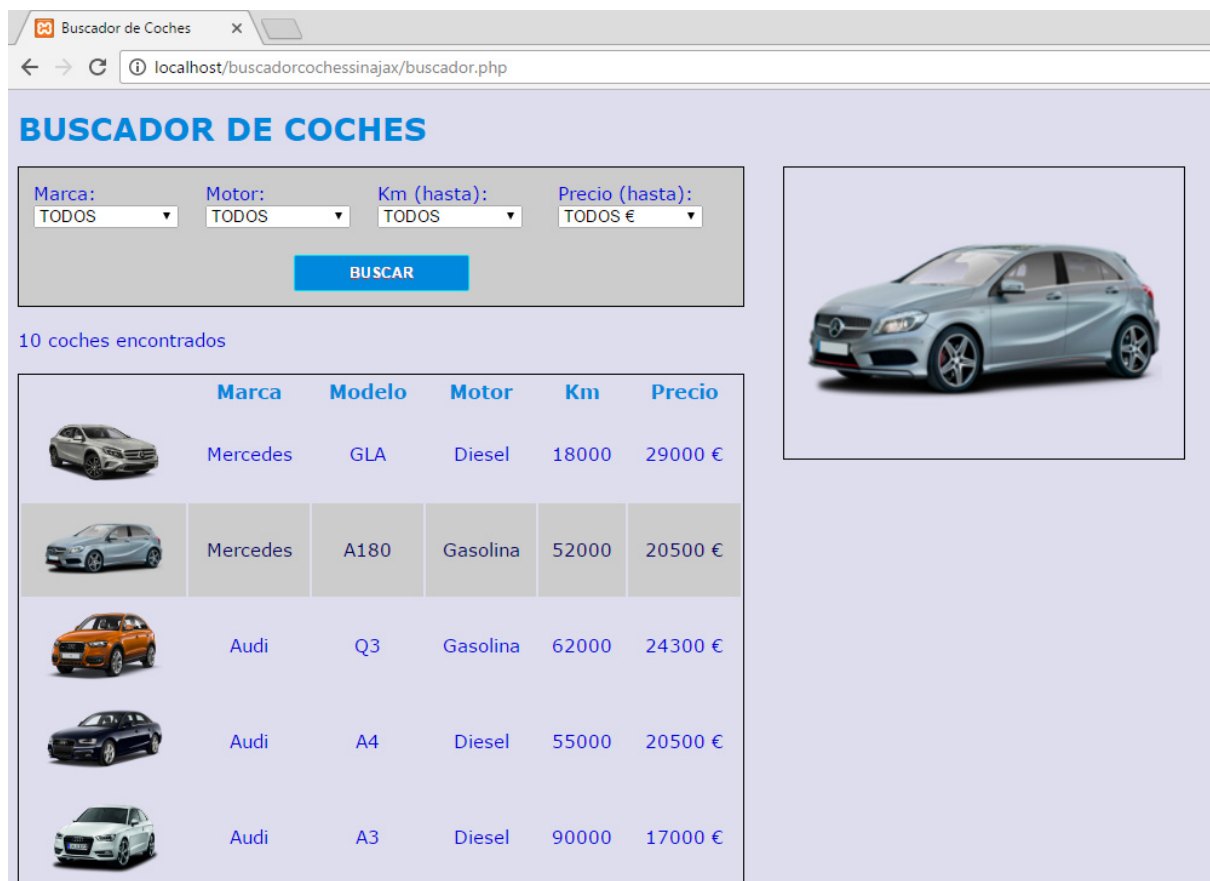





Desarrolla una aplicación web en PHP que implemente un **buscador para encontrar coches contenidos en un catálogo online**. La aplicación tomará los datos de un archivo XML llamado **coches.xml**. Este archivo define una gramática capaz de almacenar un conjunto de coches por medio de características que los describen:

```
<!DOCTYPE Coches [  
  <!ELEMENT Coches (Coche*)>  
  <!ELEMENT Coche (ID, Marca, Modelo, Motor, Año, Km, Precio, Descripcion, URLImagen)>  
  <!ELEMENT ID (#PCDATA)>  
  <!ELEMENT Marca (#PCDATA)>  
  <!ELEMENT Modelo (#PCDATA)>  
  <!ELEMENT Motor (#PCDATA)>  
  <!ELEMENT Año (#PCDATA)>  
  <!ELEMENT Km (#PCDATA)>  
  <!ELEMENT Precio (#PCDATA)>  
  <!ELEMENT Descripcion (#PCDATA)>  
  <!ELEMENT URLImagen (#PCDATA)>  
>]
```

El archivo que se suministra contiene los datos de 10 coches: 5 de gasolina y otros 5 diésel. Además, los 10 coches presentes son de 4 marcas distintas: 2 son Mercedes, 3 Audi, 3 BMW y 2 HONDA.

