

UT 4

Integración de contenido interactivo

Diseño de interfaces web
2024/2025



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Autor: Fran Gómez
2024/2025

1. Elementos interactivos
 - a. ¿Qué son?
 - b. Aplicar interactividad en un documento HTML
 - i. Etiqueta details
 - ii. Propiedad summary
 - iii. Etiqueta dialog
 - iv. Botones
 - v. Eventos
 - vi. Formularios
 - c. Herramientas
2. Configuración de navegadores
3. Librerías Javascript
 - a. Bootstrap
 - b. JQuery
 - i. Selectores
 - ii. Filtros
 - iii. Eventos
 - iv. Añadir/Eliminar/Recorrer elementos
 - v. Comandos (callbacks)
4. Reproducción de elementos multimedia e interactivos

- Permiten la comunicación entre la web y el usuario
- Web dinámica → los elementos cambian cuando el usuario actúa sobre ellos sin recargar la página

♦ Diferencias entre Web Estática y Web Dinámica

Característica	Web Estática	Web Dinámica
Contenido	No cambia sin editar el código	Se genera en tiempo real
Interactividad	Limitada (solo enlaces y estilos)	Alta (formularios, chats, animaciones, etc.)
Lenguajes usados	HTML, CSS	HTML, CSS, JavaScript, PHP, Python, Node.js, etc.
Base de datos	No usa	Puede conectarse a bases de datos
Ejemplo	Página informativa	Tiendas online, redes sociales, blogs con comentarios

Proceso de carga de contenido dinámico:

1. El usuario accede a la web y solicita información.
2. El servidor procesa la petición y consulta la base de datos.
3. La respuesta se envía al navegador en formato HTML, JSON o XML.
4. JavaScript actualiza el contenido sin recargar la página.

♦ Ejemplos de Webs Dinámicas

- ✓ **Redes sociales** (Facebook, Twitter, Instagram) → El contenido cambia según el usuario.
- ✓ **Tiendas online** (Amazon, MercadoLibre) → Los productos y precios se actualizan en tiempo real.
- ✓ **Blogs y noticias** (WordPress, Medium) → Se pueden agregar nuevos artículos sin modificar el código.
- ✓ **Plataformas de streaming** (Netflix, YouTube) → Recomiendan contenido basado en tus preferencias.
- ✓ **Aplicaciones web** (Gmail, Google Docs) → Permiten interacción en tiempo real sin recargar la página.



ELEMENTOS DE UNA PÁGINA WEB

1. Nombre de Dominio
2. Diseño Fácil de Navegar
3. Páginas internas
4. Logotipo
5. Menús
6. Textos Persuasivos
7. Imágenes complementarias
8. Datos de Contacto
9. Botones de Redes Sociales
10. Formulario de Contacto
11. Llamadas a la acción
12. Mapa de ubicación
13. Páginas complementarias
14. Herramientas para estadísticas

Ejemplos

Animaciones y Transiciones

Formularios (autocompletado, validaciones en tiempo real, mostrar/ocultar campos)

Gráficos y Visualización de Datos (dashboards, mapas, infografías...)

Contenido Multimedia Interactivo (videos con hotspots, Genially, vistas 3D de productos)

Aplicaciones Web Progresivas (PWA) (offline, cámara, micrófono, geolocalización)

Chatbots y Asistentes Virtuales

Sistemas Drag & Drop (Trello...)

Pon una web de ejemplo de cada uno de los elementos interactivos anteriores.

De por sí los elementos HTML no tienen interactividad, hay que dársela:

- CSS → Pseudoclases
- Javascript → Eventos
- HTML5 sí ofrece algunos con interactividad propia:
 - Etiqueta <details>
 - Propiedad <summary>
 - Etiqueta <dialog>

Etiqueta <details> y propiedad <summary>



HTML

CSS

```
1 <details>
2   <summary>Details</summary>
3   Something small enough to escape casual
4   notice.
5 </details>
```

OUTPUT

▼ Details

Something small enough to escape casual notice.

Ejercicio 1:

Crea un HTML con un párrafo conteniendo un texto cualquiera.

Debajo introduce unos detalles bajo el título “Mostrar más” que de inicio aparecerán visibles, pero cuando se pulse en “Mostrar más” se ocultarán.

¿Serías capaz de aplicarle estilo al título y los detalles para se vea así?:

▼ Mostrar más
Bla bla bla....

