

Memoria Proyecto

II Jornadas de Informática

HTML



CSS



ÍNDICE

Sección		Página
1. Explicación del uso de las etiquetas HTML	-----	3
1.1 Código HTML común a las tres páginas	-----	3
1.2 Uso de etiquetas en index.html	-----	4
1.3 Uso de etiquetas en programacion.html	-----	6
1.4 Uso de etiquetas en inscripcion.html	-----	8
2. Explicación del uso CSS utilizado	-----	10
2.1 Estilos que se aplican a las parte comunes de las tres páginas	-----	10
2.2 Estilos empleados en index.html	-----	14
2.3 Estilos empleados en programacion.html	-----	20
2.4 Estilos empleados en inscripcion.html	-----	24
3. Capturas íntegras mediante impresión de página en pdf del resultado en mi monitor:	-----	28
4. Validaciones W3C:	-----	31

1. Explicación del uso de las etiquetas HTML

Todas las etiquetas empleadas en las páginas, se han tratado de forma semántica atendiendo a las imágenes de muestra, tratando de discernir el uso que se hubiera podido efectuar por contexto, tipología de letra... etc.

Se ha evitado en la medida de lo posible el uso de contenedores <div> sin significado semántico a pesar de que facilitaría mucho y en muchas ocasiones el trabajo posterior en CSS pero he tratado de trabajar con las etiquetas semánticas para el posicionamiento y estilos mediante CSS. A pesar de ello, se han incluido dos <div>, uno en el <header> (repetido en las tres páginas) y otro en el index.html en la zona central ya que el uso de otra etiqueta semántica para abrazar esa zona requería el uso de un encabezado o haberlo ocultado con CSS y no me parecía una buena práctica.

El resto se ha empleado, como ya he indicado, etiquetas con significado semántico y se ha trabajado sobre ellas todo el CSS.

1.1 Código HTML común a las tres páginas:

<header>

```
<header>
  
  <div class="titular">
    <h1>II Jornadas de Informática</h1>
    <strong>Valencia, 3 y 4 de Abril de 2021</strong>
  </div>
</header>
```

Utilizada para la cabecera de la página, en ella he incluido la imagen proporcionada a modo de logotipo con la etiqueta , atributo src con la URL relativa de la imagen, y añadiendo el atributo alt con una breve descripción en caso de no cargar la imagen.

Dentro del <header> he utilizado un <div> para usarlo de contenedor para el <h1> que es el título principal empleado en todas las páginas creadas, incluir el <h1> en el <header> está considerado de forma correcta.

La etiqueta es usada dado el carácter de interés que tenía el titular de la ubicación y las fechas de realización estando además en el <header>.

<nav>

```
<nav>
  <ul>
    <li><a href="/index.html">Inicio</a></li>
    <li><a href="/programacion.html">Programación</a></li>
    <li><a href="/inscripcion.html">Registro a las jornadas</a></li>
  </ul>
</nav>
```

Etiqueta <nav> para nuestro menú de navegación, encerrado dentro de estas etiquetas he empleado un que encierra tres etiquetas y a su vez contienen un <a> cada uno, para poder enviar al usuario de una página a otra desde dicho menú usando URL relativas.

<footer>

```
<footer>
  <p>Actividad realizada por José Vicente Falcó Milla, para la asignatura de Lenguajes de Marcas</p>
  <p>Primer trimestre 2021/2022</p>
</footer>
```

Etiqueta <footer> para el pie de página de cada una de ellas. He incluido dos elementos <p> para los textos.

1.2 Uso de etiquetas en index.html

<main>

```
<main>
  <article class="video">
    <h2>II Jornadas de Informática</h2>
    <p>Las Jornadas de la Informática son un espacio de divulgación de temas
    actuales, Nuevos ámbitos de aplicación de las Tecnologías de la Informa
    <video src="/media/video_jornadas.mp4" controls></video>
  </article>
  <div class="realiza">
    <section class="donde">
      <h2>¿Dónde se realiza?</h2><br>
      <address>
        <p>Auditorio</p><br>
        <p>Plaza de la Misericordia, Valencia, 46115 Tel.</p><br>
      </address>
      <p><a href="tel:+34935555555">93 555 55 55</a></p><br>
      <p><a href="mailto:info@jornadas.org">info@jornadas.org</a></p><br>
      <p><a href=""><svg aria-hidden="true" focusable="false" data-prefix="fa" data-kind="parent" data-fas>
      512"><path fill="currentColor" d="M459.4 151.7c.325 4.548 .325 9.1
      57 130.3-44.83-46.13-.975-84.79-31.19-98.11-72.77 6.498 .974 12
      32-28.26-18.84-46.78-51.01-46.78-87.39 0-19.49 5.197-37.36 14.29
      76.67 33.14 23.72-4.548 46.46-13.32 66.6-25.34-7.798 24.37-24.37
      <a href=""><svg aria-hidden="true" focusable="false" data-prefix="fa" data-kind="parent" data-fas>
      512"><path fill="currentColor" d="M279.1 288.14,22-92.66h-88.91
      2V288z"></path></svg></a>
      </p>
    </section>
    <section class="cuando">
      <h2>¿Cuándo se realiza?</h2><br>
      <p><em>Sábado, 3 de abril, de 9:00 a 20h</em></p><br>
      <p><em>Domingo, 4 de abril, de 9:00h a 18:30h</em></p><br>
      <p><strong>Entrada libre con <a href="/inscripcion.html">inscrip
    </section>
  </div>
  <section class="eventos">
    <h2>Eventos</h2>
    <p>Trailer s
    <p>
</main>
```

