

# PRIMER PARCIAL – LABORATORIO III – 2 cuat. 2022

## 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 la comunicación con el backend, se realizará con AJAX. El pasaje de datos, principalmente con JSON.

El backend lo debe proveer el alumno (respetando nombres y tipos de parámetros).

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

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

## Parte 1 (hasta un 5)

Crear una aplicación Web que permita realizar un ABM de distintos neumáticos.

Crear la siguiente jerarquía de clases en **Typescript** (en el namespace **Entidades**):

- a. **Neumatico**: marca(cadena), medidas(cadena) y precio (numérico) como atributos. Un constructor que reciba tres parámetros.

Un método, *ToString()*, que retorne la representación de la clase en formato **cadena** (preparar la cadena para que, al juntarse con el método *ToJSON*, forme una cadena JSON válida).

Un método de instancia, *ToJSON()*, que retorne la representación de la instancia en formato de cadena JSON válido. Reutilizar código.

- b. **NeumaticoBD**, hereda de **Neumatico**, posee como atributos *id* (numérico) y *pathFoto*(cadena). Un constructor para inicializar los atributos (con todos sus parámetros opcionales). Un método *ToJSON()*, que retornará la representación del objeto en formato JSON. Reutilizar código.

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

**AgregarNeumaticoJSON**. Obtiene la marca, las medidas y el precio desde la página **neumatico.html** y se enviará (por **AJAX**) hacia **“./BACKEND/altaNeumaticoJSON.php”** que invoca al método *guardarJSON* y se pasa **‘./archivos/neumaticos.json’** como parámetro. Retornará un JSON que contendrá: *éxito*(bool) y *mensaje*(string) indicando lo acontecido.

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

**MostrarNeumaticosJSON**. Recuperará (por **AJAX**) todos los neumáticos del archivo **neumaticos.json** y generará un listado dinámico, crear una tabla HTML con cabecera (en el **FRONTEND**) que mostrará toda la información de cada uno de los neumáticos. Invocar a **“./BACKEND/listadoNeumaticosJSON.php”**, recibe la petición (por **GET**) y retornará el listado de todos los neumáticos en formato JSON.

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

**VerificarNeumaticoJSON.** Se invocará (por **AJAX**) a **“./BACKEND/verificarNeumaticoJSON.php”**. Se recibe por **POST** la **marca** y las **medidas** y retornará un **JSON** que contendrá: **éxito(bool)** y **mensaje(string)** indicando lo acontecido (agregar, aquí también, el mensaje obtenido del método **VerificarNeumaticoJSON**).  
Se mostrará (por **consola** y **alert**) lo acontecido.

**AgregarNeumaticoSinFoto.** Obtiene la marca, las medidas y el precio desde la página **neumatico.html**, y se enviará (por **AJAX**) hacia **“./BACKEND/agregarNeumaticoSinFoto.php”** que recibe por **POST** el parámetro **neumatico\_json** (marca, medidas 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.

**MostrarNeumaticosBD.** Recuperará (por **AJAX**) todos los neumáticos de la base de datos, invocando a **“./BACKEND/listadoNeumaticosBD.php”**, que recibirá el parámetro **tabla** con valor distinto a 'mostrar', para que retorne un array de objetos con formato **JSON**.  
Crear una tabla **HTML** con cabecera (en el **FRONTEND**) para mostrar la información de cada uno de los neumáticos. Preparar la tabla para que muestre la imagen, si es que la tiene. Todas las imágenes deben tener 50px por 50px de dimensiones.  
Informar por **consola** el mensaje recibido y mostrar el listado en la página (**div id='divTabla'**).

## Parte 2 (hasta un 6)

En la clase **Manejadora**, agregar la interface **Iparte2** con los métodos **EliminarNeumatico** y **ModificarNeumatico**.

**EliminarNeumatico.** Recibe como parámetro al objeto **JSON** que se ha de eliminar. Pedir confirmación, mostrando la marca y las medidas, antes de eliminar.  
Si se confirma se invocará (por **AJAX**) a **“./BACKEND/eliminarNeumaticoBD.php”** pasándole como parámetro **neumatico\_json** (**id**, **marca**, **medidas** y **precio**, en formato de cadena **JSON**) por **POST** y se deberá borrar el neumático de la base de datos (invocando al método **eliminar**).  
Si se pudo borrar en la base de datos, invocar al método **guardarJSON** y pasarle **“./BACKEND/archivos/neumaticos\_eliminados.json”** como parámetro.  
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.

**ModificarNeumatico.** Mostrará todos los datos del neumático que recibe por parámetro (objeto **JSON**), en el formulario, de tener foto, incluirla en **“imgFoto”**. Permitirá modificar cualquier campo, a excepción del **id**.  
Al pulsar el botón **Modificar sin foto** (de la página) se invocará (por **AJAX**) a **“./BACKEND/modificarNeumaticoBD.php”** Se recibirán por **POST** los siguientes valores: **neumatico\_json** (**id**, **marca**, **medidas**, y **precio**, en formato de cadena **JSON**) para modificar un neumático en la base de datos. Invocar al método **modificar**.  
**Nota:** El valor del **id**, será el **id** del neumático 'original', mientras que el resto de los valores serán los del neumático a ser modificado.  
Se retornará un **JSON** que contendrá: **éxito(bool)** y **mensaje(string)** indicando lo acontecido.  
Refrescar el listado solo si se pudo modificar, caso contrario, informar (por **alert** y **consola**) de lo acontecido.

