

Bloque 6 - Formularios

[Bloque 6 - Formularios](#)

[Introducción](#)

[Servicios para comprobación de formularios](#)

[Ejercicios](#)

[Ejercicios Opcionales](#)

Introducción

Los formularios en HTML5 se utilizan para recopilar información de los usuarios a través de la web. Un formulario en HTML5 se define utilizando el elemento `<form>`, que contiene uno o varios elementos de entrada de datos, como `<input>`, `<textarea>`, `<select>`, etc.

En este tema completamos CSS con selectores y propiedades especialmente dirigidos a formularios.

Teoría y ejemplos en Bloque 6 - Tema 1 - Formularios

A la vez que avanzamos con formularios y CSS repasamos todo lo trabajado hasta ahora.

Servicios para comprobación de formularios

Para comprobar que los formularios funcionan correctamente utilizaremos los siguientes servicios:

- <https://forms.antonioberrocal.repl.co/> → independiente del método de envío. Devuelve una página Web con los datos que se enviaron al servidor en formato de tabla. Funciona con datos múltiples en select si el nombre incluye al final []
- <https://httpbin.org/post> y <https://httpbin.org/get> → envíos POST y GET respectivamente. Devuelve un fichero JSON con los datos enviados e información sobre las cabeceras HTTP. Funciona con datos múltiples en select
- **Opcionalmente** puedes enviar los datos del formulario por mail a tu cuenta de correo electrónico corporativa utilizando el servicio externo gratuito <https://formspre.io/> u otro similar.

Linters Forms

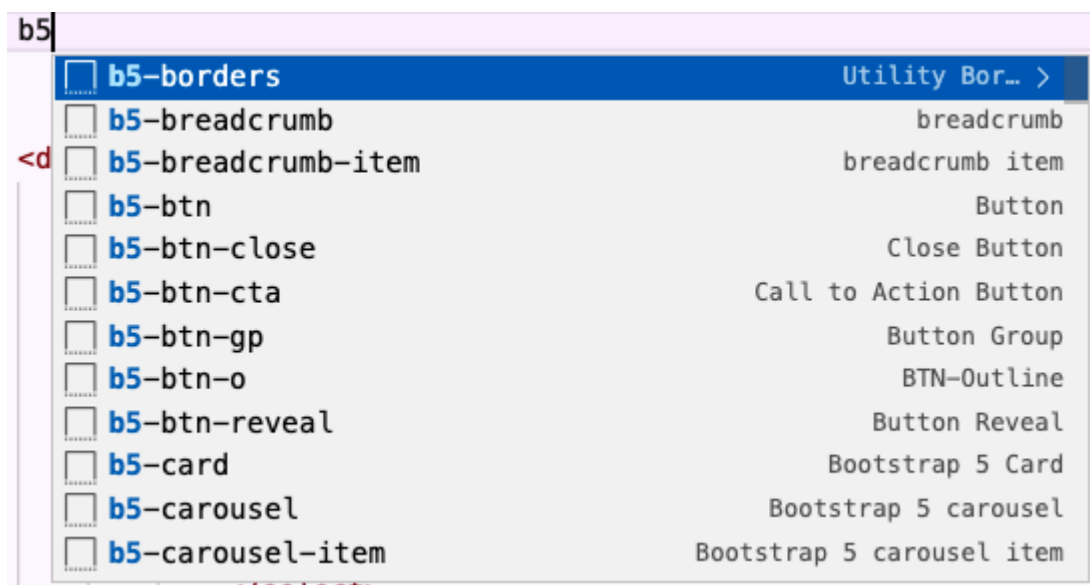
También utilizaremos <https://formlinter.com/>, herramienta de comprobación en línea gratuita que puede ayudar a identificar problemas en los formularios, como etiquetas de formulario faltantes o campos de entrada de datos mal formados.



Ejercicios

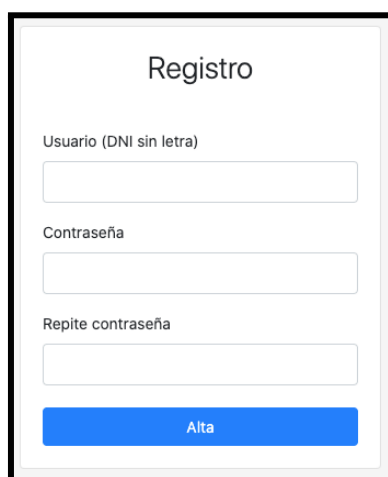
Utiliza [Bootstrap](#). En la mayoría de los ejercicios el diseño es libre.

