

# REC - PRIMER PARCIAL – LABORATORIO III – 1 cuat. 2023

## 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 pueden usar templates o reutilizar código hecho, pero en ningún caso, se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.

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) deben respetar fielmente las formas, colores, iconos, etc.

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.

La Api Rest está creada en Node.js. El script del servidor Node.js se entrega por parte del profesor, usted deberá transpilar, incluir los módulos de Node.js y agregar la estructura de directorios adecuada y NO modificarla.

La Api Rest tendrá que interactuar con la base de datos **vivero\_bd** (plantas).

La Api Rest deberá estar separada del proyecto del recuperatorio (<http://localhost:2023>).

La Api Rest NO se deberá entregar, sólo el proyecto del recuperatorio.

## Parte 1 FRONTEND – HTML5 y TypeScript (hasta 7)

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

- **Planta**, posee como atributos protegidos:
  - `codigo(cadena)`
  - `nombre(cadena)`
  - `color_flor(cadena)`
  - `precio(numérico)`
  - `foto(cadena)`

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 **RecPrimerParcial**) que posea los siguientes métodos y funcionalidades:

MostrarPlantasFotosBD, AgregarPlantaFotoBD, EliminarPlantaFotoBD, MostrarInfoDePlantaFoto y ModificarPlantaFotoBD.


Al cargar la página **vivero.html**, se deberá cargar el listado de plantas con sus respectivas fotos obtenidas desde la base de datos, para ello se invocará al método **MostrarPlantasFotosBD** que enviará por GET (utilizando **AJAX**) hacia el servidor ubicado en: **"http://localhost:2023/listarPlantasFotosBD"**.

Con el array de tipo Planta recibido, armar una tabla (HTML en el **FRONTEND**) 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='divTablaPlantaFotos'`).

**NOTA:**

La columna que indica el color de la flor de la planta debe contener un input (type="color") y estar deshabilitado (disabled).

Agregar una columna (Acciones) al listado de plantas que permita: **Eliminar** y **Modificar** la planta elegida. Para ello, agregue dos botones (input [type=button]) que invoquen a las funciones EliminarPlantaFotoBD y MostrarInfoDePlantaFoto, respectivamente.

| Código  | Nombre    | Precio | ColorFlor                             | Foto  | Acciones |           |
|---------|-----------|--------|---------------------------------------|---|----------|-----------|
| mar_002 | margarita | 3520   | <input type="color" value="#ff9900"/> |  | Eliminar | Modificar |
| pot_001 | potus     | 1500   | <input type="color" value="#008000"/> |  | Eliminar | Modificar |

**AgregarPlantaFotoBD.** Obtiene el **código**, el **nombre**, el **color** y el **precio** desde la página **vivero.html** y enviará por POST (utilizando **AJAX**) hacia el servidor ubicado en: **"http://localhost:2023/agregarPlantaFotoBD"**.

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

Refrescar el listado de las plantas con fotos.

|           |   |
|-----------|---|
| Código    | <input type="text" value="hel_912"/>  |
| Nombre    | <input type="text" value="helecho"/>  |
| ColorFlor | <input type="color" value="#008000"/>   |
| Precio    | <input type="text" value="510"/>  |
| Foto      | <input type="button" value="Seleccionar archivo"/> Sin archivos seleccionados |



**MostrarInfoDePlantaFoto.** Mostrará todos los datos de la planta que recibe por parámetro (objeto **JSON**), en el formulario, incluida la foto (mostrarla en **"imgFoto"**).

Permitirá modificar cualquier campo, a excepción del **código**, dejarlo como de sólo lectura.

Al pulsar el botón **"Modificar"**, invocará al método **ModificarPlantaFotoBD**, que llamará por POST (utilizando **AJAX**) hacia **"http://localhost:2023/modificarPlantaFotoBD"**.

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

**EliminarPlantaFotoBD.** Recibe como parámetro al objeto **JSON** que se ha de eliminar. Pedir confirmación, mostrando el **código** y **nombre**, antes de eliminar.

Si se confirma se invocará por POST (utilizando **AJAX**) hacia **"http://localhost:2023/eliminarPlantaFotoBD"**.

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

## Parte 2 FRONTEND – HTML5 y TypeScript (hasta 8)

Agregar el método **FiltrarPlantasFotoBD** a la clase Manejadora para que, de acuerdo con los valores que reciba, genere un listado filtrado de las plantas.

Dicho método llamará por GET (utilizando **AJAX**) hacia **"http://localhost:2023/listarPlantasFiltradasFotosBD"**.

Con el array de tipo Planta recibido, armar una tabla (HTML en el **FRONTEND**) 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='divTablaPlantaFotos'**).

| Plantas filtradas |           |        |   |   |
|-------------------|-----------|--------|---|---|
| Código            | Nombre    | Precio | ColorFlor   | Foto  |
| hel_912           | helecho   | 510    |  |  |
| mar_002           | margarita | 3520   |  |  |
| pot_001           | potus     | 1500   |  |  |