items: center; /* Se centran los elementos verticalmente dentro del encabezado del equipo Ferrari */
}

#headerRedbull {
    background-color: #0B162F; /* Se establece el color de fondo para el encabezado del equipo Redbull */
    color: #f2eeee; /* Se establece el color del texto para el encabezado del equipo Redbull */
    text-align: center; /* Se centra el texto dentro del encabezado del equipo Redbull */
    padding: 20px 0; /* Se establece un relleno de 20 píxeles en la parte superior e inferior del encabezado del equipo Redbull y ningún relleno en los lados */
    display: flex; /* Se utiliza un modelo de caja flexible para organizar los elementos en el encabezado del equipo Redbull */
    justify-content: center; /* Se centran los elementos horizontalmente dentro del encabezado del equipo Redbull */
    align-items: center; /* Se centran los elementos verticalmente dentro del encabezado del equipo Redbull */
}
```

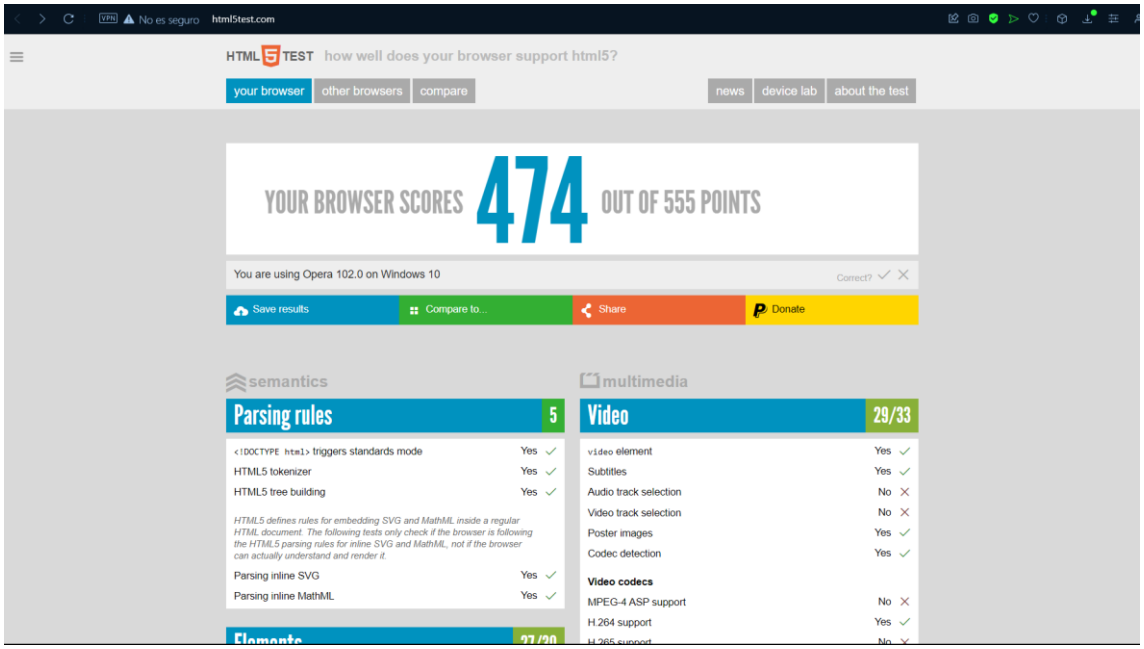
Esto seria la parte mas destacada el resto como hemos comentado anteriormente se podría ver en la carpeta **PL3_Aplicacion_de_CSS_en_una_web.**

Comprobación de funcionalidades HTML 5 en navegadores

En este apartado expondremos si los navegadores más usados cumplen con todas las funcionalidades HTML 5. Para ello examinaremos 3 navegadores que, creemos, son los más usados: Firefox, Opera, Microsoft Edge y Google Chrome.

