

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Computación IV

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

En nuestro sistemas manejamos las materias de una facultad, con tres tipo de usuarios(alumno, profesor y administrador).
para toda entidad hacemos el componente que nos permita dar de alta /editar y el que muestre un listado de esa entidad como mínimo, y la comunicación se debe realizar por input ,output y servicios.

temas a evaluar :

- can activate
- archivos
- animaciones
- PIPE
- directivas
- servicios
- Validaciones

1era parte

1- Crear el formulario de alta de un usuario con los siguientes campos: mail, clave , tipo de usuario (alumno, profesor)**Validar los datos.**

2- Crear el formulario de login, si el login es exitoso redirigir al home de la página, de lo contrario informar el error.

3- Autenticar las rutas usando canActivate y hacer al menos dos animaciones de transición de angular.

realizar el login con al menos dos usuarios por cada tipo(6 usuarios mínimo en total) harcodeados para agilizar el ingreso, con botones que nos permita cargar los datos en los cuadros de textos del login

4- Realizar un componente para que el email del usuario siempre esté visible en la página.

este componente tiene que estar en todas las páginas de nuestro sistema

5- Solo lo puede ver el administrador(canActivate)

a. Crear el formulario de alta de un administrador con los siguientes campos: mail, clave **Validar los datos.**

b. Dar de alta una materia con los siguientes datos (validar los datos):
Nombre de la materia, Cuatrimestre, Cupo de alumnos, año,
Profesor(debe existir en la base), se muestra con el componente que lista los profesores y por input o output debería comunicarse con el alta de materia.

c. inscripcion materia : muestra el listados de materias y el listado de alumnos permitiendo seleccionar y agregar alumnos mientras el cupo de alumnos lo permita de esa materia

d. Ver listado de materias y al seleccionar una materia ver los alumnos de las materias (input y output)
e. Ver listado de usuarios, puede filtrar por tipo. Usar un **PIPE** en el listado para cambiar lo que se va a mostrar.

6 Para Alumno:

- a. Inscribirse en una materia.
- b. Ver materias en las que está inscripto.
- c. Generar un PIPE que muestre aprobado, desaprobado, promocionado. (aprobado 4-6. desaprobado 1-4. promocionado 6-10)

7- Generar una directiva que cambie el color de las materias si el cupo es mayor a 10 y otro color si es mayor a 20 en los listado de materias.

NOTA : aplicar animaciones a su navegación

2da parte

8- Para profesor:

- a. Ver materias a cargo
- b. Ver alumnos inscriptos en materia que dicta.

9-Para Materia:

- a. Agregar en el alta de la materia una foto.

- b. Generar un componente para ver el detalle de la materia. (Mostrar foto y listado de alumnos inscriptos)
- c. Generar un componente que solo ingrese el profesor y muestra el listado de materias a su cargo y al seleccionar una se muestre su detalle en el componente anterior.

3ra parte

10-Para Alumno:

- a. Permitir solo al administrador, poder borrar a un alumno.
 - b. Al buscar el detalle de las materias , el alumno no debería estar
 - c. Guardar los alumnos borrados, con la fecha de borrado. Generar el componente que muestre el listado con imágenes de alumnos borrados(generar los servicios necesarios).
-

11-Para Examen:

- a. Permitir solo al Profesor, poder cargar a un examen.
- b. se carga alumno(solo alumno inscripto), fecha y nota (validada)
- c. Guardar los exámenes ,. Generar el componente que muestre el listado con imágenes de alumnos que participaron en el examen con la nota obtenida (generar los servicios necesarios).

4ta parte

12-Para actas de examen:

Requisitos:

tener 3 profesores con materias:

usuario: octavio clave: octavio

usuario :matias , clave matias

usuario :maxi clave maxi

con tres materias cada uno

octavio = {matemáticas, estadísticas, base de datos}

matias={laboratorio, programación, spd}

maxi={spd, pps, so}

con cinco alumnos en cada materia.

a. Permitir solo al Profesor, ingresa al componente "actasDeExamen"

b. este componente tiene a tres componentes en su HTML

1- "listadoDeAsignaturas" (según el profesor que recibe por @Input, muestra las asignaturas, y un @Output que al seleccionar una asignatura, cambien los dos listados)

2- "listadoAprobaciónDirecta" (recibe un listado de exámenes con el alumno y su nota , pero solo muestra los que son igual o superan la nota 7) .

3- "listadoNoDirecta" (recibe un listado de exámenes con el alumno y su nota , pero solo muestra los que NO superan la nota 7)

c. Utilizar un PIPE para diferenciar los aprobados de los no aprobados en "listadoNoDirecta"

5ta parte

13-Para Alumno:

a. permitir ver las notas de las actas de examen donde esta .

b generar un pipe que muestre el estado de la materia (aprobado, no aprobado , vacío ... etc)

para esto hacer un componente que muestre las actas en las cual está el alumno y al seleccionar esa acta ,
en otro componente que recibe el dato , se muestra la nota del alummo en esa materia y en otro
componente los datos del acta , como fecha docente, materia ,etc ...
