

### Aclaración:

Las partes se corregirán de manera secuencial (ascendentemente). Si están bien todos los puntos de una parte, habilita la corrección de la parte posterior.

Se debe crear un archivo por cada entidad de TypeScript. Todos los métodos deben estar declarados dentro de clases.

Toda comunicación con el backend se realizará con AJAX. Todo el pasaje de datos se hará con JSON.

Las vistas (páginas .php o .html) se entregan por parte del profesor.

Se deben respetar los nombres de los archivos, de las clases, de los métodos y de los parámetros de las peticiones.

Todas las referencias a archivos y/o recursos, deben estar de manera relativa.

El backend lo debe proveer el alumno (respetando nombres y tipos de parámetros) y tendrá que interactuar con la base de datos **garage\_bd** (autos).

## Parte 1 FRONTEND – HTML5 y TypeScript (hasta 5)

Crear la siguiente clase en **Typescript** en el namespace **Apellido** (del alumno):

- **Auto**, posee como atributos protegidos:
  - patente(cadena)
  - marca(cadena)
  - color(cadena)
  - precio(flotante)

Un constructor (que inicialice los atributos), un método de instancia **toJSON()**, que retornará los datos de la instancia (en una cadena con formato **JSON**).

Crear en **TypeScript** la clase **Manejadora** (en el namespace **PrimerParcial**) que posea los siguientes métodos y funcionalidades:

**AgregarAutoJSON.** Obtiene la **patente**, la **marca**, el **color** y el **precio** desde la página **auto\_json.html** y enviará por POST (utilizando **AJAX**) hacia el servidor ubicado en: **“./backend/altaAutoJSON.php”**.

*altaAutoJSON.php:* Se recibe por POST la patente, la marca, el color y el precio. Invocar al método **guardarJSON** y pasarle **‘./archivos/autos.json’** como parámetro.

Informar por **consola** y **alert** el mensaje recibido.

**ListarAutosJSON.** Se invocará vía AJAX (por GET) a **“./backend/listadoAutosJSON.php”**.

*listadoAutosJSON.php:* (GET) Se mostrará el listado de todos los autos en formato JSON (traerJSON). Pasarle **‘./archivos/autos.json’** como parámetro.

Informar por **consola** el mensaje recibido y mostrar el listado en la página (**div id='divTabla'**).

**VerificarAutoJSON.** Obtiene la **patente** desde la página **auto\_json.html** y enviará por POST (utilizando AJAX) hacia el servidor ubicado en **“./backend/verificarAutoJSON.php”**.

Informar por **consola** y **alert** el mensaje recibido.

# Apellido y nombre del alumno

## Autos - JSON

Patente	<input type="text"/>
Marca	<input type="text"/>
Color	<input type="color" value="red"/>
Precio	<input type="text" value="100000"/>
<div><button>AgregarJSON</button><button>MostrarJSON</button><button>VerificarJSON</button></div>	

tabla aquí...

**NOTA:** vincular los eventos 'click' de cada botón a la función correspondiente.

Agregar la siguiente clase en **Typescript**, en el namespace **Apellido** (del alumno):

- **AutoBD**, hereda de **Auto**, posee como único atributo **foto** (cadena). Un constructor para inicializar los atributos.

En la clase **ManejadoraAutoBD** (en el namespace **PrimerParcial**), agregar los siguientes métodos:

AgregarAutoBD y ListarAutosBD.

**AgregarAutoBD.** Obtiene la **patente**, la **marca**, el **color** y el **precio** desde la página **auto\_bd.html** y enviará por POST (utilizando **AJAX**) hacia el servidor ubicado en: **“./backend/agregarAutoSinFoto.php”**.

*agregarAutoSinFoto.php:* Se recibe por POST el parámetro **auto\_json** (patente, marca, color y precio), en formato de cadena JSON. Se invocará al método **agregar**.

Se retornará un JSON que contendrá: **éxito(bool)** y **mensaje(string)** indicando lo acontecido.

Informar por **consola** y **alert** el mensaje recibido.

**ListarAutosBD.** Recuperará (por **AJAX**) todos los autos de la base de datos, invocando por GET al servidor ubicado en: **“./backend/listadoAutosBD.php”**.

*listadoAutosBD.php:* (GET) Se mostrará el listado completo de los autos (obtenidos de la base de datos) en una tabla (HTML con cabecera). Invocar al método **traer**.

*Nota:* Si se recibe el parámetro **tabla** con el valor **mostrar**, retornará los datos en una tabla (HTML con cabecera), preparar la tabla para que muestre la imagen, si es que la tiene.

