

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio III

Apellido:		Fecha:	19/5/2020
Nombre:		Docente ⁽²⁾ :	Baus/Mutti
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around; padding: 2px;"> PP X RPP SP RSP FIN </div>		

(1) Las instancias validas 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 Bienes Raices.

- 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”.
- El formulario deberá contar con un control apropiado para cada atributo del anuncio. También deberá contar con “placeholder” y una validación para hacer los datos requeridos.
- 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.

Llena el formulario de Alta

Información Del Anuncio

TITULO:

TRANSACCION:

VENTA

ALQUILER

DESCRIPCION:

PRECIO:

CANTIDAD DE BAÑOS:

CANTIDAD DE AUTOS:

CANTIDAD DE DORMITORIOS:

Guardar

Guardar			Eliminar	Cancelar			
id	título	transaccion	descripcion	precio	num_wc	num_estacionamiento	num_dormitorio
1	Casa de Lujo en la montaña	alquiler	Casa en la montaña con excelente vista, acabados de lujo a un precio irresistible	\$4,500,0000	3	1	7

- 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:** /bajaAnuncio
- 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:** /modificarAnuncio
- 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");

4. Traer

request(GET):

- **url:** /traer

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

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