Se trata de un **diálogo modal** → No requiere interactuar con él.

The dialog element

This is some text.

This is some text.

This is some text.

This is some text.

This is an open dialog window

Ejercicio 2:

Crea un diálogo modal como el siguiente, que aparezca al iniciarse la página y desaparezca al pulsar en un botón contenido en el mismo.



```
<dialog open>
  <p>Greetings, one and all!</p>
  <form method="dialog">
    <button>OK</button>
  </form>
</dialog>
```

Ejercicio 3:

Al desplegar un <details> que el contenido se muestre lentamente.

Your wooclap poll will be displayed here



Install the **Chrome** or
Firefox extension



Make sure you are in
presentation mode

wooclap

Básicamente 2 formas:

- `iframe`
- `script`

HTML	CSS
<pre>1 <iframe 2 id="inlineFrameExample" 3 title="Inline Frame Example" 4 width="300" 5 height="200" 6 7 src="https://www.openstreetmap.org/export/embed. 8 html? 9 bbox=-0.004017949104309083%2C51.47612752641776%2 10 C0.00030577182769775396%2C51.478569861898606&lay 11 er=mapnik"> 12 </iframe> 13</pre>	



```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```


<!-- Video interactivo de YouTube -->

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/VIDEO_ID"
frameborder="0" allowfullscreen></iframe>
```

<!-- Mapa interactivo de Google Maps -->

```
<iframe src="https://www.google.com/maps/embed?pb=MAPA_ID"
width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe>
```

```
<!-- Widget de Twitter (X) -->
```

```
<a class="twitter-timeline" href="https://twitter.com/usuario">Tweets  
de Usuario</a>
```

```
<script async src="https://platform.twitter.com/widgets.js"></script>
```

```
<!-- Código de un formulario de Google Forms -->
```

```
<iframe  
src="https://docs.google.com/forms/d/e/FORM_ID/viewform?embedded=true"  
width="640" height="800" frameborder="0" marginheight="0"  
marginwidth="0">Cargando...</iframe>
```

```
<!-- Integración de API con JavaScript -->
```

```
<div id="contenido"></div>
```

```
<script>
```

```
  fetch('https://api.example.com/datos')
```

```
    .then(response => response.json())
```

```
    .then(data => {
```

```
      document.getElementById('contenido').innerHTML = `

${data.mensaje}</p>`;


```

```
    });
```

```
</script>
```



Fuente: OpenReplay

Librerías JavaScript para Interactividad



2º DAW
DIWEB
UT4 - Interactivo

```
<canvas id="grafico"></canvas>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>

  var ctx = document.getElementById('grafico').getContext('2d');

  var chart = new Chart(ctx, {

    type: 'bar',

    data: {

      labels: ['A', 'B', 'C'],

      datasets: [{ label: 'Valores', data: [10, 20, 30], backgroundColor: 'blue' }]

    }

  });

</script>
```

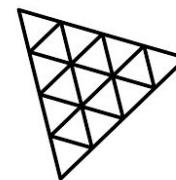


Chart.js



WEBCOMPONENTS



Ejercicio 4



2º DAW
DIWEB

UT4 - Interactivo

Buscar webs con contenido interactivo e identificar cómo está integrado ese contenido.

3 tipos:

- Submit
- Reset
- Button

CLICK ME FOR NO REASON!

Name

RESET THE FORM!

SUBMIT (DISABLED)

Ejercicio 5: crea dos botones como los de la figura, aplicando estilos sólo a través de clases.



Lanzan acciones

Hay multitud de ellos y se clasifican según quién los lanza (eventos de ratón, de teclado, de la página...). Ejemplos:

Evento	Descripción
onclick	Pulsar y soltar el ratón
ondblclick	Pulsar dos veces seguidas con el ratón
onfocus	Un elemento obtiene el foco
onkeydown	Pulsar una tecla y no soltarla

¿Cómo prevenir por CSS que se aplique un evento? → `pointer-events: none;`

Nuevos atributos:

Nombre:

Correo Electrónico:

URL:

Fecha:

Tiempo:

Fecha y hora de nacimiento:

Mes:

Semana:

Número (min -10, max 10):

Intervalo (min 0, max 10):

Teléfono:

Término de búsqueda:

Color Favorito:

Sistema operativo:

Que lenguaje de programación web conoces? ☒ Html5 ☒ Css3 ☐ JavaScript ☐ Php ☐ Python