Para estudiar el navegador que estamos ejecutando, basta con dirigirnos a la página web: http://html5test.com

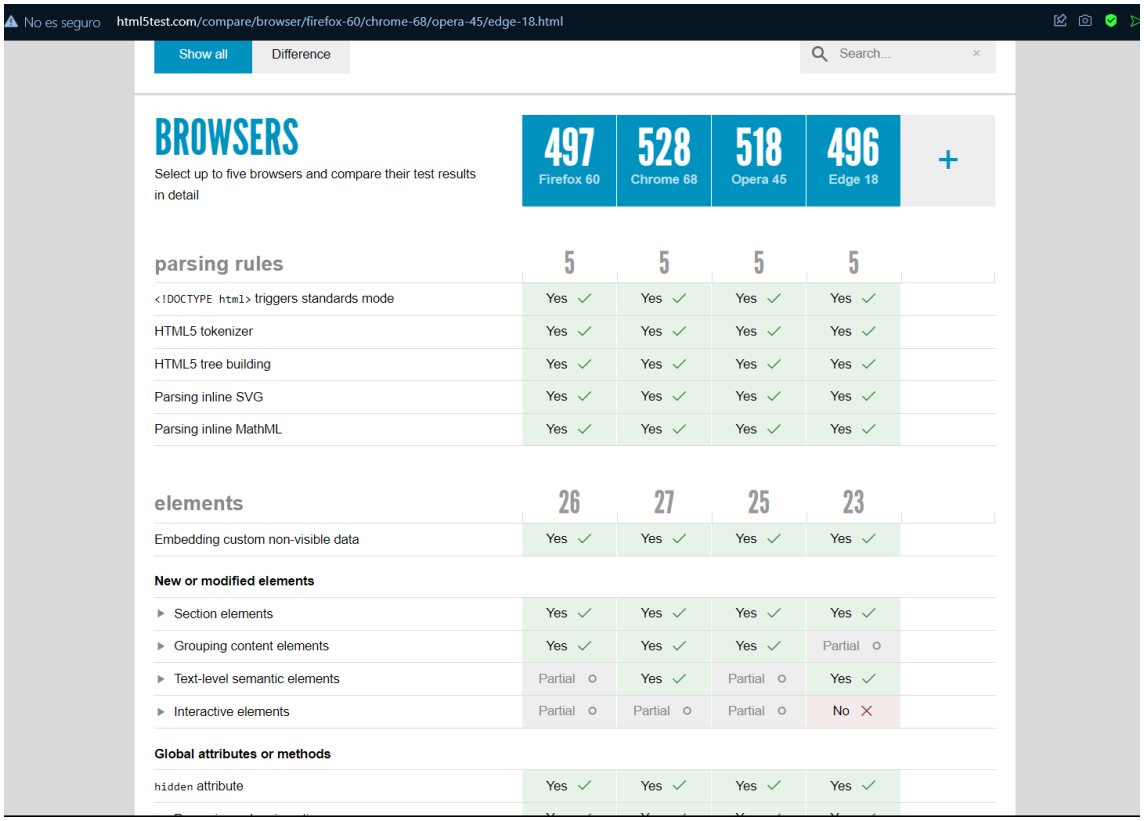
Aquí nos saldrá la puntuación de nuestro navegador respecto a las funcionalidades totales de HTML5. Por ejemplo:



Como podemos observar cumple con 474 funcionalidades.

Esta página web tiene algo interesante y es que puedes comparar las diferentes versiones de navegadores que no tienes instalados para que puedas escoger uno que cumpla con todas las funcionalidades. Para ello nos dirigimos a: http://html5test.com/compare/browser/index.html

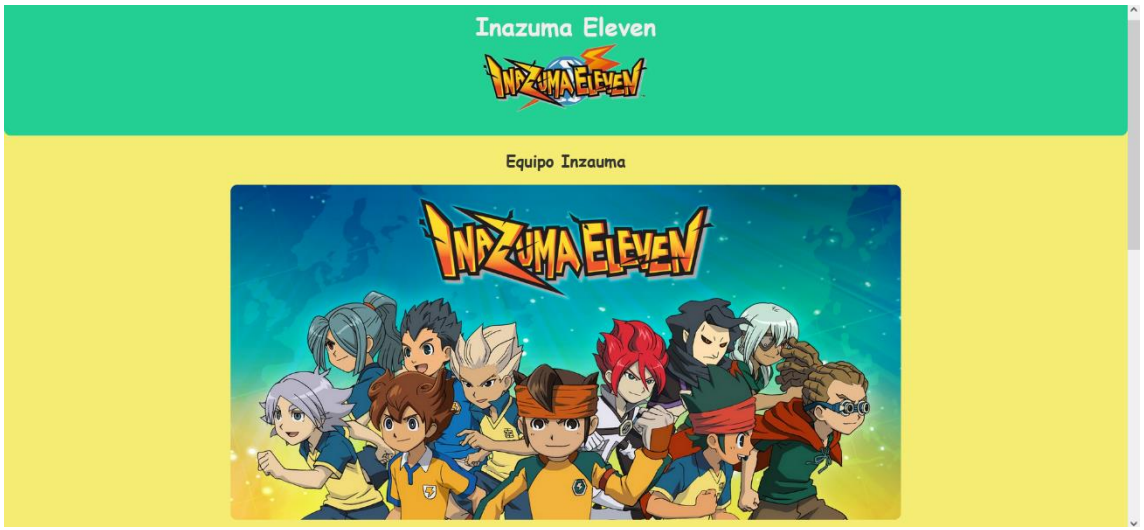
Una menor nota de validación quiere decir que esas funcionalidades no van a poder ser usadas por el navegador. Aunque esto no significa que la web no pueda ser cargada.

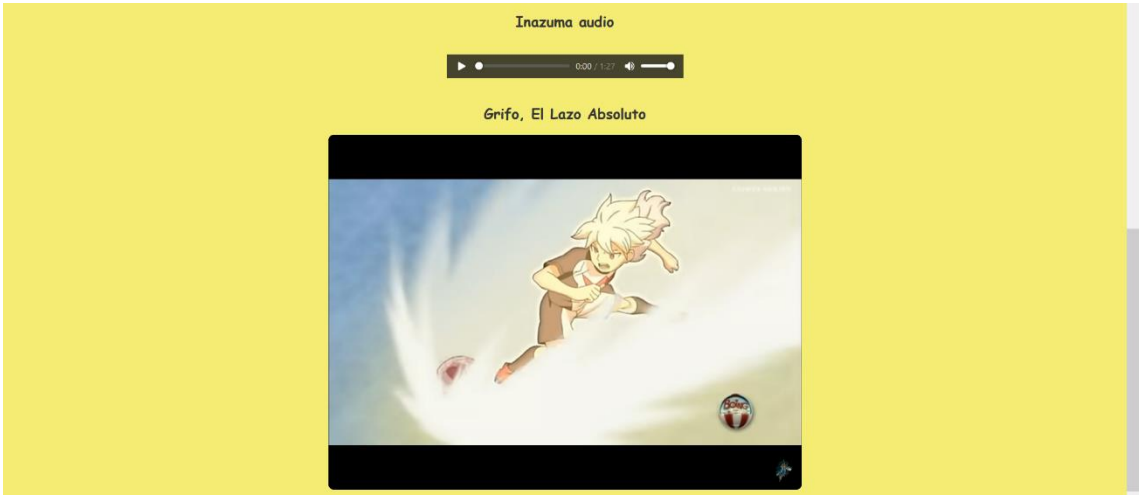


En este caso preferiríamos escoger Google Chrome, pues su puntuación es mayor.

Creación de página con elementos multimedia

En cuanto a este apartado en la que nos pedía crear una pagina con elementos multimedia sobre una película, que en este caso será la de Inazuma eleven. Vamos a ver el diseño de la interfaz y luego comentaremos los aspectos más relevantes del código.





La página esta compuesta por tres elementos multimedia una imagen, un audio y una película en formato video, vamos a ver como se ha hecho posible en código.