Dentro del cuerpo principal de la página, en el <main> he ubicado un primer <article> con su necesario <h2>, un <p> y la etiqueta <video> con el atributo controls para poder tener los controles visualmente y accesibles en el vídeo. Así mismo el atributo src con una URL relativa al vídeo.

Bajo del <article> y para seguir las buenas prácticas y viendo la imagen de muestra del PDF (zona verde), he optado por el elemento <div> al que le he añadido la propiedad class para encerrar dos <section>, uno para el “dónde” con su respectivo atributo class en la etiqueta y otro para el “cuando” con su class, hacer otro tipo de encapsulado con etiquetas con significado semántico requeriría de un <h2> vacío en la etiqueta que hiciera de contenedor por requerimientos de HTML5.

Dentro del primer <section> he usado LA etiqueta <address> para englobar la dirección, cada línea de la dirección la he puesto dentro de un <p>. He usado en algunos puntos concretos de los html etiquetas
 ya que me proporcionaban un salto de línea y un espaciado justo como para poder dejarlo visiblemente idéntico a muestra, evitando el uso de CSS innecesario.

Tras el <address> he puesto un par de etiquetas <a>, una con un href=”tel:” para el teléfono y que pueda hacer la funcionalidad para la que está destinado, y (a sabiendas que según se nos indica en los apuntes, no se considera adecuado en la actualidad por el span y recopilación de datos que se puede hacer de este tipo de links) otra etiqueta <a> con un href=”mailto:”, ambos englobados dentro de un elemento <p>, con el fin de hacer uso de sus propiedades y características predefinidas de bloque ya que entiendo que a nivel de semántica no afecta a la estructura de la web.

Tras el anterior elemento descrito, se ha hecho uso de código <svg> proporcionado por la web indicada en el enunciado fontawesome.com para los iconos de las redes sociales. Se han encapsulado cada uno de los <svg> de forma separada por un elemento <a> para poder hacer uso de esos iconos a modo de link. No se ha indicado nada en el atributo href como se nos indica.

Segundo <section> con su atributo class para posterior uso. He incluido elementos para las dos primeras frases de datos que nos dan las fechas, por remarcar dicha información y darle la importancia que creo que requieren. El último dato, lo he envuelto todo con etiquetas pues considero que es un dato resaltado en la imagen de muestra por su interés y no tanto por los estilos, dentro de dicha etiqueta semántica, he incluido el link con una etiqueta <a> dándole a su href=”” una URL relativa a nuestra página inscripcion.html, pues he supuesto que debía enviarnos allí.

Finalizado el <div> anterior abro otro <section> para la parte de Eventos. Envuelvo cada imagen con su atributo src relativo y su alt descriptivo, dentro de un <p> para hacer uso más tarde al ser un elemento de bloque que engloba a nuestras imágenes.

