

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio III

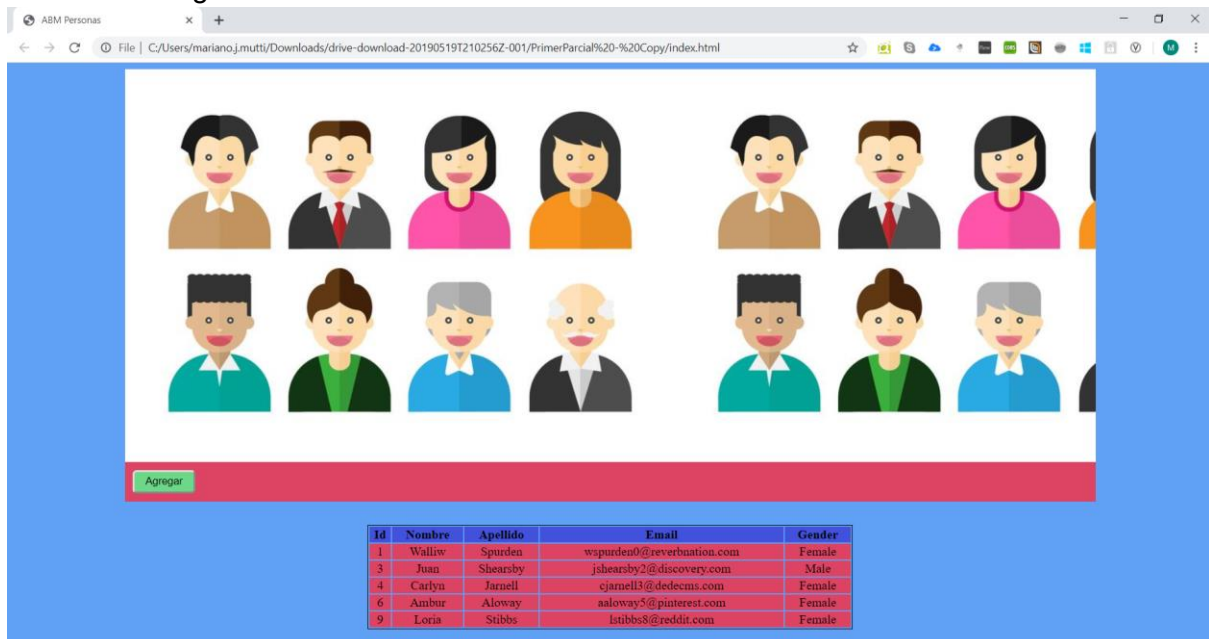
Apellido:		Fecha:	1/8/2019
Nombre:		Docente ⁽²⁾ :	Christian Baus
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP RPP SP X RSP FIN </div>		

(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.

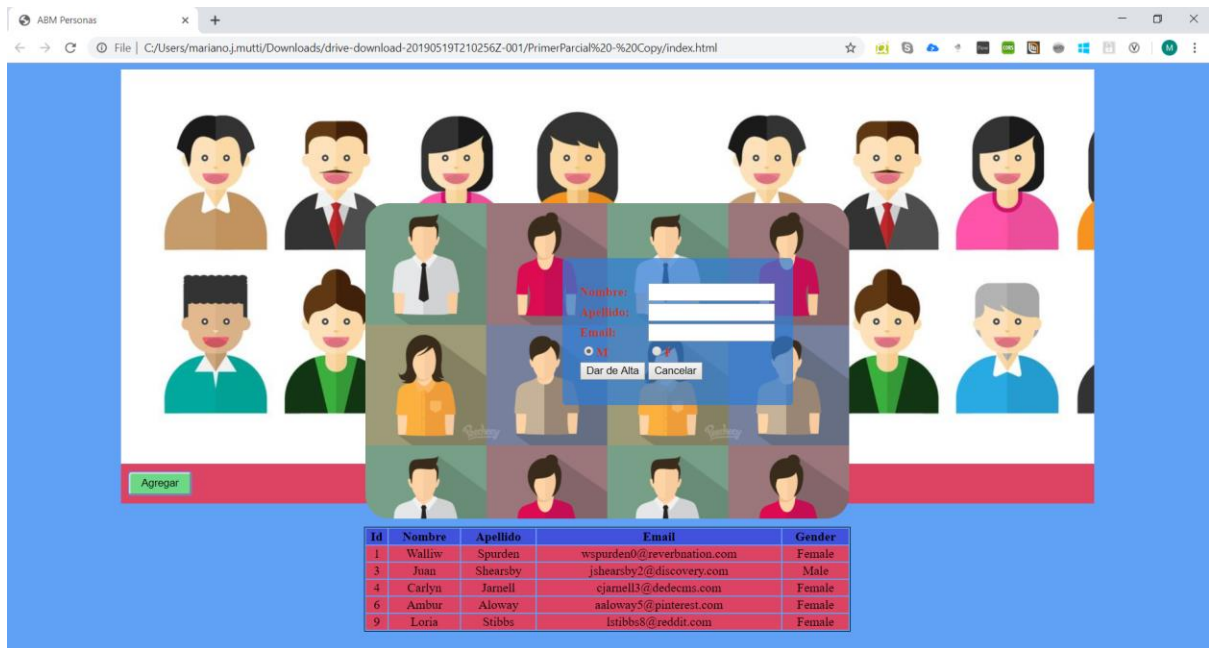
(2) Campos a ser completados por el docente.