```
<body>
        <header>
        <h1>Inazuma Eleven</h1>
        <img src="logoInazuma.webp" alt="LogoInzuma">
        </header>
```

Dentro del body, hay una sección que se define como header, que generalmente se utiliza para incluir encabezados o cabeceras. En este caso, contiene un título (<h1>) que dice "Inazuma Eleven" y una imagen (<img>) con la fuente (src) "logoInazuma.webp" y el texto alternativo (alt) "LogoInzuma". Esto mostrará un título grande y una imagen del logo de Inazuma Eleven.

```
        <h2>Equipo Inzauma</h2>
        <img src="inazuma.jpg" alt="Equipo Inzazuma">
```

Aquí se incluye un encabezado de segundo nivel (<h2>) que dice "Equipo Inzauma". Luego, hay otra imagen con la fuente "inazuma.jpg" y el texto alternativo "Equipo inazuma". Esto mostrará un subtítulo y una imagen que parece estar asociada con el equipo Inazuma.

```
        <h2 id="audio">Inazuma audio</h2>
        <audio src="cancionInazuma.mp3" controls></audio>
```

Aquí hay otro encabezado de segundo nivel que dice "Inazuma audio". Luego, se incluye un elemento de audio (<audio>) con la fuente "cancionInazuma.mp3". El atributo controls agrega controles de reproducción al reproductor de audio, permitiendo a los usuarios reproducir, pausar y ajustar el volumen del audio.