1.3 Uso de etiquetas en programacion.html

```
<main>
  <section class="tabla-sec">
    <h2>Programa de las II Jorna
    <table>
      <thead>
        <tr>
          <th colspan="5">Todo
        </tr>
      </thead>
      <tbody>
        <tr>
          <th>Fecha</th>
          <th>Taller</th>
          <th>Ponente</th>
          <th>Horario</th>
          <th>Ubicación</th>
        </tr>
```

```
<tr>
  <td rowspan="4">2 de Abri
  <td>Computación Cuántica:
  <td>Javier García</td>
  <td><svg aria-hidden="true
  fill="currentColor" d="M25
  16,0,0,1,16,16V256158,42.5
  <td><svg aria-hidden="true
  viewBox="0 0 384 512"><pat
  13.773-39.464 0zM192 272c4
</tr>
<tr>
  <td>Animación 3D en la inc
  <td>Vanessa Pérez</td>
  <td><svg aria-hidden="true
  fill="currentColor" d="M25
  16,0,0,1,16,16V256158,42.5
  <td><svg aria-hidden="true
  viewBox="0 0 384 512"><pat
  13.773-39.464 0zM192 272c4
</tr>
```

```
<tr>
  <td>Taller sobre análisis
  <td>Dionisio Vidal</td>
  <td><svg aria-hidden="tr
  fill="currentColor" d="M
  16,0,0,1,16,16V256158,42
  <td><svg aria-hidden="tr
  viewBox="0 0 384 512"><p
  13.773-39.464 0zM192 272
</tr>
<tr>
  <td class="subrayado">Me
  <td class="subrayado">Pa
  <td class="subrayado"><s
  viewBox="0 0 512 512"><p
  23V112a16,16,0,0,1,16-16
  <td class="subrayado"><s
  2000/svg" viewBox="0 0 3
  774-29.93 13.773-39.464
</tr>
```

```
<tr>
  <td rowspan="3">4 de
  <td>Realidad virtual
  <td>Óscar Sapena</td>
  <td><svg aria-hidden
  fill="currentColor"
  16,0,0,1,16,16V25615
  <td><svg aria-hidden
  viewBox="0 0 384 512
  13.773-39.464 0zM192
</tr>
<tr>
  <td>La predicción de
  <td>Nuria Medina</td>
  <td><svg aria-hidden
  fill="currentColor"
  16,0,0,1,16,16V25615
  <td><svg aria-hidden
  viewBox="0 0 384 512
  13.773-39.464 0zM192
</tr>
```

```
<tr>
  <td>First Lego Le
  <td>Pedro Sosa</td>
  <td><svg aria-hid
  fill="currentColo
  16,0,0,1,16,16V25
  <td><svg aria-hid
  viewBox="0 0 384
  13.773-39.464 0zM
</tr>
</tbody>
<tfoot>
  <tr>
    <td colspan="5">L
  </tr>
</tfoot>
</table>
```

He usado un `<section>` que abraza y da título `<h2>` a toda la tabla que se coloca tras el `<h2>`. En la tabla, he empleado todas las etiquetas y campos que tienen pues así me ha parecido la imagen de muestra, así pues, la estructura es la de `<table>` que abraza a un `<thead>` `<tbody>` `<tfoot>` todas ellas con sus respectivas etiquetas de cierre y encerrando en cada una de ellas las `<th>`, `<tr>` o `<td>` que correspondieran.

No describiré la tabla al completo, sólo indicar que se han añadido en el `<thead>` una propiedad `colspan="5"` en la etiqueta `<th>` para poder abarcar el total de columnas con la cabecera y el título de la tabla.

En el <tbody> en su primera fila, he usado <tr> para la fila (row) y en ella, he usado los <th> pues eran enunciados de cada columna.

Para el resto de campos de cada celda he usado los <td> que serían los table data y así se hace un uso correcto de la semántica en una tabla.

Se han añadido los iconos con <svg> según muestra en las celdas correspondientes.

Existen 3 celdas que por facilitar la labor a la hora de aplicar un estilo determinado, he optado por añadirles el atributo class con el nombre “subrayado”.

Otra particularidad es que existen dos celdas en las que se ha usado la propiedad rowspan para poder abarcar con esa celda tantas filas como indica su número.

Acabado el <tbody> de la tabla, abrimos y cerramos un <tfoot> con una única fila que abarca con el atributo colspan las 5 columnas. En dicha fila, he incluido el texto indicado en imagen del ejercicio junto con una etiqueta <a> para poner una URL relativa que nos envíe a la página de inscripcion.html.

Tras la <section> de la tabla, he abierto otra <section> con el atributo class, en este caso sí que contienen dos <article> con sus respectivos <h2> y también el <h2> del <section> que se usa para el enunciado de dicha <section> cumpliendo con el estándar de HTML5. Cada <article> contiene mediante listas desordenadas los horarios dentro de cada como se nos pide en el ejercicio de la actividad. Pensé en hacer uso de un <div> por cada bloque de , pero preferí darle un sentido y significado separado y no dos listas sueltas, así, como cada lista hace referencia a una cosa, que se distinga por dos artículos diferentes dentro de la misma sección.

Las horas de dichos horarios se han metido dentro de etiquetas para darle sentido semántico con la importancia de dicho dato.