1. Todos los ejercicios de este tema se realizan en el repositorio [b6](#)
2. **Opcional** - Instala la extensión [Bootstrap 5 Quick Snippets](#)
*b5 - * para crear la estructura de la web.*
!b5-\$ crear la web desde una plantilla



DAM - OPCIONAL ENTREGAR CON BOOTSTRAP

3. Crea una página web de nombre registro.html lo más parecida posible a la imagen utilizando Bootstrap 5:



- a. Utiliza Bootstrap 5. Un contenedor (.container) con una fila (.row) con una columna (.col, en breakpoint a partir de md utiliza 6 columnas y a partir de xl 4). Dentro de la columna utiliza un [card](#) básico.

Información: [Forms · Bootstrap v5.0.](#)

- b. El campo DNI estará compuesto exclusivamente por números con una longitud de entre 6 y 8 dígitos.
- c. La contraseña tendrá un mínimo de 8 caracteres, incluido 1 mayúscula, 1 minúscula, 1 número o carácter especial. Puedes utilizar la expresión regular: $(?=\w{8,})(?=\d)(?=\W+)(?![\.\n])(?=[A-Z])(?=[a-z]).*\$$

Nota: Puedes probar el funcionamiento de las expresiones regulares en <https://regex101.com/> o en <https://regexr.com/>

- d. El botón enviar envía los datos por el método post a uno de los servidores indicados al principio de la práctica.

- e. Utiliza el atributo placeholder
- f. El navegador no mostrará información de autocompletado en ninguno de los campos. Utiliza el atributo [autocomplete](#) de form. Observa en ["autocomplete" | Can I use... Support tables for HTML5, CSS3, etc](#) la compatibilidad con los distintos navegadores.
- g. Esta centrado en la vertical y horizontal. Como .row en Bootstrap utiliza flex, puedes utilizar las clases vh-100 justify-content-center align-items-center en .row
- h. Añade e interpreta el siguiente código CSS:

```
body {  
    background-color: #f5f5f5;  
}  
input {  
    border-radius: 6px;  
    border-color: #8080803d;  
}  
input:focus {  
    background-color: #ffffcb !important;  
}  
input:valid {  
    background-color: rgba(0, 128, 0, 0.586);  
}  
input:invalid {  
    background-color: rgba(255, 0, 0, 0.3);  
}
```

Nota: No comprueba si las password coinciden

ES CON BOOTSTRAP

- 4. Copia los ficheros registro.html y registro.css y nombrarlos como registroV2.{css, html}
 - a. Cambia el CSS para que los colores se apliquen al borde, no al fondo. El ancho del borde será de 2px.
 - b. Modifica los controles de formulario para utilizar [Floating Labels de Bootstrap 5](#).
 - c. Prueba la página en un móvil y en caso de que no se despliegue el teclado numérico para indicar el DNI utiliza el atributo [inputmode](#)

Registro

Usuario (DNI sin letra)

Contraseña

Repite contraseña

Alta

5. Prueba y explica el siguiente código HTML. Añade Bootstrap. Nombre: pruebasListas.html. ¿Se puede escribir en el control de formulario una ciudad no incluida en la lista?

```
<label for="ciudades" class="form-label">Ejemplo Datalist</label>
<input type="search" class="form-control" list="listaCiudades" id="ciudades"
placeholder="Ciudades ...">
<datalist id="listaCiudades">
  <option value="Plasencia">
  <option value="Cáceres">
  <option value="Badajoz">
  <option value="Madrid">
  <option value="Sevilla">
</datalist>
```

Crea una lista básica cerrada de selección simple y otra múltiple con los mismos datos que la datalist anterior. En la lista simple y la múltiple al servidor se enviará el código postal, no el nombre.

Envía los datos al servidor <https://httpbin.org/get> y comprueba que en el caso de la selección múltiple funciona correctamente si hay más de una ciudad seleccionada.

Ejemplo Datalist

Ciudades

Ciudades
Elige ciudad ...