```
        <h2>Grifo, El Lazo Absoluto</h2>
        <iframe width="560" height="315"
src="https://www.youtube.com/embed/9iQrQRG9O7k?si=qCCiMQV__PVwd71r"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>
</body>
```

Aquí se incluye otro encabezado de segundo nivel que dice "Grifo, El Lazo Absoluto". Luego, se utiliza un elemento iframe para incrustar un video de YouTube. El src especifica la fuente del video, que es una URL de YouTube. Los otros atributos proporcionan configuraciones para el reproductor de video.

Pasamos a ver lo más destacado de la parte de css:

Para enlazar el fichero html con css.

```
<link rel="stylesheet" href="estilo.css">
```

Luego ya dentro de css.

```
/* Estilos para el cuerpo de la página */
body {
    font-family: 'Arial', sans-serif; /* Selecciona la fuente y su
alternativa */
    margin: 0; /* Establece el margen exterior como cero */
    padding: 0; /* Establece el relleno interior como cero */
    text-align: center; /* Centra el texto dentro del cuerpo */
    background-color: #f5ec73; /* Establece el color de fondo como rojo mate
*/
}

/* Estilos para el encabezado */
header {
    background-color: #23cf93; /* Establece el color de fondo del encabezado
*/
    color: #f2eeee; /* Establece el color del texto en el encabezado */
    padding: 20px; /* Añade un relleno interno de 20px */
    border-radius: 10px; /* Añade bordes redondeados */
}

header h1 {
    margin: 0; /* Establece el margen superior e inferior de h1 como cero */
    font-size: 36px; /* Establece el tamaño de fuente de h1 */
    font-family: 'Pacifico', cursive; /* Cambia la fuente de h1 */
}

header img {
    max-width: 100%; /* Establece el ancho máximo de la imagen al 100% del
contenedor */
    height: auto; /* Ajusta la altura automáticamente */
    max-height: 105px; /* Establece la altura máxima de la imagen */
```

```
}

/* Estilos para los contenidos */
.container {
    margin: 20px; /* Establece un margen exterior de 20px */
    padding: 20px; /* Añade un relleno interno de 20px */
    background-color: #ffffff; /* Establece el color de fondo del contenedor
*/
    border-radius: 10px; /* Añade bordes redondeados */
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /* Añade una sombra al
contenedor */
}

/* Estilos para los encabezados */
h2 {
    color: #333; /* Establece el color del texto de h2 */
    font-size: 25px; /* Establece el tamaño de fuente de h2 */
    font-family: 'Indie Flower', cursive; /* Cambia la fuente de h2 */
}

/* Estilos para las imágenes */
img {
    max-width: 100%; /* Establece el ancho máximo de la imagen al 100% del
contenedor */
    height: auto; /* Ajusta la altura automáticamente */
    border-radius: 10px; /* Añade bordes redondeados */
    margin-bottom: 10px; /* Establece un margen inferior de 10px */
    max-height: 500px; /* Establece la altura máxima de la imagen */
}

/* Estilos para el reproductor de audio */
audio {
    width: 100%; /* Establece el ancho al 100% del contenedor */
    max-width: 400px; /* Establece el ancho máximo de 400px */
    margin: 20px 0; /* Establece un margen de 20px arriba y abajo */
}
```

Ejercicio práctico de creación de páginas con otras funcionalidades

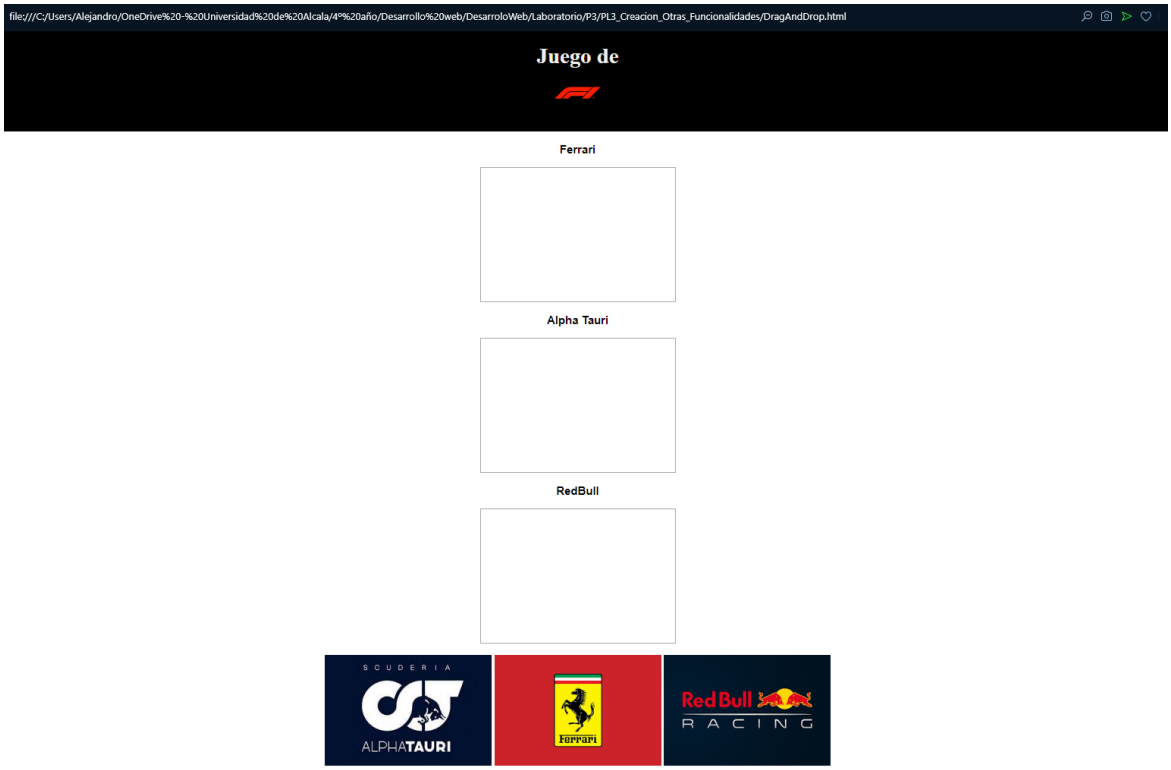
Drag & Drop

Para esta nueva funcionalidad hemos implementado una web externa que permita seleccionar una imagen (coger/agarrar) y soltarla en un cuadrado. Además, no se podrán poner dos imágenes en el mismo. Hay que añadir que se ha usado JavaScript para poder implementar la funcionalidad.

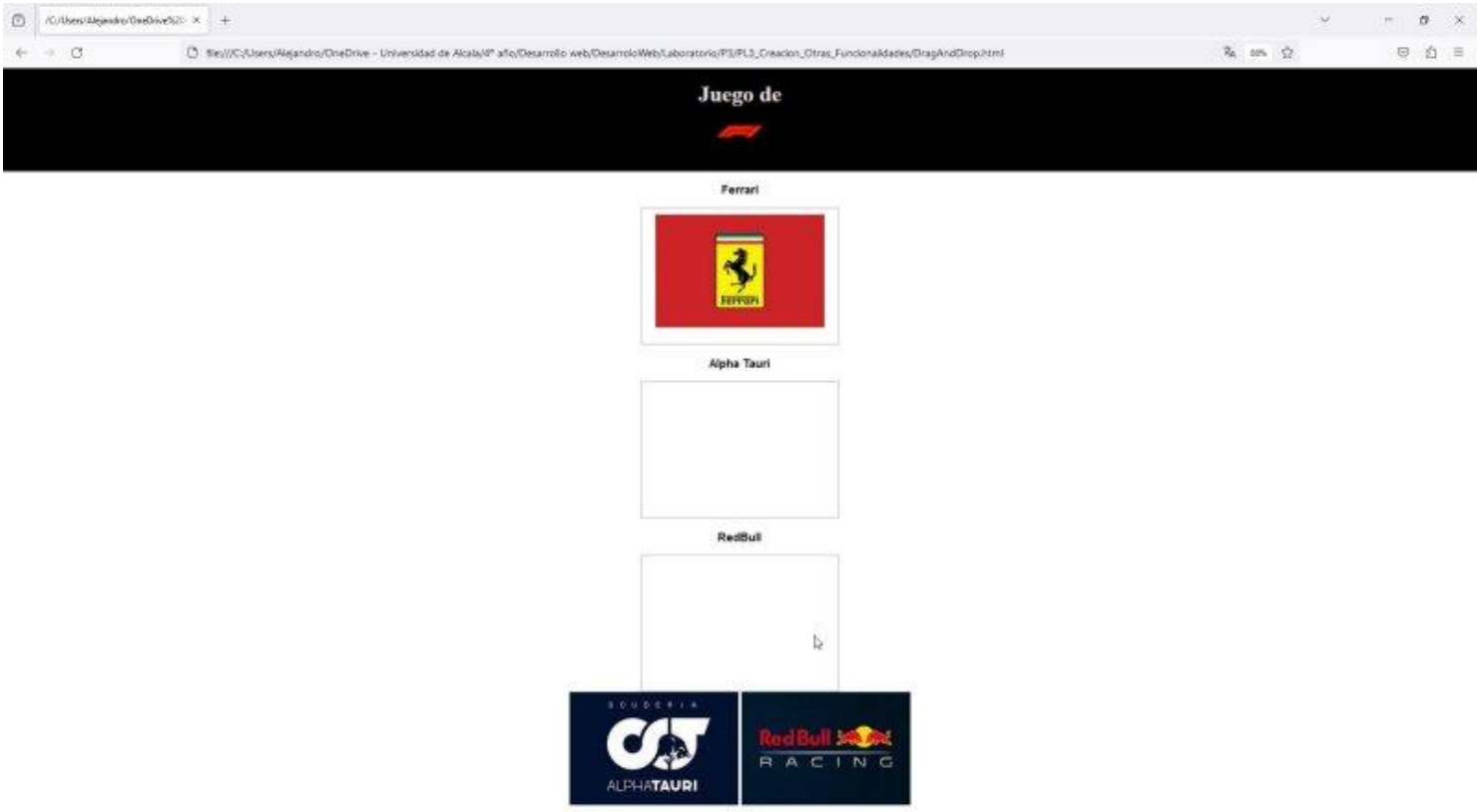
```
<!DOCTYPE html>
<html lang="en">