Se parte del archivo **buscador.php**; que se encarga de mostrar el formulario d el buscador (elemento `<form>` con `name="buscar"`). En esa misma página se muestra el listado de coches que hayan sido encontrados en el catálogo tras realizar la búsqueda, junto con la cantidad de ellos encontrados:



	Marca	Modelo	Motor	Km	Precio
	Mercedes	GLA	Diesel	18000	29000 €
	Mercedes	A180	Gasolina	52000	20500 €
	Audi	Q3	Gasolina	62000	24300 €
	Audi	A4	Diesel	55000	20500 €
	Audi	A3	Diesel	90000	17000 €

> El usuario puede buscar coches en el catálogo especificando en el **formulario** las siguientes características: la marca del coche, el tipo de motor (gasolina o diésel), los kilómetros máximos que tiene y el precio máximo. Si el usuario no desea especificar un valor concreto para alguna de estas características, puede elegir la opción TODOS que existe en la lista desplegable (elemento <select>) de cada característica.

Además, al cargar la página se rellenan las listas desplegables de cada característica con los valores posibles a elegir. Por ejemplo, la lista que permite elegir la marca del coche a buscar (elemento <select> con name="marca") se rellena con todas las marcas de coches presentes entre los coches que hay en el archivo coches.xml. Es decir, se rellena con las opciones: Audi, BMW, Mercedes y HONDA; además de la opción TODOS.

Para ello, el archivo buscador.php incluye al archivo **funciones.php**; que cuenta con la función :

```
listarValores ($arrayCoches, $atributo)
```

Esta función devuelve un array con los valores distintos de un atributo/característica de un coche que se encuentren en los coches del array de Coches XML pasado como parámetro. Esta función se usa para cargar los valores que muestra cada uno de los elementos <select> del buscador. Es decir, se usa para cargar todas las marcas de coches encontradas en el archivo XML, todos los tipos de motor presentes, los Km de cada coche que hay y todos los precios existentes. El parámetro atributo indica la propiedad del coche de la cual se desea obtener el listado. El array devuelto contiene el listado de valores ordenados de menor a mayor (o alfabéticamente) y sin que se repitan opciones.

> Una vez elegidas las características del coche a buscar, la aplicación envía el formulario hacia sí misma mediante POST; y devuelve un listado con los coches encontrados según el criterio de búsqueda indicado. Para realizar la **búsqueda**, el archivo funciones.php cuenta con la función:

```
buscarCoches ($arrayCoches, $marca, $motor, $km, $precio)
```

Esta función devuelve un array de objetos Coche XML con los coches que encuentre en arrayCoches y cuyas características coincidan con los valores indicados en los parámetros de búsqueda.

La función se ejecuta pasándole como argumentos los valores elegidos por el usuario en cada lista <select> de las características del formulario de búsqueda.

> El propio archivo buscador.php **muestra el listado** de coches encontrados en forma de tabla; cuyas columnas describen las características de cada coche. El archivo funciones.php cuenta con la función:

```
mostrarTablaCoches ($arrayCoches)
```

Esta función imprime una tabla HTML con los coches contenidos en el array de Coches XML pasado como parámetro. La tabla muestra un coche por fila; y cada coche se representa mediante 6 columnas: su imagen en miniatura, la marca, el modelo, el tipo de motor, su kilometraje y el precio. Además imprime la cantidad de coches representados en la tabla.

> Adicionalmente, si el usuario hace **click sobre** alguno de los **coches** mostrados en el listado, entonces se muestra en la sección de la derecha (elemento <div> con id="coche") un marco conteniendo la imagen de dicho coche ampliada. Para ello, en el evento OnClick de la fila de la tabla donde se encuentra el coche, se llama a la función JavaScript:

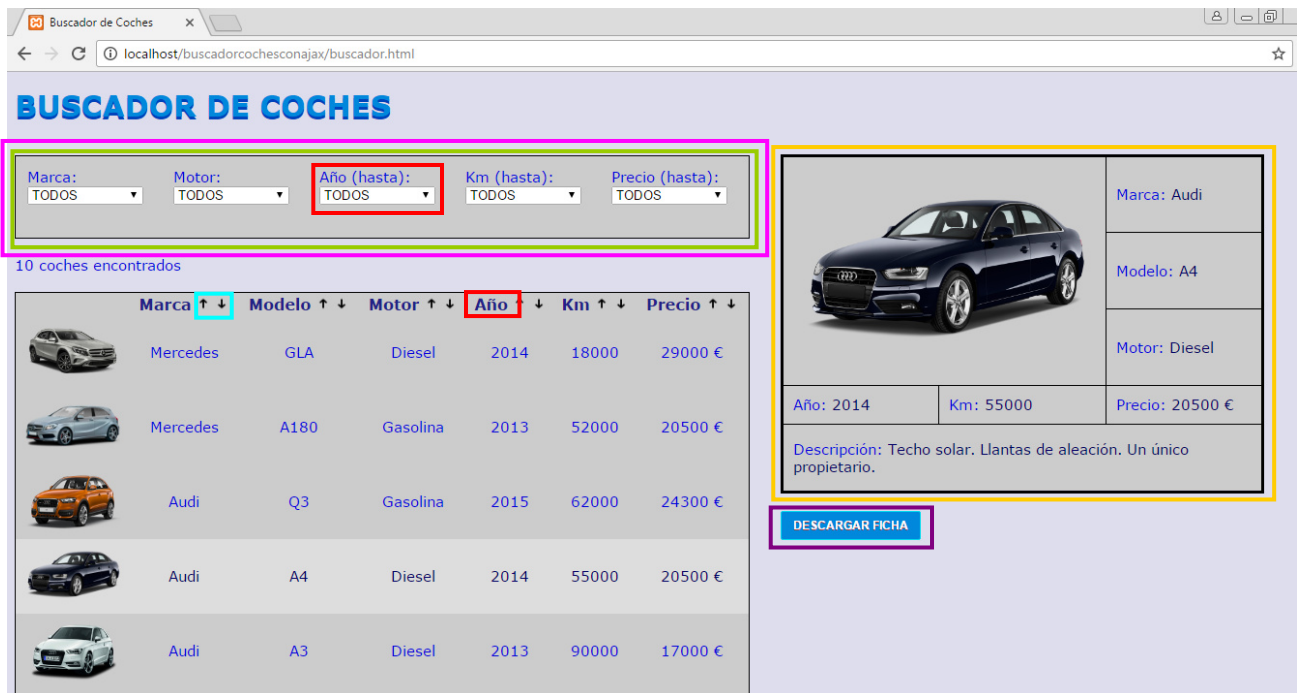
```
mostrarCoche (idImg)
```

Esta función recibe como parámetro el id del elemento con la imagen del coche que el usuario desea ampliar; y carga en el marco de la derecha la imagen de éste ampliada.

Esta versión de la aplicación requiere una actualización completa de la página web cada vez que el usuario cambie el valor de algún parámetro del formulario para volver a buscar coches.

Se pide modificar la aplicación web buscador.php para que ésta sea **más dinámica, interactiva y eficiente**; de forma que el usuario tenga la sensación de estar usando una aplicación web que responde como si fuera una aplicación de escritorio.

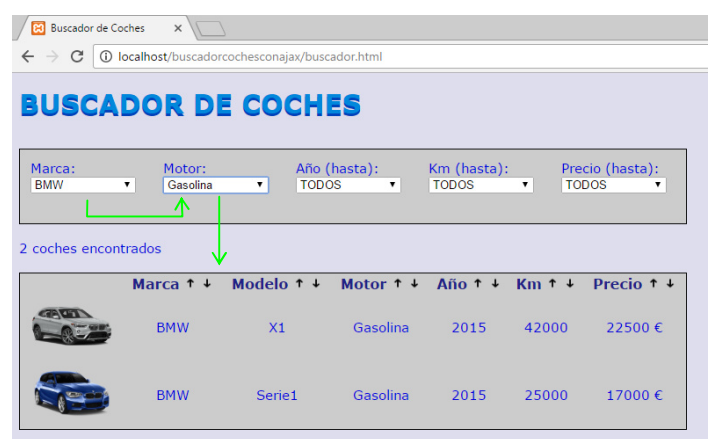
Para ello, se debe **implementar AJAX** para actualizar ciertas secciones de la web tras la interacción del usuario sin necesidad de recargar la página entera (con la consecuente mejora del ancho de banda).



Se deberá modificar la aplicación suministrada para:

1) Incluir la característica Año en el buscador para que el usuario pueda indicar en su búsqueda el año máximo de compra del coche a encontrar. El archivo XML de coches contiene esta característica registrada. La página deberá mostrar también el atributo Año en la tabla de coches encontrados.

2) Implementar mediante AJAX que las búsquedas se realicen sin tener que pulsar el botón BUSCAR. Se debe eliminar la etiqueta `<form>` del buscador y el botón BUSCAR. Se debe programar la aplicación para que cuando el usuario cambie algún valor de búsqueda (evento `OnChange` de cada `<select>`); se realice la búsqueda automáticamente tomando los valores de las características; y solo se recargue la tabla con los coches encontrados (crear un `<div>` para contener la tabla de resultados):



3) Usar AJAX para que las búsquedas sean más cómodas para el usuario; de forma que cada lista `<select>` se muestre con las opciones apropiadas según los valores de las características previas que el usuario haya elegido. Para ello, cada `<select>` debe estar en un `<div>` independiente.

