






Parte 02

registro.html – registro.ts

REGISTRO

Apellido		<input type="text" value="Apellido"/>
Nombre		<input type="text" value="Nombre"/>
E-mail		<input type="text" value="E-mail"/>
Legajo		<input type="text" value="Legajo"/>
Perfil		<div>Invitado ▼</div>
Foto		<div><input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado</div>
Clave		<input type="password" value="Clave"/>
Confirmar		<input type="password" value="Repita la clave"/>

Se registrarán los datos de un nuevo usuario.

Verificar que el correo no exista previamente. En caso de existir, mostrar en un alert de BOOTSRAP (danger) el mensaje correspondiente.

Se guardará en el array de JSON el nuevo usuario y se redirigirá hacia *login.html*.

Los campos de entrada tendrán aplicadas validaciones:

- Apellido: La longitud máxima será de 15 caracteres.
- Nombre: La longitud máxima será de 10 caracteres.
- E-mail: Valida el formato de correo.
- Legajo: Debe ser un número entero de entre 3 y 6 dígitos.

- Foto: Sólo admitirá extensiones 'jpg' y 'png'.
- Clave: Admite longitudes de entre 4 y 8 caracteres. Debe coincidir con la confirmación de clave.
- Confirmar: Admite longitudes de entre 4 y 8 caracteres. Debe coincidir con la clave.

En todos los casos, se requerirá un valor, es decir, los campos no pueden estar vacíos. Mostrar un asterisco (*) rojo en el campo erróneo y en otro alert de BOOTSRAP (danger) el mensaje de error correspondiente.

***Nota:** tener previamente guardadas distintas imágenes en la carpeta “./fotos/” y nombrarlas con los valores correspondientes del array de JSON.*

***Nota 2:** tener en cuenta que el valor del <input type= "file"> es similar a: C:\\fakepath\\NombreFoto.jpg. Guardar solo NombreFoto.jpg.*