**NOTA:** Agregar una columna extra (Acciones) al listado de neumáticos que permita: **Eliminar** y **Modificar** al neumático elegido. Para ello, agregue dos botones (input [type=button]) que invoquen a las funciones *EliminarNeumatico* y *ModificarNeumatico*, respectivamente.

## Parte 3 (hasta un 8)

En la clase Manejadora, agregar la interface **Iparte3**, con los siguientes métodos:

**VerificarNeumaticoBD.** Se recupera la marca y las medidas del neumático desde la página *neumatico\_BD.html* y se invoca (por **AJAX**) a **“./BACKEND/verificarNeumaticoBD.php”** que recibe por POST el parámetro *obj\_neumatico*, que será una cadena **JSON** (marca y medidas), si coincide con algún registro de la base de datos (invocar al método *traer*) retornará los datos del objeto (invocar al *toJSON*). Caso contrario, un JSON vacío (*{}*).

Informar por **consola** lo acontecido y mostrar el objeto recibido en la página (*div id='divInfo'*).

**AgregarNeumaticoFoto.** Obtiene la marca, las medidas, el precio y la **foto** desde la página *neumatico\_BD.html* y se enviará (por **AJAX**) hacia **“./BACKEND/agregarNeumaticoBD.php”** que recibirá por POST, la marca, las medidas, el precio y la foto para registrar un neumático en la base de datos.

Verificar la previa existencia del neumático invocando al método **existe**.

Se le pasará como parámetro el array que retorna el método *traer*. Si el neumático ya existe en la base de datos, se retornará un mensaje que indique lo acontecido. Si el neumático no existe, se invocará al método *agregar*. La imagen se guardará en *“./neumaticos/imagenes/”*, con el nombre formado por el marca punto hora, minutos y segundos del alta (Ejemplo: *pirelli.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 neumáticos reutilizando el método *MostrarNeumaticosBD*.

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

Si se confirma se invocará (por **AJAX**) a **“./BACKEND/eliminarNeumaticoBDFoto.php”** que recibe el parámetro *neumatico\_json* (*id, marca, medidas, precio y pathFoto* en formato de cadena JSON) por POST. Se deberá borrar el neumático (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.

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

**ModificarNeumaticoBDFoto.** Mostrará todos los datos del neumático que recibe por parámetro (objeto **JSON**), en el formulario, de tener foto, incluirla en **“imgFoto”**. Permitirá modificar cualquier campo (incluyendo la foto), a excepción del id.

Al pulsar el botón **Modificar** (de la página) se invocará (por **AJAX**) a

**“./BACKEND/modificarNeumaticoBDFoto.php”** dónde se recibirán por POST los siguientes valores: *neumatico\_json* (*id, marca, medidas y precio*, en formato de cadena JSON) y la foto (para modificar un neumático en la base de datos). Invocar al método *modificar*.

**Nota:** El valor del id, será el id del neumático 'original', mientras que el resto de los valores serán los del neumático a ser modificado.

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

*“./neumaticosModificados/”*, con el nombre formado por el id punto marca punto 'modificado' punto hora, minutos y segundos de la

modificación (Ejemplo: 987.fateo.modificado.105905.jpg). Se retornará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

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

**NOTA:** Adecuar el método *MostrarNeumaticosBD* para que, de acuerdo desde que página se invoque, cambie el manejador de eventos de los botones que permiten **Eliminar** y **Modificar** al neumático elegido. Para ello, agregue la lógica necesaria para que, desde **neumatico\_BD.html**, invoquen a las funciones *EliminarNeumaticoFoto* y *ModificarNeumaticoFoto*, respectivamente. Mantener en funcionamiento los manejadores de la página *neumatico.html*.

## Parte 4

En la clase Manejadora, agregar la interface **Iparte4** con los métodos:

**MostrarBorradosJSON:** Invocará (por **AJAX**) a **“./BACKEND/mostrarBorradosJSON.php”** que muestra todo lo registrado en el archivo **“neumaticos\_eliminados.json”**. Para ello, agregar un método estático (en *NeumaticoBD*), llamado **mostrarBorradosJSON**.

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

**MostrarFotosModificados:** Invocará (por **AJAX**) a **“./BACKEND/mostrarFotosDeModificados.php”** que muestra (en una tabla HTML) todas las imagenes (50px X 50px) de los neumáticos registrados en el directorio **“./BACKEND/neumaticosModificados/”**. Para ello, agregar un método estático (en *NeumaticoBD*), llamado **mostrarModificados**.  
Mostrar el listado en la página (*div id='divTabla'*).

Generar un retardo en cada petición al BACKEND y mostrar el .gif en cada espera.