1.4 Uso de etiquetas en inscripcion.html

```
<form>
  <h2>Formulario de inscripción</h2>
  <fieldset>
    <legend>Datos personales</legend>
    <label for="nombre">Nombre</label>
    <!--Utilizo este pattern para poder introducir-->
    <input type="text" name="nombre" id="nombre" required>
    <label for="apellidos">Apellidos</label>
    <input type="text" name="apellidos" id="apellidos" required>
    <label for="email">E-Mail</label>
    <input type="email" name="email" id="email" required>
    <!--RegExp cogida de w3c-->
    <label for="tel">Teléfono</label>
    <input type="tel" name="tel" id="tel" required>
    <label for="foto">Adjuntar foto <span>(no obligatorio)</span></label>
    <input type="file" name="file" id="foto">
  </fieldset>
  <fieldset>
    <legend>Perfil de usuario</legend>
    <label for="motivos">Explique, brevemente, los motivos de su inscripción</label>
    <textarea name="motivos" id="motivos" minlength="100">
    <label for="profesion">Elija una de las opciones</label>
    <select name="profesion" id="profesion" required>
      <option value="">Seleccione una opción</option>
      <option value="1">Estudiante bachillerato</option>
      <option value="2">Estudiante universitario</option>
      <option value="3">Doctorado</option>
      <option value="4">Profesional Informático</option>
      <option value="5">Profesor</option>
      <option value="6">Otros</option>
    </select><br>
    <label for="talleres">Elija el taller o talleres a los que desea asistir</label>
    <select name="talleres" id="talleres" multiple>
      <option value="1">Computación Cuántica</option>
      <option value="2">Animación 3D</option>
      <option value="3">Análisis de dispositivos</option>
      <option value="4">Metodologías Ágiles</option>
      <option value="5">Realidad Virtual y Aumentada</option>
      <option value="6">First Lego League</option>
    </select>
    <p>*Windows: Mantén la tecla CTRL para elegir más de una opción</p>
    <label for="dias">Elija una de las siguientes fechas de inscripción</label>
    <select name="dias" id="dias" required>
      <option value="">Seleccione una opción</option>
      <option value="1">Asistiré los dos días de inscripción</option>
      <option value="2">Asistiré el 3 de abril</option>
      <option value="3">Asistiré el 4 de abril</option>
    </select>
    <label class="condiciones" for="condiciones">Acepto las condiciones de uso</label>
  </fieldset>
  <button type="submit">Enviar formulario</button>
</form>
```

Todo el cuerpo del <main> está compuesto por un <form> con una cabecera <h2> y dividido en dos <fieldset> con sus respectivas etiquetas <legend> para clarificar cada apartado qué nos va a solicitar.

Todos los campos se han nombrado con el uso de la etiqueta <label> indicando a qué <input> hacían referencia mediante el atributo for en el label y el atributo id en el <input>.

Primer <fieldset>:

Los <input> nombre y apellidos son de tipo text y se han utilizado los atributos que se nos piden con un required, un minlength y maxlength. La longitud, también es verificada en el pattern, pero con los atributos minlength y maxlength hacemos que el usuario pueda recibir un aviso de la longitud mínima y máxima además de impedir también el envío con el campo sin cumplir dicho requisito, mediante el pattern impedimos la entrada de todo lo que no sean caracteres alfabéticos y el espacio en blanco para poder permitir nombres compuestos.

Se ha utilizado un <input> de tipo tel para el teléfono añadiéndole el pattern que nos restringe el dato a lo que evaluamos, obligando a que comience por 6, 7 o 9, que el resto puedan ser de 0 a 9 y longitud máxima de 9 dígitos (el restringido de 6, 7 o 9 + 8 dígitos más = total 9). Se ha activado la propiedad required como se nos indica.

Se ha utilizado un <input> de tipo email para el correo electrónico, haciendo uso a su vez de un pattern para verificar también que se trate de un mail correcto (extraído de la W3C). Dando también la propiedad placeholder para dar un ejemplo de fondo en el input con un ejemplo de formato correcto de mail válido. Se ha activado la propiedad required como se nos indica.

Para la foto, se usó un <input> de tipo file, no siendo requerido dicho campo, se usó etiqueta que abraza un trozo del texto para poderle dar estilos luego.

Segundo <fieldset>:

El primer campo he decidido que fuera un <textarea> ya que había que escribir una cantidad de texto algo más elevada y no sería lógico el uso de <input> de tipo text, he usado las propiedades minlength y maxlength para acotar un mínimo (ya que se nos dice que es required) y un máximo para limitar una entrada extensa de texto.

He usado para los campos profesión, talleres y días las etiquetas <select> con sus <option>, todos ellos required y con la diferencia que en vez de ser un desplegable, el de talleres se trata de una selección múltiple con el atributo multiple activo en la etiqueta <select>.

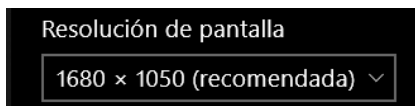
Bajo el <select> con atributo multiple he puesto un <p> con una explicación breve de cómo hacer una selección de varios elementos en Windows y en Mac.

Por último, dentro de este <fieldset> he añadido un <input> de tipo checkbox y required, con la frase de la política y privacidad, la pequeña diferencia es que he usado el <input> abrazado por el <label> para posicionar el checkbox delante de la frase del label.

Finalmente, fuera de los <fieldset> pero dentro del <form> he incluido un <button> de tipo submit para el envío del formulario.

2. Explicación del uso CSS utilizado

NOTA: Por el equipo informático que tengo, no dispongo de monitor fullHD, haciéndome imposible ver el resultado final en una resolución superior a la mía, y que, por lo tanto, las medidas fijas, a pesar de haber optado en la gran mayoría por relativas, pero siendo relativas, si lateralmente se expande 240px, harán que las páginas no se vean como realmente me han quedado, así pues, pido, ruego, se haga uso de una reducción de ventana a una resolución asimilada/aproximada a mi monitor hasta poder ver la web idéntica a las imágenes del PDF del ejercicio.



