

Buscador de coches

Desarrollo de aplicaciones de lado del servidor

Entrega para la asignatura de *Desarrollo de aplicaciones de lado del servidor* con el objetivo de comprobar el aprendizaje de AJAX. El ejercicio consiste en crear un buscador de coches a partir de un XML. Para más información puedes ver el [enunciado del ejercicio](#).

El repositorio tiene los siguientes archivos:

```
.
├── Actividad Buscador de Coches con AJAX.pdf
├── coches.xml
├── index.html
├── main.js
├── main.php
└── README.md
```

Arquitectura

Aunque es un ejercicio pequeño sigue el patrón MVC. El *coches.xml* es el modelo, *main.php* funciona como controlador únicamente sirviendo datos o generando archivos XML para su posterior descarga. Por último, la generación dinámica de la vista recae en *main.js* que es quien gestiona todas las peticiones al servidor para obtener los datos y renderizarlos.

index.html

Página que muestra el buscador de coches y renderiza cada uno de los resultados de búsqueda mediante AJAX con una petición **POST**.

coches.xml

XML facilitado por el profesor con el listado de coches a buscar. A medida que se añadan por el backend se añadirán en el XML.

main.js

Archivo JavaScript con todo el grueso del ejercicio. Las funciones que contiene son:

- **carsFilter(cars:JSON [, id:String])**: Configura y filtra todos los `selects` del buscador. Tiene el parámetro opcional *id* para definir a partir de qué `select` filtrar, si no pone todos los `selects` con todas las opciones.
- **renderCars(cars:JSON [, sortBy:String])**: Renderiza la tabla donde sale los resultados de la búsqueda. Tiene el parámetro opcional *sortBy* para ordenar ascendente o descendente y en base a qué campo. Es un parámetro que se la pasará a la función *sortCars* que se invoca dentro de *renderCars*.

- **selectFilter(selectID:String, cars:JSON)**: Devuelve el html con las opciones de un determinado `select`. El parámetro `cars` pasa el filtro que deben seguir las opciones y `selectID` determina para qué `select` deben realizarse las opciones.
- **showCar(id: String)**: función que renderiza la ficha del coche a la derecha de la página. El parámetro `id` definirá que coche es el que se renderizará.
- **sortCars(cars:JSON [, sortBy:String])**: Ordena el JSON que se le pasa y lo devuelve como un objeto. Tiene el parámetro opcional `sortBy` para ordenar ascendemente o descendente y en base a qué campo.
- **writeOptions(options:Array)**: Devuelve el html con todas las opciones pasadas al llamarlo.
- **writeRow(car: Object)**: Devuelve el html para una fila de la tabla con los datos del coche que se le ha pasado.

main.php

Archivo PHP con toda la parte de servidor del ejercicio. La funciones que contiene son:

- **searchCars(arrayCoches: Array, marca:String, motor:String, km:Number, precio:Number, ano:Number)**: Devuelve un array de objetos con los coches filtrados en función de los parámetros que se le pasan.
- **cars2Array(XML:Object)**: Devuelve un array con todos los objetos del objeto XML pasado por parámetro.
- **getCar(Coches:Object, id:String)**: Devuelve un objeto cuya ID coincida con la `id` pasada por parámetro.
- **sendXML(Coches:Object, id:String)**: Crea un fichero XML de un coche en concreto basándose en la `id` que se le pasa por parámetro y envía una cabecera http para realizar la descarga automática del XML que se acaba de crear.