PRIMER PARCIAL – LABORATORIO III – 1 cuat. 2021

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 productos y productos envasados.

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

a. ***Producto***: nombre(cadena) y origen(cadena) como atributos. Un constructor que reciba dos 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. ***ProductoEnvasado***, hereda de Producto, posee como atributos id(numérico), codigoBarra(cadena), precio(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:

AgregarProductoJSON. Obtiene el nombre y el origen desde la página ***producto.html*** y se enviará (por ***AJAX***) hacia ***"./BACKEND/AltaProductoJSON.php"*** *que invoca al método GuardarJSON pasándole* ***'./BACKEND/archivos/productos.json'*** *cómo parámetro para que agregue al producto en el archivo. Retornará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.*
Informar por **consola** y **alert** el mensaje recibido.

MostrarProductosJSON. Recuperará (por ***AJAX***) todos los productos del archivo ***productos.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 productos. Invocar a ***"./BACKEND/ListadoProductosJSON.php"***, *recibe la petición (por GET) y retornará el listado de todos los productos en formato JSON.*
Informar por **consola** el mensaje recibido y mostrar el listado en la página (*div id='divTabla'*).

VerificarProductoJSON. Se invocará (por ***AJAX***) a ***"./BACKEND/VerificarProductoJSON.php"***. *Se recibe por POST el* ***nombre*** *y el* ***origen****, si coinciden con algún registro del archivo JSON (VerificarProductoJSON), crear una* ***COOKIE*** *nombrada con el nombre y el origen del producto, separado con un guión bajo (****limon_tucuman****) que guardará la fecha actual (con horas, minutos y segundos) más el retorno del mensaje del método estático VerificarProductoJSON de la clase Producto.*

*Retornar un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido (agregar, aquí también, el mensaje obtenido del método VerificarProductoJSON).*
Se mostrará (por **consola** y **alert**) lo acontecido.

MostrarInfoCookie. Se realizará una petición (por AJAX) a ***"./BACKEND/MostrarCookie.php"*** *que recibe por GET el **nombre** y el **origen** del producto y se verificará si existe una cookie con el mismo nombre, de ser así, retornará un **JSON** que contendrá: éxito(bool) y mensaje(string), dónde se mostrará el contenido de la cookie. Caso contrario, false y el mensaje indicando lo acontecido.*
Informar por **consola** el mensaje recibido y mostrar el mensaje en la página (*div id='divInfo'*).

AgregarProductoSinFoto. Obtiene el código de barra, el nombre, el origen y el precio desde la página ***producto.html***, y se enviará (por ***AJAX***) hacia ***"./BACKEND/AgregarProductoSinFoto.php"*** *que recibe por POST el parámetro **producto_json** (codigoBarra, nombre, origen 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.

MostrarProductosEnvasados. Recuperará (por ***AJAX***) todas los productos envasados de la base de datos, invocando a ***"./BACKEND/ListadoProductosEnvasados.php"***, *que recibirá el parámetro **tabla** con el valor 'json', 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 productos envasados. Preparar la tabla para que muestre la imagen, si es que la tiene. Todas las imagenes 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 *EliminarProducto* y *ModificarProducto*.

EliminarProducto. Recibe como parámetro al objeto ***JSON*** que se ha de eliminar. Pedir confirmación, mostrando nombre y origen, antes de eliminar.
Si se confirma se invocará (por ***AJAX***) a ***"./BACKEND/EliminarProductoEnvasado.php"*** *pasándole cómo parámetro **producto_json** (id, nombre y origen, en formato de cadena JSON) por POST y se deberá borrar el producto envasado (invocando al método Eliminar).*
*Si se pudo borrar en la base de datos, invocar al método GuardarJSON y pasarle **'./BACKEND/archivos/productos_eliminados.json'** cómo 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.