Sabes utilizar Python? ☒ Si ☐ No

Comentarios:

```
input[required]{
  border-color:blue;
  box-shadow:0px 0px 5px blue;
}
input:invalid{
  border-color:red;
  box-shadow:0px 0px 5px blue;
}
input:valid{
  border-color:green;
  box-shadow:0px 0px 5px green;
}
```

- autofocus
- min
- max
- pattern
- placeholder
- required
- step

Elementos usados en los formularios:

- [<button>](#)
- [<datalist>](#)
- [<fieldset>](#)
- [<input>](#)
- [<label>](#)
- [<legend>](#)
- [<meter>](#)
- [<optgroup>](#)
- [<option>](#)
- [<output>](#)
- [<progress>](#)
- [<select>](#)
- [<textarea>](#)

Formularios: de Registro



INSCRIPCIÓN EN CSS3

Nombre

Contraseña

[Ver más opciones](#)

o [cancelar](#)

INSCRIPCIÓN EN CSS3

Nombre

Contraseña

[Ocultar las opciones](#)

Código postal

Ciudad

País

o [cancelar](#)

INSCRIPCIÓN EN CSS3

Nombre

Contraseña

[Ver más opciones](#)

o [cancelar](#)

Forma de pago

Los campos obligatorios van seguidos de *.

Información de contacto

Título

- ☒ Rey
- ☐ Reina
- ☐ Bufón

Nombre: *

Correo electrónico: *

Contraseña: *

Información de pago

Tipo de tarjeta:

Número de tarjeta: *

Fecha de caducidad: * *el formato mm/aa*

Ejercicio:

Realiza un formulario HTML5 y CSS3 a tu elección que incluya al menos:

- 3 elementos de estructuración (fieldset, legend, label...)
- 5 controles distintos (button, select, input...)
- 5 elementos con estilos (CSS3)
- 2 validaciones (combinando HTML5 Y CSS3)

Continuación:

- Añade algún tipo de interactividad con el usuario (ej: que el color cambie al pasar por encima el ratón, que el botón realice alguna acción...)
- Maneja la tabulación (tabindex), teclas de acceso (accesskey) y el deshabilitado de campos (readonly y disabled)

Your wooclap poll will be displayed here



Install the **Chrome** or
Firefox extension



Make sure you are in
presentation mode

wooclap

- Hay código CSS o Javascript que no es compatible con todos los navegadores, debido a que:
 - El navegador es una versión antigua
 - El código incluye funcionalidades ([propiedades](#), [APIs](#), tags ...) muy modernas
- User Agent: información respecto la aplicación (browser) cliente.
 - Podemos crear un código distinto en función del navegador detectado. Tanto en JS como en CSS. Ejemplo: <https://codepen.io/samiah/pen/NgQKMb>
- @Support
 - <https://developer.mozilla.org/en-US/docs/Web/CSS/@supports#examples>
- CSS browser prefixes ⇒ no son estándar, son propias de cada navegador.

```
-webkit-transition: all 4s ease;  
-moz-transition: all 4s ease;  
-ms-transition: all 4s ease;  
-o-transition: all 4s ease;  
transition: all 4s ease;
```

- **Validadores:** <https://caniuse.com/>, <https://html5test.com/>, [W3C validator](#)
- **Linters:** [Dirty Markup](#) (HTML, CSS, JavaScript), [CSS Lint](#) (CSS)
 - Ejercicio: busca plugins de linters para Vscode
- **Browser developer tools**