<head>

    <link rel="stylesheet" href="DaDstyle.css">
    <script>
        // Función que permite soltar un elemento arrastrado
        function permitirDrop(ev) {
            // Evita el comportamiento predeterminado del navegador
            ev.preventDefault();
        }

        // Función que se activa el drag de un elemento
        function drag(ev) {
            // Configura los datos del drag event con  ID del elemento
            ev.dataTransfer.setData("text", ev.target.id);
        }

        // Función soltar un elemento arrastrado
        function drop(ev) {
            ev.preventDefault();

            // Obtiene el ID del elemento drag
            var data = ev.dataTransfer.getData("text");

            // Obtiene el elemento de destino donde se soltará
            var targetElement = ev.target;

            // Si es "div1", anexa el elemento arrastrado a este contenedor
            if (targetElement.id === "div1") {
                targetElement.appendChild(document.getElementById(data));
            } else {
                //Muestra una alerta
                alert("El área de destino ya contiene una imagen o no es el área permitida para soltar.");
            }
        }
    </script>
</head>

<body>

    <header>
        <h1>Juego de </h1>
        <img src="Imagenes/f1.jpg" alt="LogoF1" style="max-height: 80px;">
    </header>

    <div id = "contenedor">
        <h3>Ferrari</h3>
        <div id="div1" ondrop="drop(event)" ondragover="permitirDrop(event)"></div>
        <h3>Alpha Tauri</h3>
        <div id="div1" ondrop="drop(event)" ondragover="permitirDrop(event)"></div>
        <h3>RedBull</h3>
        <div id="div1" ondrop="drop(event)" ondragover="permitirDrop(event)"></div>
        <br>
    </div>
        <img id="drag1" src="Imagenes/alpha.jpeg " draggable="true" ondragstart="drag(event)" width="275" height
        <img id="drag2" src="Imagenes/ferrari.png " draggable="true" ondragstart="drag(event)" width="275" heigh
        <img id="drag3" src="Imagenes/redbull.png " draggable="true" ondragstart="drag(event)" width="275" heigh
```

Tenemos tres funciones que son: coger(drag), soltar(drop) y permitirDrop(para poder soltar en ese div).

Y el HTML se presenta tal que así:



Así funciona nuestra página HTML:



Geolocalización

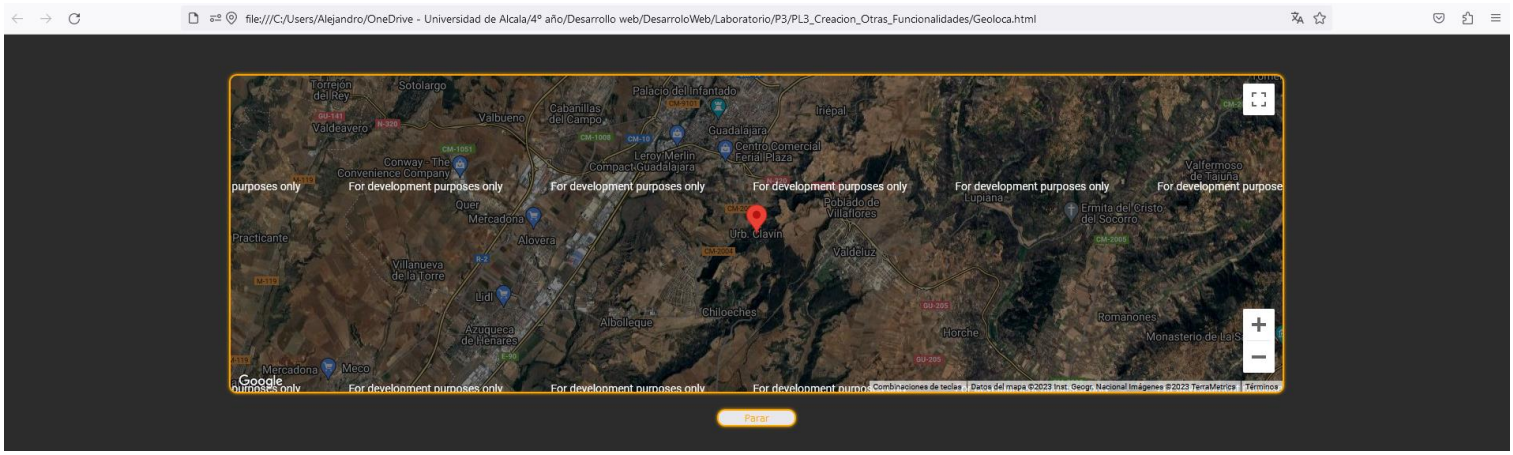
Esta funcionalidad consiste en que, usando la localización del dispositivo, el navegador sea capaz de posicionarte con una latitud y una longitud.

Nosotros además de usar esta funcionalidad, hemos usado el API de Google Maps para que pueda también posicionarte en un mapa. Recomendamos usar Firefox para que sea lo más preciso posible.

El código es este. Al igual que antes, hemos usado Javascript para poder crear esta funcionalidad.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo Geolocalización</title>
  <script type="text/javascript" src="https://code.jquery.com/jquery-1.10.1.min.js"></script>
  <script type="text/javascript" src="https://maps.google.com/maps/api/js?sensor=true"></script>
  <link rel="stylesheet" type="text/css" href="estilos.css">
  <script>
    var watchId;
    var mapa = null;
    var mapaMarcador = null;

    if (navigator.geolocation) {
      watchId = navigator.geolocation.watchPosition(mostrarPosicion, mostrarErrores, opciones);
    } else {
      alert("Tu navegador no soporta la geolocalización, actualiza tu navegador.");
    }

    function mostrarPosicion(posicion) {
      var latitud = posicion.coords.latitude;
      var longitud = posicion.coords.longitude;
      var precision = posicion.coords.accuracy;

      var miPosicion = new google.maps.LatLng(latitud, longitud);

      // Se comprueba si el mapa se ha cargado ya
      if (mapa == null) {
        // Crea el mapa y lo pone en el elemento del DOM con ID mapa
        var configuracion = {center: miPosicion, zoom: 16, mapTypeId: google.maps.MapTypeId.HYBRID};
        mapa = new google.maps.Map(document.getElementById("mapa"), configuracion);

        // Crea el marcador en la posicion actual
        mapaMarcador = new google.maps.Marker({position: miPosicion, title:"Esta es tu posición"});
        mapaMarcador.setMap(mapa);
      } else {
        // Centra el mapa en la posicion actual
        mapa.panTo(miPosicion);
        // Pone el marcador para indicar la posicion
        mapaMarcador.setPosition(miPosicion);
      }
    }

    function mostrarErrores(error) {
      switch (error.code) {
        case error.PERMISSION_DENIED:
            alert('Permiso denegado por el usuario');
            break;
          case error.POSITION_UNAVAILABLE:
            alert('Posición no disponible');
            break;
          case error.TIMEOUT:
              alert('Tiempo de espera agotado');
              break;
            default:
              alert('Error de Geolocalización desconocido :' + error.code);
      }
    }

    var opciones = {
      enableHighAccuracy: true,
      timeout: 10000,
      maximumAge: 1000
    };

    function detener() {
      navigator.geolocation.clearWatch(watchId);
    }
  </script>
</head>
<body>
  <article id="mapa">
  </article>
  <input type="button" id="parar" value="Parar" onclick="detener();"/>
</body>
```

Gracias al cual obtenemos algo así:



Por ejemplo, esto puede ser usado por las compañías para saber desde donde se compran sus productos para hacer un análisis de mercado posterior.

MiniSQL

Esta funcionalidad permite integrar una base de datos dentro de una página o servicio web, para ello hemos usado la base de datos que tienen integrada todos los navegadores, que en JavaScript aparece como *indexedDB*. Para ello, hemos hecho el siguiente programa en HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="sytles.css">
  <title>Base de Datos de Coches</title>
</head>
<body>

  <h1>Base de Datos de Coches</h1>

  <form id="formulario">
    <label for="marca">Marca:</label>
    <input type="text" id="marca" required>

    <label for="modelo">Modelo:</label>
    <input type="text" id="modelo" required>

    <label for="anio">Año:</label>
    <input type="number" id="anio" required>

    <button type="button" onclick="agregarCoche()">Agregar Coche</button>
  </form>

  <h2>Consultar Coches</h2>
  <button onclick="mostrarCoches()">Mostrar Coches</button>
  <ul id="lista-coches"></ul>

  <script src="script.js"></script>
</body>
</html>
```

Acompañado del siguiente fichero en CSS para hacer un mejor diseño:

```
body {
    font-family: 'Arial', sans-serif;
    background-color: #f2f2f2;
    color: #333;
    margin: 0;
    padding: 0;
  }

  h1, h2 {
```

```
    color: #e44d26; /* Cambiado a un tono de rojo */
    text-align: center;
  }

  form {
    max-width: 400px;
    margin: 20px auto;
    padding: 20px;
    background-color: #ffffff; /* Cambiado a blanco */
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 5px;
  }

  label {
    display: block;
    margin-bottom: 8px;
    font-weight: bold;
  }

  input {
    width: 100%;
    padding: 8px;
    margin-bottom: 16px;
    box-sizing: border-box;
    border: 1px solid #ccc;
    border-radius: 4px;
  }

  button {
    background-color: #e44d26; /* Cambiado a un tono de rojo */
    color: #fff;
    padding: 10px 15px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    display: block;
    margin: 0 auto; /* Centrar el botón */
  }

  button:hover {
    background-color: #c83a1d; /* Cambiado a un tono de rojo más oscuro
en hover */
  }

  h2 {
    margin-top: 40px;
  }

  ul {
```

```
    list-style-type: none;
    padding: 0;
    text-align: center;
  }

  ul li {
    background-color: #ffffff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 4px;
    padding: 10px;
    margin-bottom: 10px;
    text-align: center;
  }

  button, ul li {
    transition: background-color 0.3s ease;
  }
```

Y, por último, el siguiente script codificado en JavaScript para integrar la base de datos:

```
document.addEventListener("DOMContentLoaded", () => {
    let request = indexedDB.open("cochesDB", 1);
    let db;

    request.onerror = (event) => {
        console.error("Error al abrir la base de datos:",
event.target.errorCode);
    };

    request.onsuccess = (event) => {
        db = event.target.result;
        console.log("Base de datos abierta con éxito");
    };

    request.onupgradeneeded = (event) => {
        db = event.target.result;

        let cochesStore = db.createObjectStore("coches", { keyPath: "id",
autoIncrement: true });

        cochesStore.createIndex("marca", "marca", { unique: false });
        cochesStore.createIndex("modelo", "modelo", { unique: false });
        cochesStore.createIndex("anio", "anio", { unique: false });

        console.log("Base de datos creada y actualizada");
    };
```

```
    window.agregarCoche = () => {
        let marca = document.getElementById("marca").value;
        let modelo = document.getElementById("modelo").value;
        let anio = document.getElementById("anio").value;

        if (!marca || !modelo || !anio) {
            alert("Por favor, complete todos los campos.");
            return;
        }

        let transaction = db.transaction(["coches"], "readwrite");
        let cochesStore = transaction.objectStore("coches");

        let nuevoCoche = { marca, modelo, anio };

        let request = cochesStore.add(nuevoCoche);

        request.onsuccess = () => {
            console.log("Coche agregado con éxito");
            limpiarFormulario();
        };

        request.onerror = (event) => {
            console.error("Error al agregar el coche:",
event.target.errorCode);
        };
    };

    window.mostrarCoches = () => {
        let listaCoches = document.getElementById("lista-coches");
        listaCoches.innerHTML = "";

        let transaction = db.transaction(["coches"], "readonly");
        let cochesStore = transaction.objectStore("coches");

        let cursorRequest = cochesStore.openCursor();

        cursorRequest.onsuccess = (event) => {
            let cursor = event.target.result;

            if (cursor) {
                let coche = cursor.value;
                let listItem = document.createElement("li");
                listItem.textContent = `${coche.marca} ${coche.modelo}
(${coche.anio})`;
                listaCoches.appendChild(listItem);

                cursor.continue();
            }
```

```
        };

        cursorRequest.onerror = (event) => {
            console.error("Error al leer la base de datos:",
event.target.errorCode);
        };
    };

    function limpiarFormulario() {
        document.getElementById("marca").value = "";
        document.getElementById("modelo").value = "";
        document.getElementById("anio").value = "";
    }
});
```

La web funciona de la siguiente manera:

- En primer lugar, nos encontraremos con la siguiente disposición:

- Una vez que nos encontramos dentro de la página web, podremos añadir modelos de coches a nuestra base de datos rellenando todos los datos, si se deja un hueco sin rellenar, la página web nos avisará de este error, tal y como se muestra en la imagen:

- Cuando tengamos añadidos muchos coches en la base de datos, para comprobar todos los coches con sus características, simplemente pulsaremos el botón de *Mostrar Coches*:

- Para borrar los coches, bastaría con limpiar la caché del navegador que estaría la opción disponible dentro de la configuración de este.

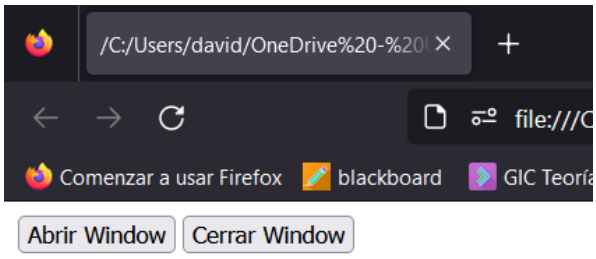
Ejemplos básicos de JavaScript

Ej1.Abrir una nueva ventana

Este código proporciona una página web con dos botones. Uno abre una nueva ventana con el sitio web de la Universidad de Alcalá y el otro cierra esa ventana.

```
<html>
 <head>
 <script language=javascript>
 function abrirwindow() {
 m = window.open("http://www.uah.es");
 }
 function cerrarwindow() {
 m.close()
 }
 </script>
 </head>
 <body>
 <form>
 <input type=button value="Abrir Window"
onclick="abrirwindow()">
 <input type=button value="Cerrar Window"
onclick="cerrarwindow()">
 </form>
 </body>
</html>
```

Vamos a ver su funcionalidad:



Si pulsamos abrir Window se abre https://www.uah.es/es/ , y si pulsamos cerrar se cierra la pagina abierta.

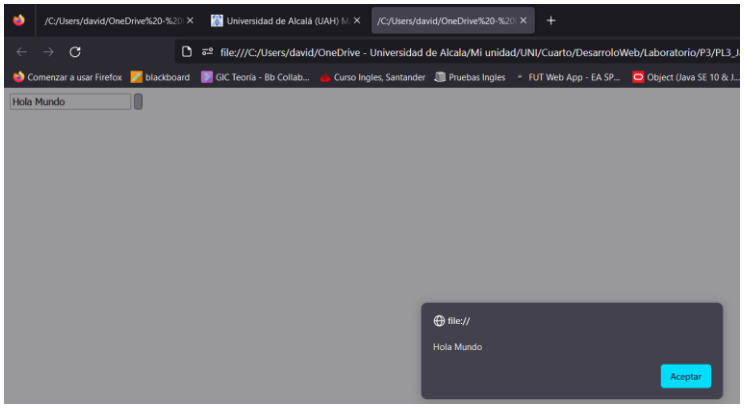
Ej2. Mostrar en una ventana un valor del formulario

Este código HTML con dos funciones de JavaScript que contiene una página con un campo de entrada de texto y un botón. Cuando se hace clic en el botón, se ejecuta una función de

JavaScript que muestra una ventana de alerta con el texto ingresado por el usuario en el campo de entrada.