Selecciona una o más ciudades

Plasencia
Cáceres
Badajoz
Madrid

Enviar

DAM terminar la datalist de formato

6. Crea una página web, `textoavoz.html`, con un formulario para trabajar con la API <https://voicerss.org/api/> y transformar un texto a audio. [Video](#)

Convertir texto a vozAPI <http://api.voicerss.org/>

Opciones ☐ Base64 ☐ SSML

Voz Diego (Male)

Velocidad del habla 2

Audio Codec AAC

Formato 22khz_8bit_mono

Texto
Buenas tardes a todos, seguimos avanzando con formularios. El siguiente paso sería completar la funcionalidad con Javascript.

Convertir (iframe) Convertir (nueva vta)

0:05 / 0:05

Requisitos:

- Para seguir probando Bootstrap, en este ejercicio utiliza <https://getbootstrap.com/docs/5.3/forms/input-group/>
- El ancho mínimo del contenedor es 400px
- La key para poder utilizar la api `api.voicerss.org`, que transforma texto a voz, es `2b1f1a6c0aa54c7fb1f94bf39674060d` Está asignada a la cuenta

antoniobp@iesvjp.es y [limitada a 350 peticiones diarias con un máximo de 100KB por petición](#)

- Todos los parámetros, a excepción de src, tendrán valores por defecto y cumplirán las restricciones impuestas por la API. No olvides que hay parámetros, como key e idioma, que también se tienen que enviar, aunque el usuario no los vea, para lo que utilizaremos **input** de tipo **hidden**. Un ejemplo de llamada a la API con método get:

```
http://api.voicerss.org/?key=2b1f1a6c0aa54c7fb1f94bf39674060d&hl=es-es&src=Hola a todos desde el IES Valle del Jerte de Plasencia
```

- Realiza validación del formulario básica.
 - El control textarea tiene un alto de 150px y utiliza el atributo autofocus ¿para qué sirve [autofocus](#)?
 - Observa la alineación de los elementos y el iframe.
 - Hay dos botones, el primero permite reproducir el audio en un iframe en la misma página, el segundo lo hace en una ventana nueva, siempre la misma. Utiliza el atributo formtarget de input.
 - El parámetro b64 y ssm1 por defecto son false, y no se permite su modificación. Comprueba en ventana nueva la url con método get y observa si se envían estos dos parámetros, en caso de que no sea así, revisa el checkbox para que se haga de forma correcta.
 - Descarga en formato MP3 el audio del texto “Hola a todos desde el IES Valle del Jerte de Plasencia”, almacena el fichero en tu proyecto y añade un enlace de descarga (utiliza el atributo **download** en <a>). Observa la diferencia entre poner y no poner el atributo download.
 - **Opcional** - Divide en dos zonas el formulario, la primera con los parámetros obligatorios y la segunda con los opcionales, a la que llamamos “Parámetros avanzados” y se podrá mostrar/ocultar.
7. Crea una página Web de nombre genopenwindow.html para generar automáticamente código Javascript para abrir ventanas en función de las características indicadas por el usuario. Añade la validación que creas necesaria. **La funcionalidad no la realizaremos. Los datos se envían a uno de los servidores de prueba. Utiliza Bootstrap (container-fluid, col-md-6 y offset-md-3).**

Generar Código Javascript

URL

Indica una URL absoluta

Opciones

- ☐ Barra de herramientas
- ☐ Barra dirección
- ☐ Barra de estado
- ☐ Barra de menús
- ☐ Barras de desplazamiento
- ☐ Permitir ajustar el tamaño

Width

Height

Generar

Código generado

Probar código

8. Crea un formulario de nombre genTablas.html que permita la creación de tablas online. Tendrá los controles que se ven en la imagen.

Generación automática de tablas HTML

Nº Filas	Nº Columnas
<input type="text" value="2"/>	<input type="text" value="2"/>
<input type="checkbox"/> 1ª Fila es encabezado	<input type="checkbox"/> 1ª Columna es encabezado
Id de la tabla	Clases de la tabla
<input type="text" value="pruebas"/>	<input type="text" value="table table-hover table-bordered"/>
<input type="button" value="Generar"/>	

Código

```
<table class="table table-hover table-bordered" id="pruebas">
<tbody>
<tr>
<td>[1,1]</td>
<td>[1,2]</td>
</tr>
<tr>
<td>[2,1]</td>
<td>[2,2]</td>
</tr>
</tbody>
</table>
```

Preview

Características de los campos:

- Filas y columnas: obligatorios (requeridos), números enteros, mínimo 1, máximo 100.
- Nombre de la clase/s separadas por espacios. Valor por defecto table table-hover table-bordered
- Nombre de la id será válido según descrito en HTML5. Puedes utilizar la expresión regular `^[\\S]+$`

Ver [vídeo](#). No añadiremos la funcionalidad, los datos se envían a uno de los servidores.