2.1 Estilos que se aplican a las parte comunes de las tres páginas:

```
* {  
  box-sizing: border-box;  
  margin: 0;  
  padding: 0;  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}
```

Empiezo con un pequeño “reset” que sería el equivalente a poner las propiedades y valores de dentro de esta regla a todo el documento, por ello hacemos uso del selector universal *, indicando así que es una regla que afecta a todo el documento.

En ella ajusto el tamaño de caja al borde de la misma facilitando los movimientos y ajustes al evitar desplazamientos indeseados al sumar border y padding al content.

Quitamos los márgenes por defecto por todos los lados (arriba, derecha, abajo e izquierda) con la propiedad margin y un valor de 0 y hacemos lo mismo con el padding.

Elijo unos tipos de letra lo más asimilados posible al de las imágenes de muestra.

```
header {  
  background-color: #216288;  
  width: 100%;  
  height: 190px;  
  overflow: hidden;  
}
```

A la cabecera (header) mediante un selector de etiqueta le damos el color extraído mediante herramienta de Gimp de las fotos que se nos adjuntaron. Damos el 100% del ancho pudiendo haber usado indistintamente un 100vw, ambas medidas relativas, para que ocupe siempre el ancho disponible. Le doy un alto fijo en este caso de 190px pues me parece lo más aproximado a la muestra. Lo doy fijo pues es un valor que no va a variar. Propiedad overflow hidden para poder abrazar a los elementos que vamos a flotar en su interior.

```
img {
  height: 135px;
  margin: 15px 0 0 15%;
  float: left;
}
```

Selector de etiqueta en esta regla, modificamos las propiedades de la imagen que está en el header a modo de logotipo. Le decimos que ocupe un alto fijo de 135px y no le damos el ancho para que lo calcule automáticamente el navegador, evitando deformar la imagen. Mediante el shorthand le decimos que queremos por arriba 15px con respecto a su contenedor, 0 derecha, 0 abajo y un izquierdo relativo del 15% para conseguir que se pueda adaptar a otros anchos. Flotamos este elemento a la izquierda.

```
.titular {
  width: 45%;
  height: 100%;
  margin-left: 4%;
  text-align: right;
  float: left;
}
```

Selector de clase para el contenedor div que tiene en su interior el h1 y el strong. Le damos a este contenedor un 45% del ancho con respecto a quién lo contiene, en este caso el header. Flotamos a la izquierda este contenedor para ponerlo al lado del logotipo, guardando un margen con él del 4%. Alineamos los textos que contenga el div a la derecha.

```
h1 {
  color: white;
  margin-top: 40px;
  margin-bottom: 20px;
}
```

Selector de etiqueta, damos color al texto del h1, separamos con respecto a su contenedor 40px por su parte superior. Por abajo tenemos la etiqueta strong con la que, en este caso, mantendremos siempre una medida fija de 20px.

```
.titular strong {
  font-size: 1.3em;
}
```

Selector de clase y descendiente, seleccionamos la etiqueta strong que está dentro del contenedor .titular (div) y le variamos el tamaño de letra haciéndolo más grande para que vaya en conjunto con el h1 (por defecto 2em) y el header.

```
nav {
  width: 100%;
  height: 70px;
  background-color: #b8d414;
  text-align: center;
}
```

Selector de etiqueta, le damos un ancho del 100%, un alto fijo de 70px y alineamos los textos que estén contenidos al centro (ul>li>a).

```
nav ul,
nav li,
nav a {
  display: inline-block;
}
```

Selector descendiente de etiquetas y múltiple, aplicamos a los elementos ul, li y a contenidos dentro del nav un display inline-block, ya que ul y li son de bloque y a es de línea, pero necesitamos poderlos poner en línea y aprovechar sus propiedades de bloque para márgenes.

```
nav li {
  list-style: none;
}
```

Selector de etiqueta descendiente, quitamos el estilo de lista al li.

```
nav a {
  margin: 25px 15px 0;
  color: black;
  font-weight: bold;
  font-size: 1.05em;
  text-decoration: none;
}
```

Selector descendiente de etiqueta, afecta a la etiqueta a dentro de nav. Le damos un margen fijo superior de 25px, derecho e izquierdo de 15px, abajo de 0. Texto color negro y en negrita con bold, tamaño relativo y quitamos el subrayado por defecto.

```
nav ul li:first-of-type a {
  text-decoration: underline;
}
```

Selector descendiente de etiqueta y hacemos uso de un selector de pseudo clase :first-of-type para poder coger sólo al primer elemento a, contenido dentro de un nav dentro de un ul y dentro de un li... ignorando el resto de siguientes elementos a, le devolvemos el subrayado eliminado a todos los elementos a en la regla anterior.

```
nav a:hover {  
  color: #216288;  
  transition: 0.4s;  
}
```

