

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio III


Apellido:		Fecha:	7/8/2020
Nombre:		Docente ⁽²⁾ :	Baus/Mutti
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> <div>PP</div> <div></div> <div>RPP</div> <div></div> <div>SP</div> <div></div> <div>RSP</div> <div>X</div> <div>FIN</div> <div></div> </div>		

(1) Las instancias válidas son: 1^{er} Parcial (**PP**), Recuperatorio 1^{er} Parcial (**RPP**), 2^{do} Parcial (**SP**), Recuperatorio 2^{do} Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

(2) Campos a ser completados por el docente.


Realizar una ABM de Anuncios de Automoviles.

- Desarrollar una aplicación que cuente con una única página donde exista el listado de los anuncios dados de alta (mostrar un spinner mientras cargan), así como un formulario para dar de alta a nuevos anuncios. Para esto deberá usar HTML y CSS para darle agradable "look and feel".
- La lista de anuncios deberá ser generada dinámicamente (con código JS) en base a la estructura del objeto utilizado, además deberá contar con algún manejador de eventos, tal que al hacer click en algún elemento de la lista, se carguen los valores del anuncio seleccionado en el formulario. Se deberán habilitar los botones necesarios para permitirnos cancelar la edición o eliminar el anuncio.



[Nosotros](#)
[Anuncios](#)
[Blog](#)
[Contacto](#)

Administracion



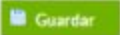
Información Del Anuncio:


TITULO:


TRANSACCION:


DESCRIPCION:

PRECIO:









id	titulo	transaccion	descripcion	precio	num_puertas	num_KMs	potencia
1	Jeep Renegade	venta	Jeep Renegade 2017	1.000.000	5	10.000	120
2	Renault Megane	venta	Renault Megane 2014	800.000	5	80.000	140

[Nosotros](#)
[Anuncios](#)
[Blog](#)
[Contacto](#)

Todos los Derechos Reservados 2019 ©

Especificaciones

header



[Nosotros](#)
[Anuncios](#)
[Blog](#)
[Contacto](#)

1) Crear un Logo de la marca

2) Agregar un navbar

titulo

Administracion



- 1)Agregar un encabezado de Administración
- 2)Una imagen relacionada al negocio de Automóviles.

form

Información Del Anuncio

TITULO:

TRANSACCION:

DESCRIPCION:

PRECIO:



Información Del Anuncio

TITULO:

Titulo

TRANSACCION:

----Elegir Transaccion----

----Elegir Transaccion----

Venta

Alquiler

Permuta

PRECIO:

Precio

Puertas

KMs

Potencia

Guardar

- 1)Respetar los controles de la imagen(el posicionamiento es a gusto).
- 2)Agregar iconos a los input de características del automóvil.
- 3)Agregar icono al botón de “Guardar”.

tabla

Guardar

id	titulo	transaccion	descripcion	precio	num_puertas	num_KMs	potencia
1	Jeep Renegade	venta	Jeep Renegade 2017	1.000.000	5	10.000	120
2	Renault Megane	venta	Renault Megane 2014	800.000	5	80.000	140

- 1)pintar por css el encabezado
- 2)centrar la tabla en la pantalla

footer

1)Respetar posicionamiento

spinner



1)Agregar un spinner acorde al negocio de Automóviles.

Javascript

1)Utilizar una clase “Anuncio” que contenga los atributos:

- id
- titulo
- transaccion
- descripcion
- precio

2)Utilizar una clase “Anuncio_Auto” que herede de “Anuncio” con el resto de los atributos necesarios

Server

- Utilizar JS nativo(XMLHttpRequest) para realizar las llamadas al servidor.
- El servidor será provisto en un archivo llamado **server.js** y no deberá ser modificado. Para correrlo, bastará con correr por línea de comando: “**node server**”. El servidor quedará escuchando en el puerto 3000.
- El navegador por default va a devolvernos **index.html** al apuntarle al puerto 3000 de localhost.

SERVER:

Contrato con el servidor:

1. ALTA:

request(POST):

- **url:** /alta
- El body es un objeto **Anuncio** creado en base a los datos del formulario.
- Se debe usar la función de JSON que convierte a string los objetos al momento de realizar la llamada AJAX.
- Agregar el header "Content-Type":
 - **JS:** xhr.setRequestHeader("Content-Type", "application/json");

2. BAJA:

request(POST):

- **url:** /baja
- Para el borrado alcanza con mandar el id del anuncio.
- Agregar el header "Content-Type":
 - **JS:** xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

3. MODIFICACIÓN

request(POST):

