

Profesor:

Neiner Maximiliano

## 1.- Creación del formulario de Login

Realizar la página **login.html**, en HTML 5, que posea una estructura como la de la siguiente imagen:

**Login**

**Ingreso de credenciales**

DNI:

Apellido:

Detalles técnicos

### login.html

- El título de la página debe ser: HTML 5 – Formulario Login (<title>)
- El encabezado principal de la página se hará de nivel 2 (<h2>)
- El encabezado de la sección del formulario será <h4>
- El campo para el DNI (*txtDni*) será de tipo numérico, con un valor mínimo de 1.000.000 y un valor máximo de 55.000.000.

**Nota:** Utilizar elementos de tipo <form>, <table>, <h#> y <hr> para el maquetado de la página.

## 2.- Validación de datos del Login

Crear en TypeScript las funciones necesarias para verificar que todos los campos de la página sean enviados correctamente.

Para indicar que algún campo es erróneo, se agregará al lado de cada campo un tag `<span>`, que como contenido tendrá el carácter `*`.

Dicho elemento poseerá, por defecto, el valor `'display:none'` del atributo `"style"` (esto hace que el tag, y su contenido, sean 'invisibles').

Al detectarse un error sobre el campo, se deberá hacer visible al tag `<span>`, colocando el valor `'display:block'` al atributo `style`.

Para ello, asociar al evento click del botón Enviar (*btnEnviar*) una función que administre dicha tarea, indicando el error o permitiendo el "envío", según corresponda.

La función se llamará *AdministrarValidacionesLogin* y será la encargada de invocar a otras funciones que verifiquen:

- Campos no vacíos.
  - **ValidarCamposVacios(string): boolean**. Utilizar la función creada en la parte 2 del TP.
- Rangos numéricos correctos.
  - **ValidarRangoNumerico(number, number, number): boolean**. Utilizar la función creada en la parte 2 del TP.
- Indicar el campo con errores.
  - **AdministrarSpanError(string, boolean): void**. Es la encargada de, según el parámetro booleano, ocultar o no al elemento cuyo id coincida con el parámetro de tipo string.
- Verificar si se envía el formulario.
  - **VerificarValidacionesLogin(): boolean**. Deberá determinar si todos los campos del formulario están validados. Retornará true, si ningún `<span>` posee `display:block` como valor del atributo `style`, false caso contrario.

**Nota:** Utilizar los mismos archivos creados en la parte 2 del TP.

## 3.- Validación de datos del alta de empleados

Reutilizar las funciones creadas en TypeScript y verificar que todos los campos de la página *index.html* sean enviados correctamente. Reemplazar los mensajes (alert) por los `*` (`<span>`).

## Alta de Empleados

### Datos Personales

DNI:

\*

Apellido:

\*

Nombre:

\*

Sexo:

\*

### Datos Laborales

Legajo:

\*

Sueldo:

\*

Turno:

- ☒ Mañana  
☐ Tarde  
☐ Noche

Limpiar

Enviar

#### 4.- Maquetar el listado de empleados

Realizar en la página ***backend/mostrar.php*** una tabla, en HTML 5, que contenga la información de los empleados.

La misma tendrá un aspecto similar a la de la siguiente imagen.

## Listado de Empleados

### Info

1000001 - Juan - Perez - M - 101 - 8500 - T	<a href="#">Eliminar</a>
1000033 - Susana - Gimenez - F - 112 - 9500 - N	<a href="#">Eliminar</a>
1000008 - Gustavo - Cordera - M - 112 - 25000 - N	<a href="#">Eliminar</a>
1000038 - Roberto - Sanchez - M - 106 - 20000 - M	<a href="#">Eliminar</a>

[Alta de Empleados](#)

mostrar.php

- El título de la página debe ser: HTML 5 – Listado de Empleados (<title>)
- El encabezado principal de la página se hará de nivel 2 (<h2>)
- El encabezado de la sección de la tabla será <h4>
- La tabla tendrá dos columnas (<td>) y tantas filas (<tr>) como empleados tenga el archivo de texto.
- Antes y después de la información de los empleados se colocará una línea divisoria (<hr>)

**Nota:** Utilizar elementos de tipo <table>, <h#> y <hr> para el maquetado de la página.