**CSS Validation Service**
Check Cascading Style Sheets (CSS) and (X)HTML documents

By URI

By file upload

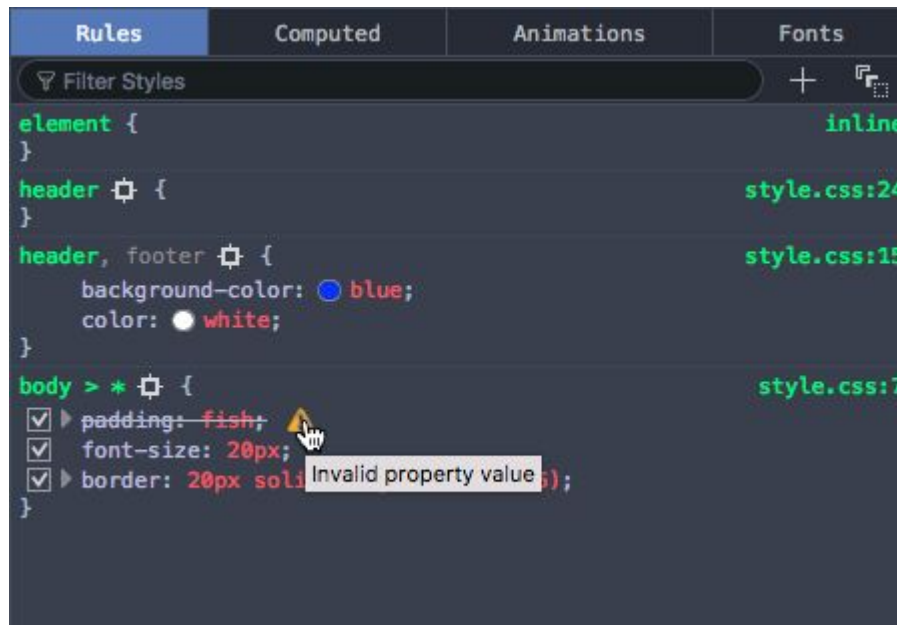
By direct input

Validate by URI

Enter the URI of a document (HTML with CSS or CSS only) you would like validate

Address:

[▶ More Options](#)



```
Rules    Computed    Animations    Fonts
Filter Styles
element { inline
}
header { style.css:24
}
header, footer { style.css:15
  background-color: blue;
  color: white;
}
body > * { style.css:7
  padding: fish;
  font-size: 20px;
  border: 20px solid ;
}
```

Your wooclap poll will be displayed here

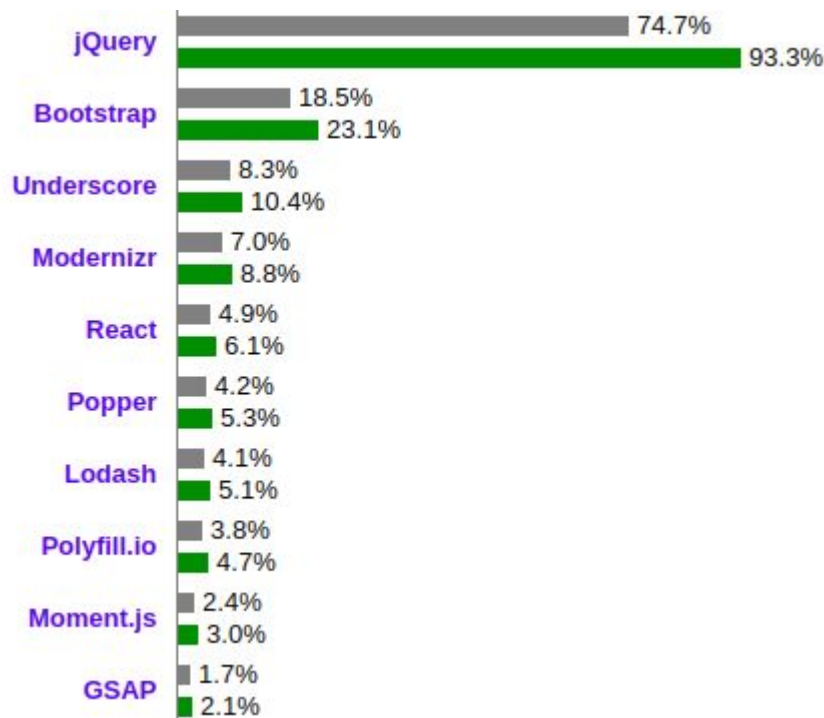


Install the **Chrome** or
Firefox extension



Make sure you are in
presentation mode

wooclap



Ejercicio: tabla comparativa de los principales

¿Cuáles son los más adecuados para el diseño de interfaces?

<https://www.eniun.com/jquery-integracion-contenido-interactivo/>



<http://learn.jquery.com/jquery-ui/getting-started/>



Imagen: Wikimedia commons

<https://www.eniun.com/frameworks-diseno-web-responsive/>

<https://www.eniun.com/tutorial-bootstrap-5/>



- Podemos manipular y personalizar los elementos multimedia usando Javascript. Ej:
<https://www.eniun.com/control-reproduccion-video-audio-javascript/>
- También podemos añadir elementos multimedia usando librerías Javascript. Ej:
<https://videojs.com/getting-started>

- Ejercicios
- Examen
- Proyecto



Fin