Selector de etiqueta y selector de pseudo clase :hover, que al pasar el ratón por encima de dicho elemento nos cambie el color del texto y lo haga en un tiempo de transición de 0.4 segundos para evitar la brusquedad en el cambio de color.

```
footer {  
  width: 100%;  
  height: 200px;  
  background-color: black;  
}
```

Selector de etiqueta a la que le aplicamos la regla indicada en la imagen, dándole un 100% del ancho disponible y una altura fija de 200px, color de fondo con background-color en negro.

```
footer p {  
  width: 80%;  
  margin-left: 20%;  
  padding-top: 20px;  
  color: white;  
}
```

Selector descendiente de etiqueta, y le decimos que el ancho del elemento de bloque p sea de un 80% y lo separamos lateralmente por la izquierda mediante otra medida relativa para dejarlo posicionado en el lugar correspondiente para imitar la muestra. Para separar ambos p, añado un padding interno en la parte superior, haciendo que se separe del borde superior de su contenedor footer y a la vez, el segundo elemento p se separe del que tiene encima. Le damos color de letra blanco a todos los p del footer.

2.2 Estilos empleados en index.html

```
.video {  
  width: 90%;  
  margin-left: auto;  
  margin-right: auto;  
  overflow: hidden;  
}
```

Selector de clase para el contenedor article que contendrá el resto de elementos de esta parte de la web. Le doy un ancho del 90% del espacio disponible y margin-left y right auto para que siempre esté centrado. Añado la propiedad overflow con el valor de hidden ya que en su interior van a existir elementos flotados y queremos que no se nos escapen del flujo de este contenedor.

```
.video h2 {  
  margin: 40px 0 25px;  
  text-align: center;  
}
```

Selector descendente de clase y etiqueta para dar estilos mediante esta regla al encabezado del article. Damos un margen superior que nos separe de arriba 40px, derecho 0 y por bajo con respecto al p que hay bajo 25px. Alineamos el texto de forma centrada.

```
.video p {  
  width: 45%;  
  margin: 15px 50px 0 7%;  
  float: left;  
  text-align: justify;  
  font-size: 1.1em;  
}
```

Selector descendiente de clase y etiqueta para dar estilos al p que hay dentro del article. Le doy un ancho relativo del 45%, margen superior de 15px con respecto al h1, 50px con respecto al elemento que estará flotado a su lado derecho, el video, 0 por bajo y un 7% por el lado izquierdo. Flotamos el elemento a la izquierda, justificamos su texto, ampliamos tamaño de letra.

```
video {  
  height: 200px;  
  clear: left;  
}
```

Selector de etiqueta, al video le doy un alto fijo de 200px para obtener resultado asimilado a muestra. Limpio el float left del p, quedando posicionado este elemento junto al p al existir hueco suficiente y estar roto el flujo normal por el float dado al elemento p.

```
.realiza {
  width: 100%;
  height: 300px;
  margin-top: 60px;
  background-color: #e5fed4;
  overflow: hidden;
}
```

Selector de clase para el contenedor div creado para los dos section (donde y cuándo). Le doy el 100% del espacio disponible, alto fijo de 300px. Evito que los elementos flotados en su interior se salgan del flujo con overflow:hidden y queden dentro del contenedor padre. Separo superiormente con respecto al contenedor anterior 60px y le doy el mismo color que se obtiene de la foto proporcionada con Gimp.

```
.donde {
  width: 30%;
  margin: 20px 70px 10px 20%;
  float: left;
}
```

Selector de clase para el primer section, le doy un ancho del 30% del espacio de su contenedor padre, le doy un margen superior con respecto al mismo de 20px, a la derecha con el elemento que tendrá a su lado flotado (.cuando) de 70px, 10px por la parte de abajo y un margen lateral izquierdo relativo del 20% para adaptarse a cambios de tamaño. Floto este contenedor a la izquierda.

```
.donde a {
  color: grey;
}
```

Selector de clase y descendiente de etiqueta, ponemos color gris a ambos links y luego cambiaremos el link del teléfono tanto su color como su aspecto.

```
.donde :first-of-type a{
  text-decoration: none;
  color: black;
}
```

Selector de clase descendiente con etiqueta y pseudo clase del tipo :first-of-type para trabajar con el primer elemento a que es el teléfono, al que le quitamos el underline y le ponemos color negro al texto.

```
.donde a:hover {
  color: #216288;
  transition: 0.4s;
}
```

Selector de clase, descendiente de etiqueta y pseudo clase para hacer un pequeño efecto de cambio de color al pasar el mouse por encima y con un pequeño tiempo para que no sea un cambio brusco a todos los elementos a.

```
address {
  font-style: normal;
}
```