- **url:** /modificar
- El body es un objeto **Anuncio_Auto** creado en base a los datos del formulario.
- Se debe usar la función de JSON que convierte a string los objetos al momento de realizar la llamada AJAX.
- Agregar el header "Content-Type":
 - **JS:** xhr.setRequestHeader("Content-Type", "application/json");

4. Traer

request(GET):

- **url:** /traer

La respuesta del servidor será enviada por ésta función, donde "message" es el mensaje de la operación, y "data" será el array de anuncios si la llamada es al método "traer":

```
function sendResponse(response, message, data) {  
    response.send({ "message": message, "data": data });  
}
```

SEGUNDO PARCIAL

Map/Reduce/Filter

- Agregar una sección que permita filtrar la tabla que se muestra por pantalla por tipo de transacción, y muestre por pantalla el promedio de precio(solo mostrar en caso de que se seleccione “venta”, “alquiler”, o “permuta”). Mostrar “N/A” si no hay filtros aplicados.
- Agregar controles que permitan seleccionar qué columnas de la tabla mostrar por pantalla

FILTRAR TRANSACCION

Venta

PROMEDIO

900,000

TITULO☒TRANSACCION☒DESCRIPCION☒PRECIO☒NUM_PUERTAS☒NUM_KMS☒

POTENCIA☒

id	titulo	transaccion	descripcion	precio	num_puertas	num_KMs	potencia
1	Jeep Renegade	venta	Jeep Renegade 2017	1000000	5	10.000	120
2	Renault Megane	venta	Renault Megane 2014	800000	5	80.000	140

Bootstrap

- Utilizar la librería de Bootstrap y su esquema de columnas donde crea conveniente para lograr un comportamiento “responsive”.
- Utilizar entre 5 y 10 clases de Bootstrap(además de las ya utilizadas en el punto anterior).

Typescript

- Cargar el select de transacciones con un enum Etransaccion

localStorage

- Almacenar las columnas seleccionadas por el usuario en localStorage, de manera que al volver a cargar la página, se aplique automáticamente la última selección.

PRIMER RECUPERATORIO

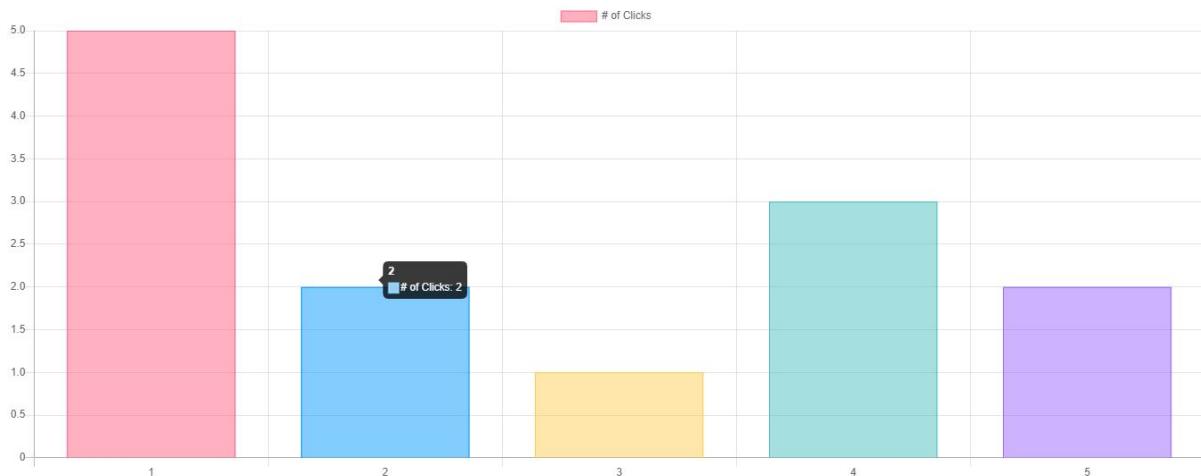
- Utilizando map/reduce/filter, agregar(junto con el promedio de precios), el precio maximo, el minimo, y el promedio de potencia.
- Convertir las clases existentes en JS a Typescript
- Eliminar por completo la interacción con el servidor. Realizar el ABM contra localStorage.

SEGUNDO RECUPERATORIO

- Dado un array almacenado en el localStorage con los clicks que el usuario le dio a los anuncios:

`[1, 2, 3, 1, 2, 1, 1, 1, 4, 5, 5, 4, 4];`

crear, usando algún framework de de charts para JS, un gráfico que muestre cual es el anuncio más visitado:



- Crear una clase de TypeScript llamada DataAccess que contenga todos los métodos de las 4 llamadas del ABM(trazer,insertar,borrar,modificar). Utilizar dicha clase en el código de la aplicación.