9. Haz un formulario de nombre pruebas.html para probar los diferentes controles. Tiene que ser similar al mostrado. Utiliza Bootstrap.

Pruebas de controles de formulario: ☐ Solo Lectura ☐ Deshabilitado

Ejemplos de input: ☐ Solo Lectura ☐ Deshabilitado

Nombre

Teléfono

Página web

Correo electrónico

Contraseña

Fecha

Mes

Semana

Hora

Fecha y hora

Edad

Rango

Color

Radio ☒ Uno ☐ Dos ☐ Tres

Checkbox ☐ Casilla

Fichero

Ejemplo de textarea: ☐ Solo Lectura ☐ Deshabilitado

Escriba una descripción:

Ejemplo de input con datalist: ☐ Solo Lectura ☐ Deshabilitado

Escriba su navegador:

Ejemplos de select: ☐ Solo Lectura ☐ Deshabilitado

Elija un dinosaurio:

Elija una opción:

Elija una o varias opciones:

Ejemplo de progress:

Ejemplo de meter:

- Todo el contenido encerrado en un [container](#).
- Alinea las etiquetas y los controles en dos columnas en dispositivos sm (tablet) y superiores, en el resto en 1 columna. Las filas pares aparecerán con un color de fondo distinto a las impares.
- Incluye un ejemplo de cada uno de los controles estudiados: input - todos los tipos menos submit y reset, textarea, input con datalist, select con optgroup selección simple y múltiple, progress, meter y button (pero no submit ni reset). Todos tendrán su etiqueta (label) asociada.
- Agrupar todos los controles utilizando fieldset y etiqueta "Pruebas de controles de formulario". Dentro de este, cada tipo de control se

agrupará en otro fieldset con etiqueta indicando el tipo “Ejemplos de input”, “Ejemplo textarea”,

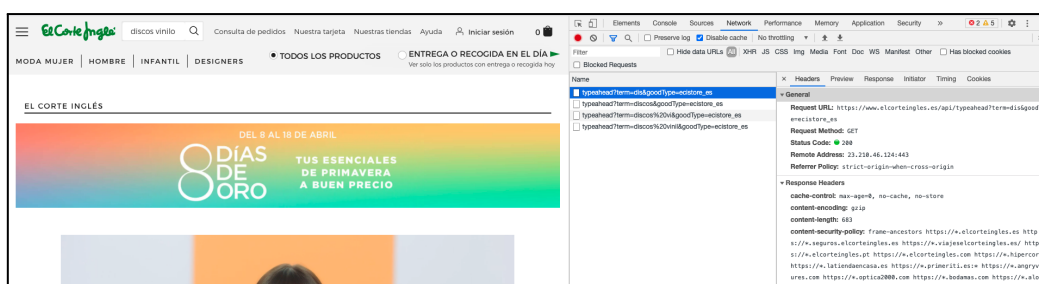
- e. Indica valores por defecto en alguna lista y en algún input de tipo checkbox y radio.
- f. Utiliza el atributo placeholder en alguno de los controles.
- g. Añade atributos para hacer una correcta validación de datos (max, min, maxlength, pattern, etc.) , marca algunos como requeridos.
- h. ¿Podemos utilizar min y max en campos de fechas y horas?
- i. Añade fuera de los fieldset dos botones de envío (submit) y uno de reset:
 - i. Uno de los botones submit envía los datos al servidor indicado en el atributo action del formulario (form) y el otro a un servidor distinto, utiliza el atributo [formaction](#). El método de envío en ambos casos será el mismo.
 - ii. ¿Qué método y codificación recomiendas?
 - iii. ¿Qué ocurre cuando pulsamos el botón reset?
 - iv. ¿Qué ocurre cuando pulsamos el botón de envío y algún campo requerido no ha sido rellenado?
 - v. ¿Qué ocurre cuando pulsamos el botón de envío y algún campo no requerido ha sido rellenado con un valor incorrecto?
 - vi. Prueba a seleccionar más de un elemento en las listas de selección múltiple y en los checkbox y comprueba si el servidor los ha recibido correctamente, prueba los dos servidores.
- j. Añade dos checkbox en la etiqueta de cada uno de los fieldset, uno con la etiqueta “[Solo Lectura](#)” y otro con “[Deshabilitado](#)”. Los datos de estos checkbox no se enviarán al servidor.
- k. Prueba a enviar los datos del formulario al servidor con atributos deshabilitados y otros en solo lectura ¿qué datos se envían?