Selector de etiqueta, le quitamos el estilo por defecto de letra italic y se lo pasamos a normal.

```
.fa-twitter, .fa-facebook-f {
  color: black;
}

.fa-twitter {
  width: 20px;
}

.fa-facebook-f {
  width: 12px;
}

.fa-facebook-f:hover {
  width: 15px;
}

.fa-twitter:hover {
  width: 25px;
}

.fa-twitter:hover,
.fa-facebook-f:hover {
  color: #216288;
  transition: 0.4s;
}
```

Reglas aplicadas a los iconos svg utilizando selectores de clase y pseudo clases para darles el aspecto deseado tanto en tamaño, color y efecto de cambio de tamaño y color al pasar el ratón por encima. Separo los tamaños ya que la F de facebook es mayor, el resto de propiedades que comparten atributos los uno en la misma regla.


```
.cuando {
  width: 30%;
  margin-top: 20px;
  float: left;
}
```

Selector de clase para el contenedor, le doy un ancho del 30% del disponible del padre, dejando así entre los dos elementos flotados un 60% más el 20% de margen lateral aplicado al elemento que está a la izquierda de este, suman un 80%, por lo que, pueden centrarse con los márgenes aplicados previamente al elemento que está a la izquierda de este. Pongo un margen superior idéntico al elemento contiguo para alinearlos y lo floto a izquierda para que se posicione al lado. No hago limpieza posterior del float pues se queda en el interior de este contenedor y el próximo contenedor no coge los float aquí aplicados.

```
.cuando h2,
.donde h2 {
  width: 95%;
  padding-left: 0;
  border-bottom: 3px dotted #b8d414;
}
```

Selector de clase descendiente con etiqueta h2, unifico reglas a ambos h2. Le doy un ancho del 95% ya que no deben alcanzar el subrayado visto en muestra el total del ancho del contenedor padre. Quito el padding left para poner el h2 a la izquierda completamente. Borde inferior de 3px, punteado y color de los patrones de la web.

```
.cuando em {
  font-style: normal;
}
```

Quito a los em contenidos dentro del contenedor con la clase .cuando el estilo que tienen por defecto y los pongo normales.

```
.cuando p:last-of-type {
  text-align: justify;
  font-size: 1.2em;
}
```

Cojo mediante el selector de clase, de etiqueta descendiente y la pseudo clase el último p dentro del .cuando para hacer justificar el texto y darle un tamaño mayor.

```
.cuando p strong a {
  color: grey;
}
```

Selector de clase descendiente con etiquetas, apunto a la etiqueta a, y le cambio el color a gris como en la muestra.

```
.eventos {
  width: 920px;
  height: 430px;
  margin: 40px auto 20px auto;
  overflow: hidden;
}
```

Selector de clase, aquí he optado por hacer un contenedor fijo, sin medidas relativas, por lo tanto, no será adaptable a cambios de tamaño de ventana de visualización. He tomado esta decisión al contener muchos elementos flotados en su interior. Haciendo que aparezca un scroll lateral en el caso de reducción de ventana (inferior a 920px de lado) si alcanza dicho contenedor y dejándolo en idénticas condiciones si la ventana se amplía. Lo que sí será adaptado será el margen derecho e izquierdo a auto para mantener dicho contenedor siempre centrado. Por la parte superior separo 40px con su antecesor y 20px por debajo. Añado un overflow:hidden para que no se escapen del flujo de este contenedor los elementos flotados en su interior.

```
.eventos h2 {
  width: 100%;
  text-align: center;
}
```

Selector de clase y descendiente de etiqueta para seleccionar el h2. Le doy 100% de ancho para que abarque el total del contenedor que lo contiene y para posicionar el título hago un text-align: center, llevando el texto al centro.

```
.eventos img {
  width: 220px;
  height: 120px;
  margin: 5px 20px;
}
```

Selector de clase y descendiente de etiqueta para todas las etiquetas img les he dado valores fijos de tamaño alto y ancho (a pesar de la deformidad de alguna imagen, que es como se muestra en la referencia del ejercicio). Les he dado un margen determinado para separarlas de su contenedor, que tal y como lo estructuré en HTML, será un p que las abraza para dales el título bajo cada imagen. Los márgenes aplicados con respecto al p son 5px por arriba y abajo y de 20px a ambos lados.

```
.eventos p {  
  width: 260px;  
  margin: 10px 10px 10px 30px;  
  font-size: 0.85em;  
  text-align: center;  
  float: left;  
}
```

Selector de clase y descendiente de etiqueta, a todos los p les he dado un ancho fijo suficiente para envolver a las img y que ellas puedan tener los márgenes anteriormente dados. Le doy márgenes para separar cada p entre ellos. Le doy una dimensión de fuente más pequeña para que puedan quedar bajo las imágenes, alineo dicho texto de forma centrada y finalmente floto los p a la izquierda. Dadas las dimensiones y márgenes ya establecidos al contenedor padre, quedarán ubicadas tres arriba y al no existir espacio para más, las tres siguientes quedarán bajo.

