

**Universidad Tecnológica Nacional  
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio III**

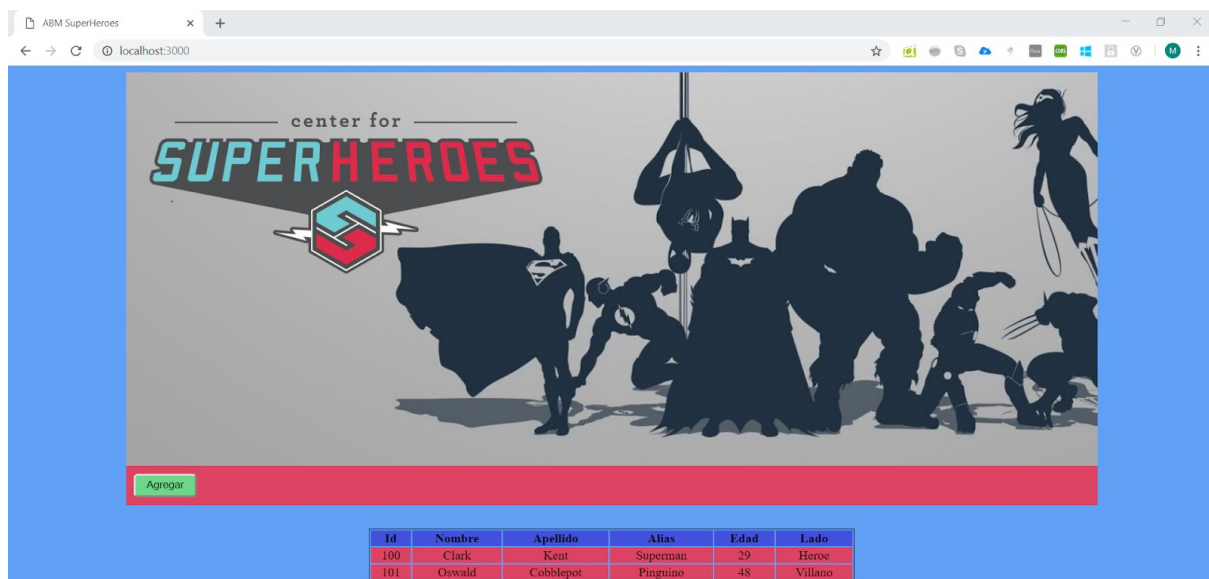
Apellido:		Fecha:	27/11/2018
Nombre:		Docente <sup>(2)</sup> :	Baus/Mutti
División:		Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	PP	RPP	SP X RSP FIN

(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.

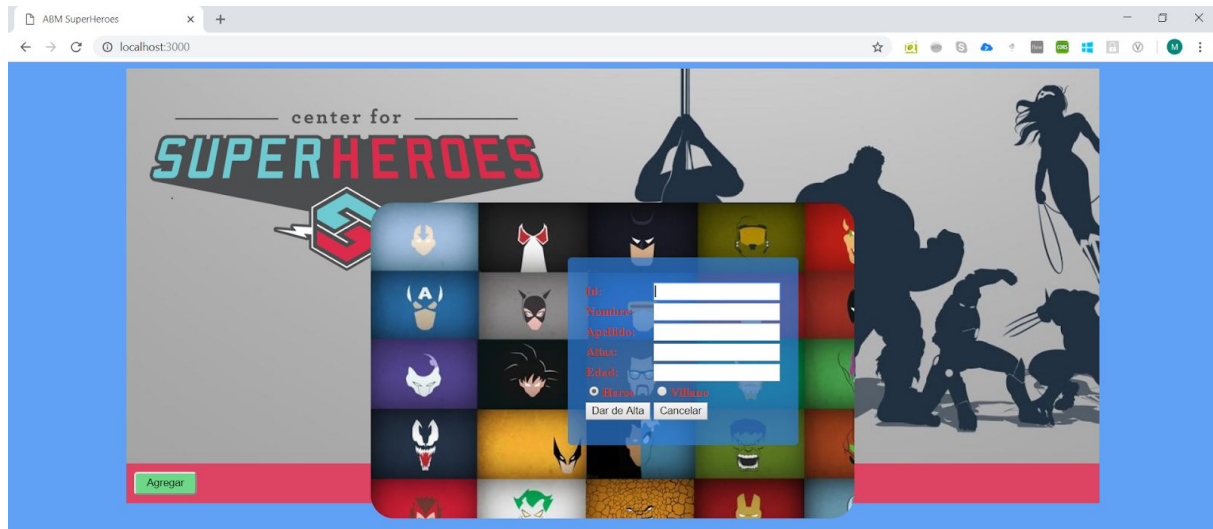
(2) Campos a ser completados por el docente.

