

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



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio III**

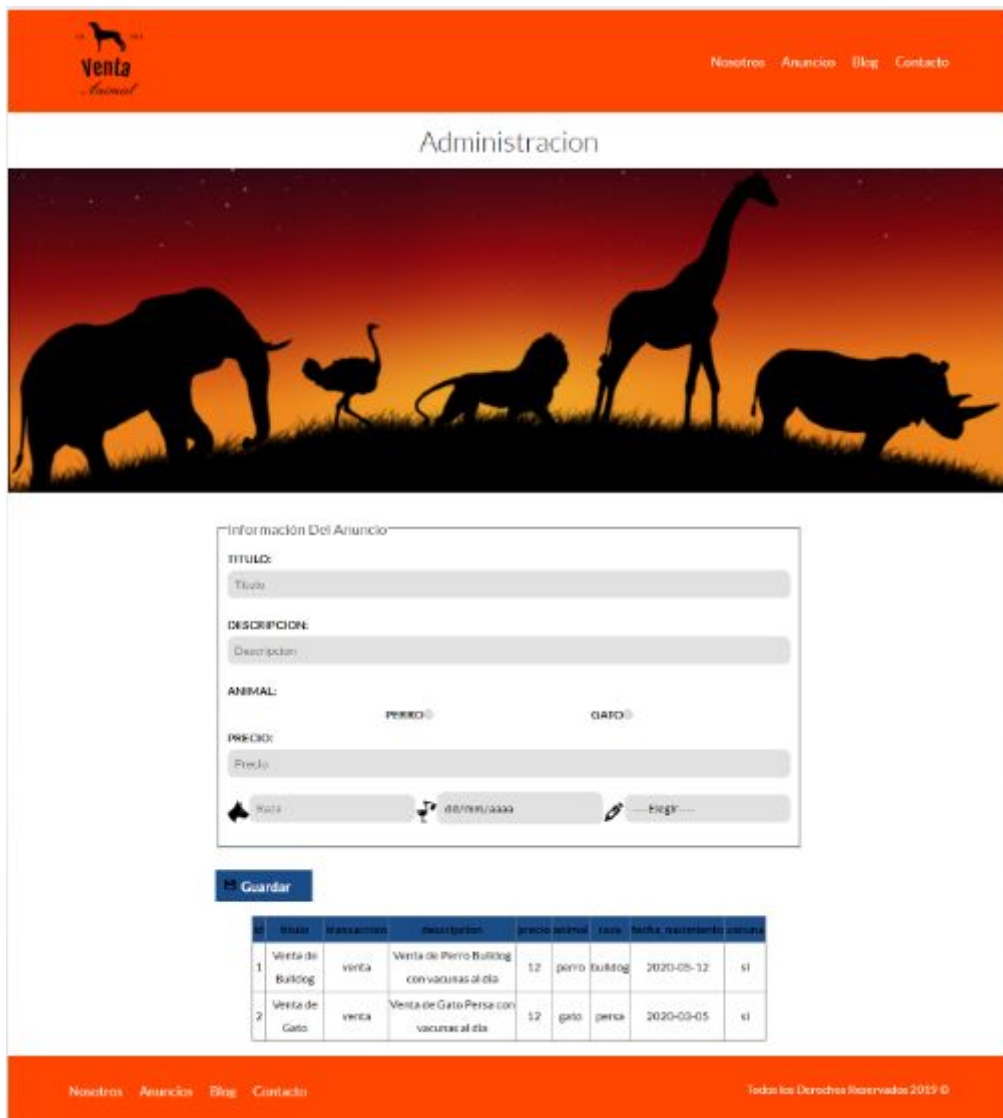
Apellido:		Fecha:	26/5/2020
Nombre:		Docente <sup>(2)</sup> :	Baus/Mutti
División:		Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around; padding: 2px;"> <div style="border: 1px solid black; padding: 2px 5px;">PP</div> <div style="border: 1px solid black; padding: 2px 5px;">X</div> <div style="border: 1px solid black; padding: 2px 5px;">RPP</div> <div style="border: 1px solid black; padding: 2px 5px;"></div> <div style="border: 1px solid black; padding: 2px 5px;">SP</div> <div style="border: 1px solid black; padding: 2px 5px;"></div> <div style="border: 1px solid black; padding: 2px 5px;">RSP</div> <div style="border: 1px solid black; padding: 2px 5px;"></div> <div style="border: 1px solid black; padding: 2px 5px;">FIN</div> <div style="border: 1px solid black; padding: 2px 5px;"></div> </div>		

(1) Las instancias válidas 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 de Mascotas.**

- 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.



## Especificaciones

### header



1) Crear un Logo de la marca

2) Agregar un navbar

**titulo**

Administracion



- 1)Agregar un encabezado de Administración
- 2)Una foto relacionada al negocio de Animales.

**form**

Información Del Anuncio

TITULO:

Titulo

DESCRIPCION:

Descripcion

ANIMAL:

PERRO

GATO

PRECIO:

Precio

Raza


dd/mm/aaaa


----


Elegir----

 Guardar

- 1)Respetar los controles de la imagen(el posicionamiento es a gusto).
  - El control de vacunas es un select con opciones de “Si” y “No”.
- 2)Agregar iconos a los input de características del automóvil.
- 3)A partir de cierto ancho de pantalla, cambiar(a gusto, usando media queries) la manera en la que se muestran las características de la mascota. Ejemplo:








**Guardar**

4)Agregar icono al botón de “Guardar”.

**tabla**

--	--	--	--	--	--	--	--	--	--


**Guardar**

id	titulo	transaccion	descripcion	precio	animal	raza	fecha_nacimiento	vacuna
1	Venta de Bulldog	venta	Venta de Perro Bulldog con vacunas al día	12	perro	bulldog	2020-05-12	si
2	Venta de Gato	venta	Venta de Gato Persa con vacunas al día	12	gato	persa	2020-03-05	si

1)pintar por css el encabezado

2)centrar la tabla en la pantalla

**footer**

1)Respetar posicionamiento

## spinner

 Guardar



1) Agregar un spinner acorde al negocio de Animales.

## Javascript

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

- id
- titulo
- transaccion (por defecto todas van a ser ventas)
- descripcion
- precio

2) Utilizar una clase “Anuncio\_Mascota” 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\_Mascota** 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\_Mascota** 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 });  
}
```