Si el parámetro no es pasado o no contiene el valor 'mostrar', retornará el array de objetos con formato JSON.

Con el array recibido, crear una tabla HTML con cabecera (en el **FRONTEND**) que mostrará toda la información de cada uno de los autos.

Informar por **consola** el mensaje recibido y mostrar el listado en la página (**div id='divTabla'**).

**NOTA:**

La columna que indica el color del auto debe contener un input (type="color") y estar deshabilitado (disabled).

## Apellido y nombre del alumno

### Autos BD

Patente	<input type="text"/>
Marca	<input type="text"/>
Color	<input type="color" value="red"/>
Precio	<input type="text" value="100000"/>

AgregarBDMostrarBDModificarBD

tabla aquí...

**NOTA:** vincular los eventos 'click' de cada botón a la función correspondiente.

## Parte 2 FRONTEND – HTML5 y TypeScript (hasta 6)

Del listado del punto anterior, *agregar una columna (Acciones) al listado de autos que permita: **Eliminar** y **Modificar** al auto elegido. Para ello, agregue dos botones (input [type=button]) que invoquen a las funciones (de la clase Manejadora), EliminarAutoBD y ModificarAutoBD, respectivamente.*

Patente	Marca	Precio	Color	Acciones	
abc123	ford	150999	<input type="color" value="red"/>	Eliminar	Modificar
abc321	renault	150999	<input type="color" value="blue"/>	Eliminar	Modificar
cba123	fiat	250000	<input type="color" value="yellow"/>	Eliminar	Modificar
cba321	porche	650000	<input type="color" value="green"/>	Eliminar	Modificar

**ModificarAuto.** Mostrará todos los datos del auto que recibe por parámetro (objeto **JSON**), en el formulario de la página **auto\_bd.html**. Permitirá modificar cualquier campo, a excepción de la **patente**, dejarlo como de sólo lectura.

Al pulsar el botón “*ModificarBD*”, se invocará por POST (utilizando **AJAX**) al servidor ubicado en:

**“./backend/modificarAutoBD.php”**. Realizar una función para tal fin.

*modificarAutoBD.php: Se recibirán por POST los siguientes valores: auto\_json (patente, marca, color y precio, en formato de cadena JSON) para modificar un auto en la base de datos. Invocar al método modificar.*

*Se retornará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.*

Refrescar el listado sólo si se pudo modificar, caso contrario, informar (por alert y consola) de lo acontecido.

**EliminarAutoBD.** Recibe como parámetro al objeto **JSON** que se ha de eliminar. Pedir confirmación, mostrando la **patente y marca**, antes de eliminar.

Si se confirma se invocará por POST (utilizando **AJAX**) al servidor ubicado en **“./backend/eliminarAutoBD.php”**.

*eliminarAutoBD.php: Recibe el parámetro auto\_json (patente, marca, color y precio, en formato de cadena JSON) por POST y se deberá borrar el auto (invocando al método eliminar).*

*Si se pudo borrar en la base de datos, invocar al método guardarJSON y pasarle cómo parámetro el valor './archivos/autos\_eliminados.json'.*

*Retornar un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.*

Informar por consola y alert lo acontecido. Refrescar el listado para visualizar los cambios.

## Parte 3 FRONTEND – HTML5 y TypeScript (hasta 8)

En la clase **ManejadoraAutoFotos** (en el namespace **PrimerParcial**), agregar los siguientes métodos:

AgregarAutoFotoBD, VerificarAutoFotoBD, MostrarAutoFotosBD, EliminarAutoFotoBD y ModificarAutoFotoBD.

Al cargar la página **auto\_foto\_bd.html**, se deberá cargar el listado de autos con sus respectivas fotos obtenidos desde la base de datos, para ello se invocará al método **listadoAutosBD.php** que enviará por GET (utilizando **AJAX**) hacia el servidor ubicado en: **“./backend/listadoAutosBD.php”**.

Con el array de AutoFoto recibido, armar una tabla (HTML) para mostrar los datos.

Preparar la tabla (HTML) con una columna extra para que muestre la imagen de la foto (50px por 50px).

Mostrar el listado en la página (*div id='divTablaAutoFotos'*).