**Realizar una ABM de Super Heroes y Villanos.**

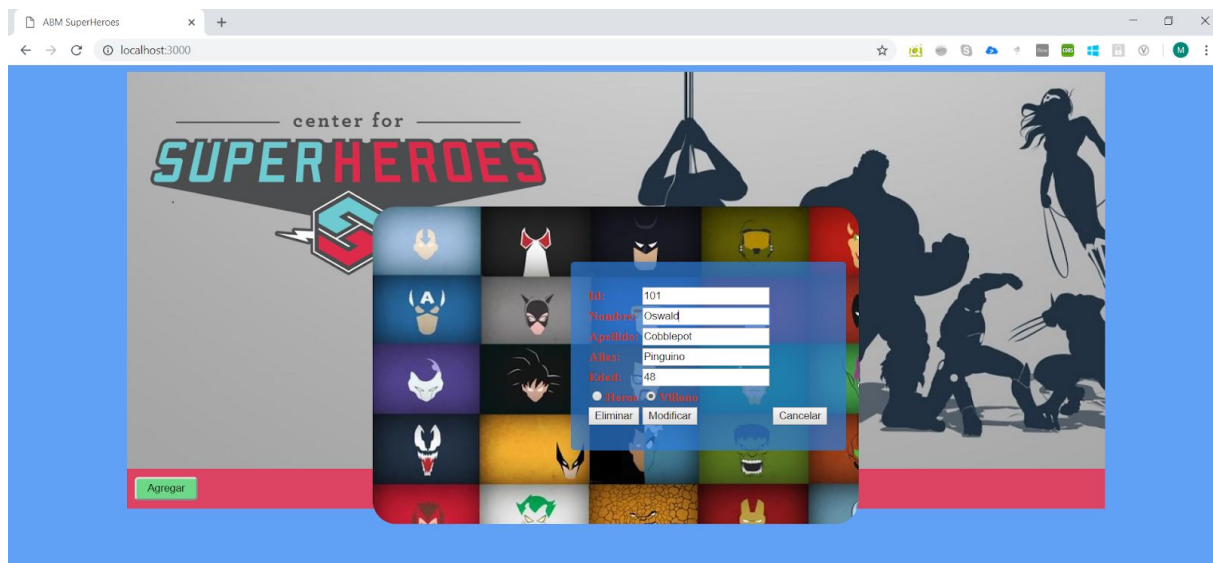
- Desarrollar una aplicación **en capas**, que cuente con una única página donde exista el listado de los héroes y villanos dados de alta, así como un botón para dar de alta a nuevos personajes. Para esto deberá usar HTML y CSS para darle agradable “look and feel”.



- El botón de alta abrirá un formulario(utilizando alguna animación) donde se podrán ingresar los datos del nuevo personaje. Asimismo, el formulario contendrá botones para aceptar y cancelar. Mientras se espera la respuesta del servidor, mostrar un spinner. Todos los campos del formulario deberán tener al menos una validación.



- La lista de personajes deberá contar con algún manejador de eventos, tal que al hacer click en algún elemento de la lista, se muestre un nuevo formulario que nos permita eliminar o modificar el personaje.



- Utilizar JQuery en al menos un método de AJAX.
- Utilizar JS nativo(XMLHttpRequest) en al menos un método de AJAX.
- 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.

## SEGUNDO PARCIAL:

- Reemplazar íntegramente la capa de acceso a los datos de manera tal que ahora todas nuestras consultas se hagan contra el **localStorage**.
- Agregar una sección en la que, en base a un filtro por tipo seleccionado, se pueda visualizar el promedio de edad, y el personaje mas viejo. En caso de no haber filtro seleccionado, se hará el cálculo sobre la lista completa de personajes.

Filtrar por Tipo

Seleccionar...

▼

Promedio de Edad:

32.50

Personaje mas viejo:

LexLuthor

Campos mostrados

Tildar los campos a mostrar

☒ Id
☒ Nombre
☒ Apellido
☒ Alias
☒ Edad:
☒ Lado
☒ Editorial

Id	Nombre	Apellido	Alias	Edad	Lado	Editorial
111	Clark	Kent	Superman	29	Heroe	dc
222	Bruce	Wayne	Batman	40	Heroe	dc
333	Alexander	Luthor	LexLuthor	40	Villano	dc
444	Anthony	Stark	Ironman	21	Heroe	marvel

- Los nuevos controles deberán ser “**responsive**”. Significa que al achicar la pantalla, deberán adoptar otra disposición:

Filtrar por Tipo

Seleccionar...

▼

Promedio de Edad:

32.50

Personaje mas viejo:

LexLuthor

- Agregar una seccion que permita elegir las columnas de la tabla que son visualizadas:

Campos mostrados

Tildar los campos a mostrar

☒ Id
☒ Nombre
☒ Apellido
☒ Alias
☒ Edad:
☒ Lado
☒ Editorial

- Usando **TypeScript**, realizar las clases **Personaje**, y **Héroe**(que hereda de Personaje). Agregar las propiedades públicas que crea necesarias para encapsular los atributos privados(hacer al menos una para leer y una para escribir).