Por ejemplo, supongamos que el usuario desea buscar un coche de la Marca Audi. En el archivo `coches.xml` hay presentes tres coches de esta marca: 1 de gasolina y 2 diésel. El objetivo es que cuando el usuario seleccione la Marca Audi, automáticamente se liste en el elemento `<select>` de la siguiente característica (Motor); todos los valores posibles de Motor encontrados en los coches Audi que hay en el archivo XML (Gasolina y Diesel). La lista se debe mostrar con los valores ordenados, sin repetidos e incluyendo el valor TODOS como opción seleccionada. Además se deberá lanzar automáticamente la búsqueda de coches con la Marca Audi y con el valor TODOS para el resto de características:



De igual modo, si el usuario quiere precisar más buscando un coche de la Marca Audi y con Motor Diesel; al elegir la opción diésel en el `<select>`, automáticamente se cargará en la lista de la siguiente característica (Año) todos los valores posibles de Año encontrados en los coches Audi Diesel que hay en el archivo XML. Análogamente, también se ejecutará de forma automática la búsqueda de coches con la Marca Audi, el Motor Diesel y con el valor TODOS para el resto de características:



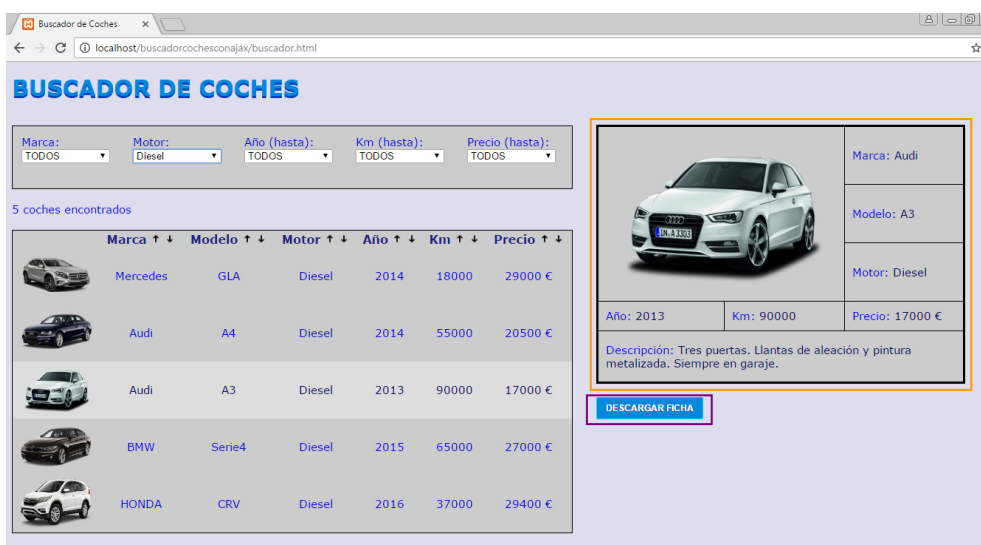
Este proceso de búsqueda guiada se realizará con el resto de características del coche (Km y Precio); de forma que se vayan filtrando los coches a medida que se completa los parámetros de la búsqueda. El proceso de filtrado se realizará por medio del evento `OnChange` de cada elemento `<select>`:



4) Añadir a cada columna de la tabla de coches dos flechas (↑ y ↓) a modo de botones. Se implementarán mediante elementos que respondan al evento OnClick. Al pulsarlas, permitirán mediante AJAX ordenar la tabla con los coches encontrados de manera ascendente o descendente según el valor de dicha característica:



5) Mostrar la ficha del coche seleccionado al pasar el ratón sobre su fila en la tabla de resultados (evento OnMouseOver de cada elemento <tr> del <table>). La ficha se mostrará en un elemento <div> creado en la parte derecha de la página (con posiciones absolutas). La ficha se representará usando una tabla que contenga todas las características del coche y su imagen ampliada. Debe quedar con una disposición de filas y columnas como la que se ilustra:



6) Añadir un botón para descargar en el cliente un archivo XML con los datos del coche que esté seleccionado en la tabla (una ficha XML del coche). Antes de descargar la ficha, se debe crear en el servidor un archivo XML para la ficha con los elementos de las características del coche elegido.

> Iniciar **descarga automática** de archivos en el cliente:

<https://parasitovirtual.wordpress.com/2010/06/15/curso-php-capitulo-10-subida-y-descarga-de-ficheros>

> Secciones <div>: http://www.w3schools.com/tags/tag_div.asp http://www.w3schools.com/css/css_positioning.asp

> Tablas <table>: http://www.w3schools.com/html/html_tables.asp

> Estilos CSS para elementos HTML: http://www.w3schools.com/html/html_css.asp

> Gestión de **eventos** HTML mediante código JavaScript: http://www.w3schools.com/js/js_events.asp