### NOTA:

*La columna que indica el color del auto debe contener un input (type="color") y estar deshabilitado (disabled).*

*Agregar una columna (Acciones) al listado de autos que permita: **Eliminar** y **Modificar** al auto con foto elegido. Para ello, agregue dos botones (input [type=button]) que invoquen a las funciones EliminarAutoFotoBD y Modificar, respectivamente.*

Patente	Marca	Precio	Color	Foto	Acciones	
abc123	ford	150999			Eliminar	Modificar
abc321	renault	150999			Eliminar	Modificar
cba123	fiat	250000			Eliminar	Modificar
cba321	porche	650000			Eliminar	Modificar

**AgregarAutoFotoBD.** Obtiene los datos del auto (incluyendo la foto) desde la página *auto\_foto\_bd.html* y se enviará por POST (utilizando **AJAX**) hacia *“./backend/agregarAutoBD.php”*.

*agregarAutoBD.php:* Se recibirán por POST los valores: patente, marca, color, precio y la foto para registrar un auto en la base de datos.

Verificar la previa existencia del auto invocando al método *existe*. Se le pasará como parámetro el array que retorna el método *traer*.

Si el auto ya existe en la base de datos, se retornará un mensaje que indique lo acontecido.

Si el auto no existe, se invocará al método *agregar*. La imagen se guardará en *“./autos/imagenes/”*, con el nombre formado por la patente punto hora, minutos y segundos del alta (Ejemplo: *AYF714.105905.jpg*).

Se retornará un JSON que contendrá: *éxito(bool)* y *mensaje(string)* indicando lo acontecido.

Informar por **consola** y **alert** el mensaje recibido.

Refrescar el listado de los autos con fotos.

**Modificar.** Mostrará todos los datos del auto que recibe por parámetro (objeto **JSON**), en el formulario, incluida la foto (mostrarla en *“imgFoto”*).

Permitirá modificar cualquier campo, a excepción de la **patente**, dejarlo como de sólo lectura.

Al pulsar el botón *“ModificarBD”*, invocará al método **ModificarAutoBD**, que llamará por POST (utilizando **AJAX**) hacia *“./backend/modificarAutoBDFoto.php”*.

*modificarAutoBDFoto.php:* Se recibirán por POST los siguientes valores: *auto\_json* (patente, marca, color y precio, en formato de cadena JSON) y la foto (para modificar un auto en la base de datos). Invocar al método *modificar*.

Si se pudo modificar en la base de datos, la foto original del registro modificado se moverá al subdirectorío

*“./autosModificados/”*, con el nombre formado por la patente punto 'modificado' punto hora, minutos y segundos de la modificación (Ejemplo: *AYF714.renault.modificado.105905.jpg*).

Se retornará un JSON que contendrá: *éxito(bool)* y *mensaje(string)* indicando lo acontecido.

Si se invoca por GET (sin parámetros), se mostrarán en una tabla (HTML) la información de todos los autos modificados y sus respectivas imágenes.

Refrescar el listado sólo si se pudo modificar, caso contrario, informar (por alert y consola) de lo acontecido.

Patente	<input type="text" value="abc321"/>
Marca	<input type="text" value="renault"/>
Color	<input type="color" value="#4169E1"/>
Precio	<input type="text" value="150999"/>
Foto	<input type="button" value="Seleccionar archivo"/> Sin archivos seleccionados



**EliminarAutoFotoBD.** Recibe como parámetro al objeto **JSON** que se ha de eliminar. Pedir confirmación, mostrando la **patente** y **marca**, antes de eliminar.

Si se confirma se invocará por POST (utilizando **AJAX**) hacia **“./backend/eliminarAutoBDFoto.php”**.

*eliminarAutoBDFoto.php: Se recibe el parámetro auto\_json (patente, marca, color, precio y pathFoto en formato de cadena JSON) por POST. Se deberá borrar el auto (invocando al método eliminar).*

*Si se pudo borrar en la base de datos, invocar al método guardarEnArchivo.*

*Retornar un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.*

*Si se invoca por GET (sin parámetros), se mostrarán en una tabla (HTML) la información de todos los autos borrados y sus respectivas imágenes.*

Informar por consola y alert lo acontecido. Refrescar el listado para visualizar los cambios.

## Parte 4 FRONTEND – HTML5 y TypeScript (hasta 10)

Al pulsar el botón **Listado PDF**, se invocará por GET (utilizando AJAX) hacia **“./backend/listadoAutosPDF.php”**.

*listadoAutosPDF.php: (GET) Generar un listado de los autos de la base de datos y mostrarlo con las siguientes características:*

- *Encabezado (apellido y nombre del alumno a la izquierda y número de página a la derecha).*
- *Cuerpo (Título del listado, listado completo de los autos con su respectiva foto).*
- *Pie de página (fecha actual, centrada).*