disabled	button; input; optgroup; option; select; textarea; form-associated custom elements	Whether the form control is disabled
disabled	fieldset	Whether the descendant form controls, except any inside legend, are disabled
disabled	link	Whether the link is disabled
readonly	input; textarea	Whether to allow the value to be edited by the user

- l. **Aplica estilos (fondo, color, bordes, fuente, tamaño, ...) diferentes a los controles en función de su estado de validación, de si están rellenos, de si son requeridos, de si son de solo lectura, etc. → [CSS específico de formularios](#).**

10. Desde las herramientas de inspección, pestaña Network, observa cómo se realizan los envíos al servidor cuando completamos datos en el buscador de El Corte Inglés.

Nota: antes de realizar búsquedas limpia el listado de Network



Identifica el método de envío y a dónde se envía (action).

¿Qué tipo de respuesta devuelve el servidor al pulsar el botón envío, es decir, al pulsar el icono lupa?

Observa cómo se ha construido la URL y cómo se envían los datos. Cambia el atributo action para que apunte a uno de los servidores de prueba para ver los datos que recibe el servidor.

11. Crea un formulario de nombre boe.html para facilitar la consulta del BOE por fecha. El resultado se abrirá en ventana nueva, pero siempre en la misma.

Consulta del BOE

Fecha:

12. Modifica la página boe.html para que incluya un segundo formulario que permita una búsqueda avanzada. Los campos a incluir son los mismos que ofrecen en la web del BOE, <https://www.boe.es/buscar/boe.php>, a excepción de los "Y". Utiliza estilos específicos de formulario y haz una validación básica basada en HTML. *Recomendado empezar copiando el html del formulario una vez quitados los controles Y desde el inspector.*

Sección

☒ Sección I
☒ Sección II
☒ Sección III
☒ Sección IV
☒ Sección V
☒ Sección TC

Título

protección datos

Departamento

Texto

Núm. de boletín

n/aaaa

Núm. oficial

n/aaaa o DPTO/n/aaaa

Fecha de publicación de

dd/mm/aaaa

a

dd/mm/aaaa

Documentos por página

50

Ordenados por fecha de publicación

descendente

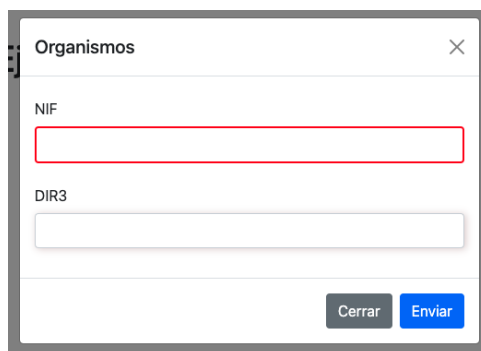
Buscar

Limpiar

13. Crea un formulario Web sencillo de nombre `consultaOrganismos.html` basado en el [esquema XSD de consulta de organismos](#) de [Fondos 2020](#). Utiliza [CSS específico de formularios](#) y [modal Bootstrap](#).

Ejercicios formulario básico XSD

Consulta Organismos



Organismos

NIF

DIR3

Cerrar Enviar

14. Crea un formulario de nombre `alta.html` para crear un perfil, utiliza los datos que ves en la imagen (recuerda, ejercicio perfiles propuesto en [Bloque 5 - XSD, XPATH, XQUERY, CSS en XML y XSLT](#)). Aplica estilos a la página Web **utilizando los [selectores explicados](#)**.



