Universidad Tecnológica Nacional **Facultad Regional Avellaneda**										
Técnico Superior en Programación										
Materia: Laboratorio III										
Apellido:	Coarasa			Fecha:		07/08/2020				
Nombre:	Walter			Docente(2):		Ramos Matias				
División:	3c			Nota(2):						
Legajo:	102099			Firma(2):						
Instancia:	**PP**		**RPP**		**SP**		**RSP**	**X**	**FIN**	

Crear una aplicación web que permita realizar un ABM, de distintos empleados. El objetivo es tener la siguiente funcionalidad:

1- (1pts) Crear la siguiente jerarquía de clases en Typescript:
   a. ***Persona***: nombre(string), apellido(string) y edad(entero) como atributos. Un constructor que reciba tres parámetros. Un método, p*ersonaToJson* que retorne la representación de la clase en formato cadena.
   b. ***Empleado***, hereda de persona, posee como atributos legajo(entero) y horario(string). Un constructor para inicializar los atributos. Un método *empleadoToJson* que retornará la representación del objeto en formato JSON. Se debe de reutilizar el método *personaToJson*.
2- (2pts) Crear la página ***index.html***, la misma contendrá dos secciones:
   a. Formulario de ingreso/modificación de datos
   b. Listado de todos los empleados y listado de empleados filtrados.
   c. Para el nombre, apellido, edad y legajo se deberá utilizar input del tipo que corresponda y para horario select
   d. Utilice los tipos de inputs de html 5 para que solo se puedan ingresar números en los inputs edad y legajo
3- Crear la clase ***Manejadora*** que posea los siguientes métodos:
4- (2pts) **agregarEmpleado**. Tomará los distintos valores desde la página index.html y creará y guardará en el array a ese empleado. El método debe poder almacenar 'n' empleados.
5- (1pts) **mostrarEmpleados**. Generará un listado dinámico que mostrará toda la información del empleado. Agregar columnas que permitan: Eliminar y Modificar al empleado elegido.
6- (1pts) **eliminarEmpleado**. Eliminará al empleado del array. Refrescar el listado para visualizar los cambios.
7- (2pts) **modificarEmpleado**. Mostrará todos los datos del empleado y permitirá modificar cualquier campo. Refrescar el listado para visualizar los cambios.
8- (1pts) Agregar botón filtrar por horario. Invocará al método **filtrarPorHorario** y retornará un listado completo de todos los empleados según el horario seleccionado. Utilice filter.
9- (2pts) Agregar un botón obtener promedio de edad según horario. Invocará al método **promedioEdadPorHorario** y mostrar en un input el promedio de edad. Utilice reduce.
10- (2pts) Agregar un botón para que solo se muestren los nombres y apellidos. Utilice el map.

IMPORTANTE:

Se pueden bajar templates de internet o traer código hecho, pero en ningún caso se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.

Se tiene que sumar 8 puntos para lograr un cuatro (4). Nueve puntos equivalen a un cinco (5), diez puntos equivalen a seis (6), once puntos: siete (7), doce puntos a un ocho (8), trece puntos equivalen a un nueve (9) y si se suman los catorce puntos la nota será de diez (10).

https://github.com/coarasaw/RSParcialLab3.git