

**Universidad Tecnológica Nacional**  
**Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio III**

Apellido:		Fecha:	26/11/2019
Nombre:		Docente <sup>(2)</sup> :	Baus/Mutti
División:		Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around;"> <div>PP</div> <div></div> <div>RPP</div> <div></div> <div>SP</div> <div>X</div> <div>RSP</div> <div></div> <div>FIN</div> <div></div> </div>		

(1) Las instancias validas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

(2) Campos a ser completados por el docente.

**Realizar una ABM de Anuncios.**

- 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:** /altaAnuncio
- 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");

#### response:

```
setTimeout(function(){response.send("Alta Exitosa"); },5000);
```

### 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");

#### response:

```
setTimeout(() => {response.send({"message":"Baja exitosa"}});}, 1000);
```

### 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");

#### response:

```
setTimeout(function () {response.send("Modificacion exitosa"); }, 1000);
```

### 4. Traer

#### request(GET):

- **url:** /traerAnuncios

#### response:

```
setTimeout(function(){response.send({"message": "Carga exitosa","data":array}});,5000);
```

## Segundo Parcial.

- **LocalStorage:** Retirar por completo el código del servidor. Realizar el ABM contra el localStorage.
- **Typescript:** Realizar las clases **Anuncio** y su heredada **BienRaiz**. Estas clases servirán para soportar el negocio de la aplicación. También realizar **ETransaccion** para almacenar el tipo de transacción(**Alquiler, Venta**).
- **Bootstrap:** Aplicar las clases de Bootstrap necesarias para lograr hacer responsive alguna sección de la página.
- **JQuery:** Utilizar JQuery para interactuar con el DOM.
- **Map/Reduce/Filter:** Utilizar las 3 funciones de Array para lograr columnas dinamicas(seleccionables), para calculo de promedio de precios, y para filtrar la tabla en base al tipo de Transacción seleccionada, respectivamente.

### Filtros

FILTRAR POR:

PROMEDIO DE PRECIOS:

Todos

4

FILTRAR POR:

Todos

Todos

Alquiler

Venta

Campos mostrados Tildar los campos a mostrar							
<input checked="" type="checkbox"/> TITULO <input checked="" type="checkbox"/> TRANSACCION <input checked="" type="checkbox"/> DESCRIPCION <input checked="" type="checkbox"/> PRECIO <input checked="" type="checkbox"/> BAÑOS <input checked="" type="checkbox"/> ESTAC. <input checked="" type="checkbox"/> DORMITORIOS							
Id	Titulo	Transaccion	Descripcion	Precio	Baños	Estacionamientos	Dormitorios
1	a	Venta	a	2	2	2	2
2	b	Alquiler	b	4	4	4	4
3	c	Venta	c	6	6	6	6

**Nota1:** El select del filtro puede estar cargado con los valores de ETRansaccion. Además al select se le puede agregar la opción “Todos” como seleccionada por defecto.

**Nota2:** El promedio de precios se calcula en base a los datos filtrados.

**Nota3:** Los datos mostrados en la tabla son los que resultan de aplicar el filtro seleccionado, y restringiendo las columnas a las seleccionadas en los checkbox.