ModificarProducto. Mostrará todos los datos del producto 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/ModificarProductoEnvadado.php"*** *Se recibirán por POST los siguientes valores: **producto_json** (id, codigoBarra, nombre, origen y precio, en formato de cadena JSON) para modificar un producto envasado en la base de datos. Invocar al método Modificar.*
***Nota:** El valor del id, será el id del producto envasado 'original', mientras que el resto de los valores serán los del producto envasado 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 productos envasados que permita:* ***Eliminar*** *y* ***Modificar*** *al producto elegido. Para ello, agregue dos botones (input [type=button]) que invoquen a las funciones EliminarProducto y ModificarProducto, respectivamente.*

Parte 3 (hasta un 8)

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

VerificarProductoEnvasado. Se recupera el nombre y el origen del producto envasado desde la página ***producto_envasado.html*** y se invoca (por ***AJAX***) a ***"./BACKEND/VerificarProductoEnvadado.php"*** que recibe por POST el parámetro ***obj_producto***, que será una cadena **JSON** (nombre y origen), 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'*).

AgregarProductoFoto. Obtiene el código de barra, el nombre, el origen, el precio y la ***foto*** desde la página ***producto_envasado.html*** y se enviará (por ***AJAX***) hacia ***"./BACKEND/AgregarProductoEnvasado.php"*** *que recibirá por POST todos los valores:* ***codigoBarra, nombre, origen, precio*** *y la* ***foto*** *para registrar un producto envasado en la base de datos.*
Verificar la previa existencia del producto envasado invocando al método Existe. Se le pasará como parámetro el array que retorna el método Traer.
Si el producto envasado ya existe en la base de datos, se retornará un mensaje que indique lo acontecido.
Si el producto envasado no existe, se invocará al método Agregar. La imagen se guardará en "./BACKEND/productos/imagenes/", con el nombre formado por el ***nombre*** *punto* ***origen*** *punto hora, minutos y segundos del alta (****Ejemplo: tomate.argentina.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 productos reutilizando el método *MostrarProductosEnvasados*.

BorrarProductoFoto. Recibe como parámetro al objeto ***JSON*** que se ha de eliminar. Pedir confirmación, mostrando nombre y código de barra, antes de eliminar.
Si se confirma se invocará (por ***AJAX***) a ***"./BACKEND/BorrarProductoEnvasado.php"*** *que recibe el parámetro* ***producto_json*** *(id, codigoBarra, nombre, origen, precio y pathFoto en formato de cadena JSON) por POST. Se deberá borrar el producto envasado (invocando al método Eliminar).*
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.

ModificarProductoFoto. Mostrará todos los datos del producto 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/ModificarProductoEnvadadoFoto.php"*** *dónde se recibirán por POST los siguientes valores:* ***producto_json*** *(id, codigoBarra, nombre, origen y precio, en formato de cadena JSON) y la* ***foto*** *(para modificar un producto envasado en la base de datos. Invocar al método Modificar.*
Nota: *El valor del id, será el id del producto envasado 'original', mientras que el resto de los valores serán los del producto envasado a ser modificado.*
Si se pudo modificar en la base de datos, la foto original del registro modificado se moverá al subdirectorio "./BACKEND/productosModificados/", con el nombre formado por el ***nombre*** *punto* ***origen*** *punto '****modificado****' punto hora, minutos y segundos de la modificación (****Ejemplo: aceite.italia.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 MostrarProductosEnvasados para que, de acuerdo desde que página se invoque, cambie el manejador de eventos de los botones que permiten* ***Eliminar*** *y* ***Modificar*** *al producto elegido. Para ello, agregue la lógica necesaria para que, desde* ***producto_envasado.html****, invoquen a las funciones EliminarProductoFoto y ModificarProductoFoto, respectivamente. Mantener en funcionamiento los manejadores de la página producto.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 "productos_eliminados.json". Para ello, agregar un método estático (en ProductoEnvasado), 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 productos envasados registrados en el directorio "./BACKEND/productosModificados/". Para ello, agregar un método estático (en ProductoEnvasado), llamado **MostrarModificados**.
Mostrar el listado en la página (*div id='divTabla'*).

FiltrarListado: Invocará (por ***AJAX***) a ***"./BACKEND/FiltrarProductos.php"*** que recibe por POST el ***origen***, se mostrarán en una tabla (HTML) los productos envasados cuyo origen coincidan con el pasado por parámetro.
Si se recibe por POST el ***nombre***, se mostrarán en una tabla (HTML) los productos envasados cuyo nombre coincida con el pasado por parámetro.
Si se recibe por POST el ***nombre*** y el ***origen***, se mostrarán en una tabla (HTML) los productos envasados cuyo nombre y origen coincidan con los pasados por parámetro.
Mostrar el listado en la página (*div id='divTabla'*).