Realizar una ABM de Personas.

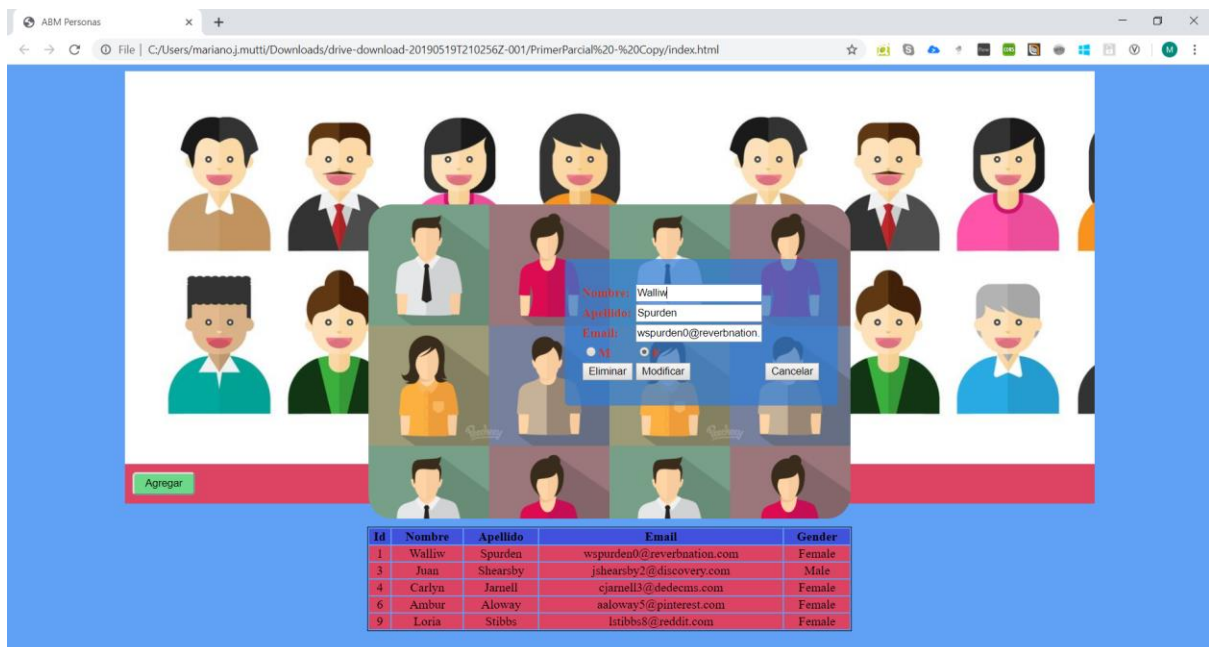
- Desarrollar una aplicación que cuente con una única página donde exista el listado de las personas dadas de alta(mostrar un spinner mientras cargan), así como un botón para dar de alta a nuevas personas. Para esto deberá usar HTML y CSS para darle agradable “look and feel”.



- El botón de alta abrirá un formulario donde se podrán ingresar los datos de la nueva persona. Asimismo, el formulario contendrá botones para aceptar y cancelar.
- El formulario deberá contar con un control apropiado para ingresar emails. También deberá contar con una validación para hacer los datos requeridos.



- La lista de personas 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 a la persona seleccionada.



- 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**.

- Generar el ID(autoincremental) para cada legislador que se inserta.
- Agregar un campo numérico para ingresar la edad.
- Agregar un par de input radio para elegir el tipo de legislador (Diputado/Senador).

Nombre:
Apellido:
Email:
Edad:
Sexo: ☒ **Mujer** ☐ **Hombre**
☒ **Diputado** ☐ **Senador**

- Agregar una sección en la que, en base a un filtro por tipo de legislador seleccionado, se pueda visualizar el “gender mix”(porcentaje de mujeres sobre el total), y el promedio de edad de los legisladores.

Filtros

Filtrar por: Todos ▾
 Promedio de Edad: 39.5
 Gender Mix: 50 %

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

Filtros

Filtrar por: Todos ▾
 Promedio de Edad: 39.5

Gender Mix: 50 %

Filtros

Filtrar por: Todos ▾

Promedio de Edad: 39.5

Gender Mix: 50 %

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

Campos mostrados Tildar los campos a mostrar

☒ Id ☒ Nombre ☒ Apellido ☒ Email ☒ Edad ☒ Sexo ☒ Tipo

id	nombre	apellido	email	edad	sexo	tipo
100	Pedro	Perez	pedro.perez@a.com	44	hombre	Diputado
101	Silvia	Rodriguez	silvia.rodriguez@b.com	35	mujer	Senador

- Usando **TypeScript**, realizar las clases **Persona**, y **Legislador** (que hereda de Persona). Agregar las propiedades públicas que crea necesarias para encapsular los atributos privados (hacer al menos una para leer y una para escribir). Además, agregar un enumerado **tipo Legislador** que contenga los valores para Diputado y Senador y que sea utilizado por la clase Legislador.