```
<html>
 <head>
 <script language="javascript">

function ventana(txt){
 alert(txt)
 }
 </script>
 </head>
 <body>
 <form name="form1">
 <input name="entrada">
 <input type="button" onclick="ventana(form1.entrada.value)">
 </form>
 </body>
</html>
```

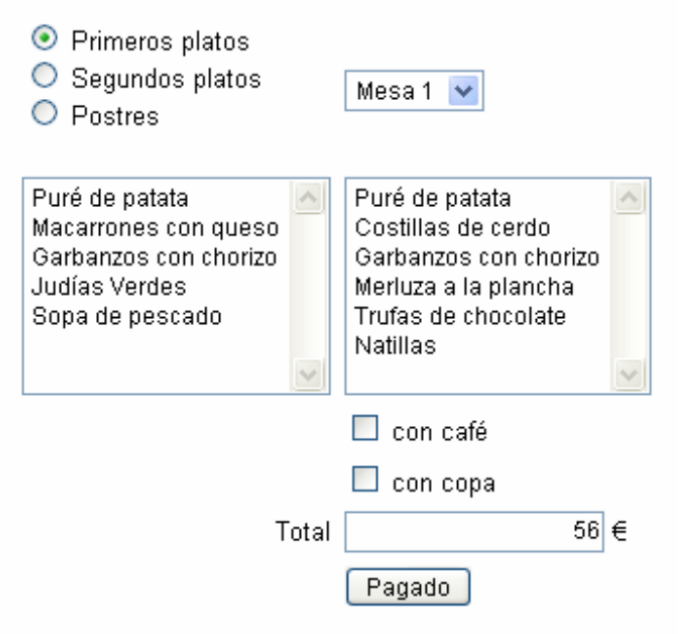
Vamos a ver como se vería esto en la web:



Como vemos salta una ventana de alerta con el mensaje que he introducido.

Ejercicio JavaScript

En este ejercicio lo que se pide es construir una aplicación de un restaurante que se base en el lenguaje JavaScript con la ayuda de la librería JQuery, para gestionar los pedidos de las mesas.



Funcionalidad:

En la **caja de la izquierda** tenemos la lista de comidas que ofrece el restaurante, mostrando en cada momento la lista de los platos seleccionados en los botones de opción de arriba. En la imagen se está mostrando la lista de Primeros Platos. Por tanto, dependiendo de la opción así cambiará la lista de comidas que el restaurante ofrece.

En la **caja de la derecha** se muestra la lista de platos totales que han pedido en esa mesa. Como se puede ver en la figura aparecen primeros platos, segundos y postres, es decir el total de platos servidos a esa mesa. La mesa se selecciona en el desplegable de la parte de arriba. Si cambiamos de mesa automáticamente deberá cambiar la elección de los platos y aparecer los de esa mesa. La aplicación debe soportar cinco mesas a la vez.

Luego podemos indicar si se ha pedido café y/o copa y se muestra el total a pagar por esa mesa. Cuando se pulsa el botón pagado se reinicia la lista de platos de esa mesa.

Inicialmente las mesas no tienen platos pedidos. Para añadir un plato a una mesa se selecciona la mesa, y se pulsa doble click en el plato que queremos añadir a la mesa. Cada plato tiene asociado un precio. Y el café y la copa tienen un precio añadido constante. Cada vez que se añade un plato se debe calcular el precio total a pagar.

Nuestra implementación en HTML ha sido como sigue:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <script src="jquery-3.7.1.js" type="text/javascript"></script>
    <link rel="stylesheet" href="index.css">
</head>
```

```
<body>
    <h1>Bachis Restaurant</h1>

    <form>
        <label>Selecciona tu plato:</label>
        <br>
        <input type="radio" name="plato" value="primer_plato"
id="P_plato"> Primer Plato
        <input type="radio" name="plato" value="segundo_plato"
id="S_plato"> Segundo Plato
        <input type="radio" name="plato" value="postre" id="Postre">
Postre
        <br><br>

        <label for="mesa">Selecciona tu mesa:</label>
        <select name="mesa" id="mesa">
            <option value="1">Mesa 1</option>
            <option value="2">Mesa 2</option>
            <option value="3">Mesa 3</option>
            <option value="4">Mesa 4</option>
            <option value="5">Mesa 5</option>
        </select>

        <br><br>
    </form>

    <div class="menu-container">
        <div class="menu">
            <select id="Opciones" size="5" multiple>
                <!-- Aquí se generan las opciones -->
            </select>
        </div>
    </div>

    <div class="menu-container-selecciones">
        <div class="menu">
            <select id="Selecciones" size="5" multiple>
                <!-- Aquí se mostrarán las selecciones -->
            </select>
        </div>
    </div>
    <br><br>

    <input type="checkbox" name="extras" value="cafe" id="Cafe_x"> Con
Café
    <input type="checkbox" name="extras" value="copa" id="Copa_x"> Con
Copa
    <br><br>
```

```
    <p>Total: <input type="text" placeholder="Total" id="total_cuenta"
required> €</p><br><br>
    <button type="submit" id="Pagar_Total">Pagado</button>

    <script src="./index.js"></script>
</body>
</html>
```