Trinidad Iglesias

615-634-609

Trinidad Iglesias

DNI 21424642-G

Calle de Toledo, 7344
Arrecife - Asturias

28 años
trinidad.iglesias@example.com

Trinidad Iglesias

615-634-609

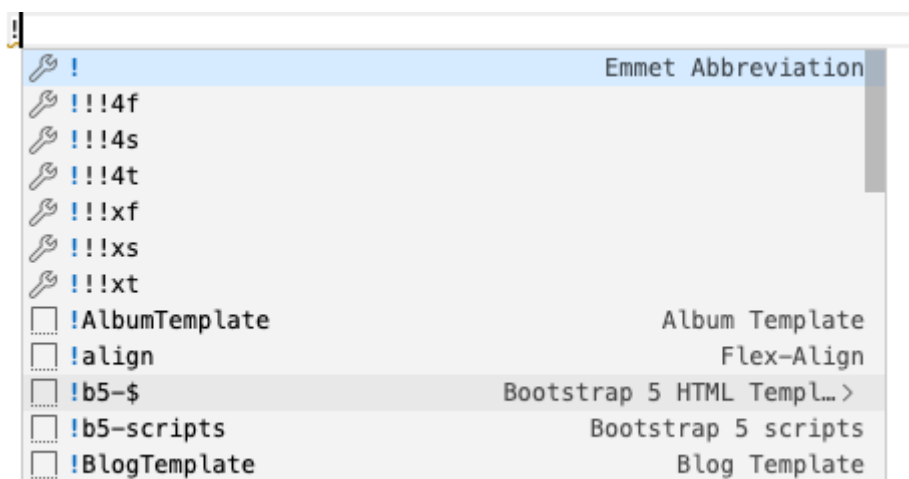
- El formulario se enviará por el método `get` a uno de los servidores.
 - Impide el envío del formulario a menores de 14 años.
 - Permite elegir entre las distintas provincias, utiliza una lista de selección simple con las provincias agrupadas por comunidad autónoma. [Lista estándar de provincias](#) y [en formato HTML](#). *Nota: convendría utilizar JS para mostrar las provincias en función de la comunidad autónoma seleccionada.*
 - La fotografía solo admite formato `.jpg` y `.png` (atributo `accept=".jpg, .png"` de `input`), permite subir varias, atributo `multiple` en `input file`
15. Crea un formulario, `perfilesaleatorios.html`, para obtener en diferentes formatos (xml, json, pretty json, etc) datos de perfiles generados aleatoriamente obtenidos de <https://randomuser.me/documentation#intro>. El resultado se abrirá en una ventana nueva, siempre en la misma.
16. Crea un formulario Web de nombre `altaOrganismo.html` basado en el [esquema XSD de alta de organismos](#) de [Fondos 2020](#), sin incluir el elemento **roles**. En los

elementos enumerados que tengan anotaciones, crea un datalist de forma que el value sea el valor válido según el XSD y la información sea la anotación. *Nota: con JS podríamos utilizar [Bootstrap con validación de formularios](#).*

17. Pide al profesor que publique el proyecto y prueba el linter explicado en el apartado "[Linter Forms](#)" con cada formulario.
18. Crea la página index.html del proyecto.
19. Completa el **índice principal** de tu sitio web con un enlace a este proyecto.

Ejercicios Opcionales

20. Copia el ejercicio genopenwindow.html, nombre de la copia genopenwindowV2.html, y utiliza alguno de los "Toggle switches" propuestos en [Toggle switches made with CSS and HTML](#), como por ejemplo: [Switch by JulioCodesSM](#) Color de fondo: #eee
21. Utiliza una de las [herramientas online](#) explicadas para generar formularios.
22. Duplica la página registroV2.html, nombre del duplicado registroV3.html, con las siguientes restricciones:
 - a. En la página únicamente aparecerá un botón con el texto Registro.
 - b. Al pulsar el botón se abrirá el formulario de alta en un <dialog>
 - c. El formulario de registro incluye un botón con el texto "Cancelar". Al pulsarlo se cerrará el dialog. Utiliza JS.
 - d. La funcionalidad del <form> no se verá alterada (no puedes utilizar method="dialog")
23. Instala la extensión [Bootstrap 5 y Font Awesome](#) para VSCode.
!b5-\$ para crear la estructura de la web.



24. Crea un formulario Web de nombre autorizacion.html basado en el [esquema XSD de autorización para la puesta en servicio de estaciones radioeléctricas \(fichero EsquemaPSI.xsd\)](#) de la Secretaría de Estado de Telecomunicaciones e Infraestructuras Digitales.
25. Crea un formulario Web de nombre horarioDia.html para añadir un elemento \\Horario\\Dia siguiendo las restricciones del XSD de los horarios. **Utiliza CSS específico de formularios.**
26. Busca un XSD de una empresa u organismo público y crea un formulario Web que facilite la entrada de datos cumpliendo las restricciones indicadas en el XSD.
27. Busca y explica alguna extensión de VSCode relacionada con Bootstrap
28. [Website Accessibility Checker - Free Instant Accessibility Check](#)