

Ejercicios básicos de validaciones

[1.01] Registro de usuarios

Descargar el proyecto inicial desde [aquí](#).

Examinar el contenido y verificar las validaciones básicas de HTML5.

Se pide:

- ❖ agregar el atributo **novalidate** al formulario (esto impide la validación automática del navegador).
- ❖ agregar, en **./js/manejadora.js**, las verificaciones necesarias para determinar si todos los campos del formulario son correctos o no. Utilizar la función **checkValidity()** del formulario e indicar mediante ventanas de tipo **alert** lo acontecido.
- ❖ agregar al formulario la clase **was-validated** de Bootstrap al final de las instrucciones de JavaScript.

[1.02] Mensajes para cada error

Al proyecto del ejercicio anterior, se pide modificar el archivo **./index.html** para agregar debajo de cada control un `<div>`.

- ❖ agregar a cada elemento `<div>` la clase de Bootstrap **invalid-feedback** (hará no visible el contenido del elemento hasta que el formulario sea validado y solo en caso de error se visibiliza).
- ❖ agregar un mensaje significativo a cada elemento `<div>` que describa el error.

Verificar el funcionamiento de la página.

Nota: por convención se agrega la clase **needs-validation** al formulario para indicar que los campos del formulario serán validados de forma manual con JavaScript.

[1.03] Mensajes dinámicos

A partir del ejercicio anterior, se pide identificar cada uno de los campos que contienen errores mediante JavaScript.

Utilizar la pseudo clase de Bootstrap **:invalid** que se aplica a los elementos del formulario que están en estado inválido (*required*, *type='email'*, *minlength*, etc).

- ❖ mostrar por consola los identificadores de dichos elementos (**campo.id**).
- ❖ mostrar por consola y analizar el objeto **validityState** de los elementos inválidos (acceder por medio del atributo **validity** del elemento).
- ❖ a todos los elementos `<div>` agregados en el punto anterior, quitarles el texto del mensaje y agregarles identificadores significativos (*error-campo*).
- ❖ mediante JavaScript generar dichos mensajes de forma dinámica y personalizada.

Verificar el correcto funcionamiento de la página.