Hasta ahora no tenemos funcionalidad. Esta la obtenemos añadiendo la librería jQuery al HTML que haremos uso de ella en el JavaScript siguiente:

```
$(document).ready(function () {
    const platoCheckboxes = $('input[name="plato"]');
    const opcionesSelect = $('#Opciones');
    const seleccionesSelect = $('#Selecciones');
    const mesaSelect = $('#mesa');
    const extrasCheckboxes = $('input[name="extras"]');
    const totalCuentaInput = $('#total_cuenta');
    const pagarTotalBtn = $('#Pagar_Total');
    const platos = {
        primer_plato: ["Ensalada", "Sopa", "Ceviche", "Macarrones",
"Lentejas", "Pure", "Judias Pintas", "Arroz con pollo", "Croquetas de
jamón", "Tortilla española"],
        segundo_plato: ["Lasaña", "Pescado a la plancha", "Pollo asado",
"Cabrito", "Patatas Deluxe", "Entrecot a la parrilla", "Paella",
"Costillas a la barbacoa", "Sushi", "Hamburguesa"],
        postre: ["Tarta de chocolate", "Helado", "Frutas", "Flan",
"Brownie", "Mousse de limón", "Tiramisú", "Gelatina", "Crema catalana",
"Pastel de zanahoria"]
    };
    let platosPorMesa = {
        1: { platos: [], extras: [] },
        2: { platos: [], extras: [] },
        3: { platos: [], extras: [] },
        4: { platos: [], extras: [] },
        5: { platos: [], extras: [] }
    };
    let totalPorMesa = {
        1: 0,
        2: 0,
        3: 0,
        4: 0,
        5: 0
    }

    // Función para mostrar los platos según la selección de la mesa
    function mostrarPlatosPorMesa() {
```

```
        const mesaSeleccionada = mesaSelect.val();

        // Limpiar platos y extras seleccionados al cambiar la mesa
        seleccionesSelect.html('');

        // Mostrar los platos y extras correspondientes a la mesa actual
        mostrarPlatos(platosPorMesa[mesaSeleccionada].platos);
        mostrarExtras(platosPorMesa[mesaSeleccionada].extras);
    }

    // Asociar eventos a los checkboxes de plato
    platoCheckboxes.change(function () {
        mostrarPlatosPorMesa();
    });

    // Función para mostrar los platos según la selección
    function mostrarPlatos(platosMostrados) {
        opcionesSelect.html(''); // Limpiar las opciones

        const platosDisponibles =
obtenerPlatosDisponibles(mesaSelect.val());

        $.each(platosDisponibles, function (index, plato) {
            const option = $('<option>').text(plato);
            opcionesSelect.append(option);
            // Agregar evento de doble clic a cada opción generada
            option.dblclick(function () {
                const seleccion = $('<option>').text(plato);
                seleccionesSelect.append(seleccion);

                // Almacenar el plato seleccionado para la mesa actual
                platosPorMesa[mesaSelect.val()].platos.push(plato);

                calcularTotal();
            });
        });

        // Mostrar los platos previamente seleccionados en la nueva mesa
        $.each(platosPorMesa[mesaSelect.val()].platos, function (index,
plato) {
            const seleccion = $('<option>').text(plato);
            seleccionesSelect.append(seleccion);
        });
    }

    // Función para mostrar los extras según la selección
    function mostrarExtras(extrasMostrados) {
        extrasCheckboxes.prop('checked', false); // Desmarcar todos los
checkboxes
```

```
        $.each(extrasMostrados, function (index, extra) {
            extrasCheckboxes.filter(`[value="${extra}"]`).prop('checked',
true);
        });
    }

    // Asociar evento a la selección de mesa
    mesaSelect.change(mostrarPlatosPorMesa);

    // Función para obtener los platos disponibles según la categoría y
mesa
    function obtenerPlatosDisponibles(mesa) {
        const platoSeleccionado = obtenerPlatoSeleccionado();
        return platos[platoSeleccionado] || [];
    }

    // Función para obtener la categoría de plato seleccionada
    function obtenerPlatoSeleccionado() {
        let platoSeleccionado = '';
        platoCheckboxes.each(function () {
            if ($(this).prop('checked')) {
                platoSeleccionado = $(this).val();
            }
        });
        return platoSeleccionado;
    }

    // Función para calcular el total
    function calcularTotal() {
        let total = 0;

        // Sumar el precio de cada plato seleccionado
        $.each(platosPorMesa[mesaSelect.val()].platos, function (index,
plato) {
            total += obtenerPrecioPlato(plato);
        });

        // Sumar el precio de los extras (café o copa)
        $.each(platosPorMesa[mesaSelect.val()].extras, function (index,
extra) {
            total += obtenerPrecioExtra(extra);
        });

        // Mostrar el total en el campo
        totalCuentaInput.val(total.toFixed(2));
    }

    // Función para obtener el precio de un plato
```

```
    function obtenerPrecioPlato(plato) {
        const preciosPlatos = {
            "Ensalada": 10,
            "Sopa": 8,
            "Ceviche": 12,
            "Macarrones": 9,
            "Lentejas": 7,
            "Pure": 8,
            "Judias Pintas": 9,
            "Arroz con pollo": 11,
            "Croquetas de jamón": 10,
            "Tortilla española": 9,
            "Lasaña": 12,
            "Pescado a la plancha": 14,
            "Pollo asado": 11,
            "Cabrito": 15,
            "Patatas Deluxe": 9,
            "Entrecot a la parrilla": 16,
            "Paella": 13,
            "Costillas a la barbacoa": 14,
            "Sushi": 18,
            "Hamburguesa": 10,
            "Tarta de chocolate": 8,
            "Helado": 6,
            "Frutas": 5,
            "Flan": 7,
            "Brownie": 7,
            "Mousse de limón": 6,
            "Tiramisú": 9,
            "Gelatina": 5,
            "Crema catalana": 7,
            "Pastel de zanahoria": 8
        };

        return preciosPlatos[plato];
    }

    // Función para obtener el precio de un extra
    function obtenerPrecioExtra(extra) {
        const preciosExtras = {
            "cafe": 2,
            "copa": 2.5,
        };

        return preciosExtras[extra] || 0;
    }

    // Asociar eventos a los checkboxes de extras para recalcular el
total
```

```
    extrasCheckboxes.change(function () {
        const extrasSeleccionados = obtenerExtrasSeleccionados();
        platosPorMesa[mesaSelect.val()].extras = extrasSeleccionados;
        calcularTotal();
    });

    // Función para obtener los extras seleccionados
    function obtenerExtrasSeleccionados() {
        const extrasSeleccionados = [];
        extrasCheckboxes.each(function () {
            if ($(this).prop('checked')) {
                extrasSeleccionados.push($(this).val());
            }
        });
        return extrasSeleccionados;
    }

    // Asociar eventos al botón "Pagado" para eliminar los productos y
mostrar una alerta
    pagarTotalBtn.click(function () {
        const mesaSeleccionada = mesaSelect.val();
        const platosSeleccionados =
platosPorMesa[mesaSeleccionada].platos;

        if (platosSeleccionados.length > 0) {

            // Limpiar platos y extras seleccionados para la mesa actual
            platosPorMesa[mesaSeleccionada].platos = [];
            platosPorMesa[mesaSeleccionada].extras = [];
            seleccionesSelect.html('');
            totalCuentaInput.val(0);
            extrasCheckboxes.prop('checked', false);

            // Mostrar alerta
            alert('¡Pago realizado con éxito!');

        } else {
            alert('No hay productos seleccionados para pagar.');
        }
    });

    // Limpiar la selección de platos y extras al cambiar la mesa
    mesaSelect.change(function () {
        mostrarPlatosPorMesa();
        calcularTotal();
    });
});
```

Esto implementa todas las funcionalidades descritas.

El funcionamiento del programa es:

1. Seleccionamos una mesa (por defecto esta seleccionada la mesa 1)
2. Seleccionamos el plato que consumirán, es decir, escogemos primero, segundo o postre.
3. Seleccionamos el extra, podemos seleccionar uno, dos o ninguno.
4. Ahora pulsamos el botón para pagar que nos muestra una alerta como que el pago se ha efectuado con éxito.

Hay que añadir que si cambiamos de mesa la información de cada una se almacena, para poder atender otras mesas mientras mantenemos la información de cada una. Esto es para que podamos pagar en el momento que la mesa acabe de comer. También, el total a pagar se actualiza cada vez que añadimos un producto nuevo.

Conclusión

En conclusión, al realizar esta práctica nos ha permitido conocer la implementación y manejo de técnicas en el lado del cliente. También hay que destacar la utilidad del lenguaje JavaScript, ya que nos ha servido para implementar jQuery y una base de datos, junto con HTML5 y CSS. Sin duda estos conocimientos adquiridos nos servirán para proyectos futuros dentro del lado cliente.