2.3 Estilos empleados en programacion.html

```
.tabla-sec {  
    width: 100%;  
    height: 350px;  
}
```

Selector de clase para el section con la clase .tabla-sec. Le doy ancho del 100% y un alto fijo de 350px. Esta sección contiene la tabla.

```
.tabla-sec h2 {  
    margin-top: 55px;  
    margin-bottom: 25px;  
    text-align: center;  
}
```

Selector descendiente de clase y de etiqueta h2 a la que le damos un margen superior con respecto a su contenedor de 55px y un margen inferior de 25px. También centramos el texto.

```
table {  
    margin-left: auto;  
    margin-right: auto;  
}
```

Selector de etiqueta para la tabla, a la que le decimos que siempre esté al centro con ambas propiedades y valores que se muestran. Al darle anchos a sus columnas que no llenarán el 100% del ancho, mantendremos de esta forma centrada la tabla en todo momento.

```
tbody tr th {  
    background-color: #b8d414;  
    text-align: left;  
}
```

Selector descendiente de etiquetas en la que le decimos el color de fondo y alineamos texto de la cabecera de la tabla.

```
tbody tr:nth-of-type(odd) {  
    background-color: #cae9ab;  
}
```

Selector de etiqueta descendiente con un selector de pseudo clase en el que indicamos que los impares tengan dicho color de fondo extraído con Gimp y es diferente al otro verde claro.

```
table tr th:first-child, table tr th:nth-child(4), table tr th:nth-child(5) {
|   width: 100px;
| }

table tr th:nth-child(2) {
|   width: 570px;
| }

table tr th:nth-child(3) {
|   width: 130px;
| }
}
```

Mediante selector de etiquetas descendientes y dos pseudo clases diferentes, seleccionamos cada hijo indicado para darle un ancho fijo a cada columna con las dimensiones que se muestran en la captura, desde el primero hasta el quinto hijo (aquí podría poner otra pseudo clase diferente, pero he continuado numerando).

```
.subrayado {
|   border-bottom: 2px solid black;
| }
}
```

Mediante el selector de clase .subrayado, he puesto un borde inferior negro, sólido de 2px. He usado una clase para facilitar la selección concreta de esas 3 celdas dentro de la tabla.

```
.fa-clock, .fa-map-marker-alt{
|   margin-right: 4px;
| }

.fa-clock {
|   width: 16px;
| }

.fa-map-marker-alt {
|   width: 14px;
| }
}
```

Mediante selectores de clase proporcionados por los svg, le doy los estilos necesarios en tamaño y margen derecho para separar el texto un poco del icono.

```
tfoot {
  text-align: center;
}

tfoot a {
  color: gray;
}

tfoot a:hover {
  color: #216288;
  transition: 0.4s;
}
```

Con selector de etiqueta, descendiente y una pseudo clase, le doy alineación al texto del table foot, le doy un color gris al link que hay dentro y le hago un pequeño estilo de color y transición al paso del mouse por encima del link.

```
.programacion {
  width: 100%;
  height: 350px;
  overflow: hidden;
  background-color: #e5fed4;
}
```

Selector de clase en la que le decimos a este section que tenga un ancho del 100% de lo disponible, un alto fijo de 350px, añadimos la propiedad overflow con el valor de hidden para que los elementos que vamos a flotar en su interior, permanezcan dentro y le damos un color de fondo.

```
.programacion h2 {
  padding-left: 6%;
  font-size: 2em;
}
```

Selector de clase descendiente con selector de etiqueta y le damos al h2 del section un desplazamiento de su contenido por la izquierda relativo al ancho y aumentamos su tamaño para darle la apariencia de la muestra, estando ligeramente desplazado por la izquierda no siendo un tipo de alineación establecido el que vemos en la muestra.

```
.dia1,
.dia2 {
  width: 350px;
  margin-top: 30px;
  float: left;
}
```

Selector de clase múltiple, con lo que le damos los mismos valores a ambos contenedores article en este caso. Ancho fijo de 350px y un margen superior con respecto al h2 de 30px. Flotamos ambos a izquierda.

```
.dia1 {
  margin-left: 14%;
  margin-right: 30px;
}
```

Seleccionamos con la clase y usamos este contenedor para que nos coloque ambos flotados. Le damos un margen relativo por la izquierda con su contenedor padre y un margen fijo de 30px con respecto al contenedor que tiene a su derecha, .dia2.

```
.dia1 h2,
.dia2 h2 {
  display: inline-block;
  padding: 0;
  margin-left: 9px;
  margin-bottom: 15px;
  background-color: #bdfc80;
}
```

Juntamos estilos en esta regla que selecciona mediante selector descendiente de clase y etiqueta, los h2 de ambos contenedores. Les hacemos que pasen a ser inline-block mediante la propiedad display y el citado valor para poder luego colorear su fondo y afecte sólo al contenido. Retiramos todos los padding para ajustar el tamaño de caja a contenido. Movemos la caja 9px por la izquierda para que quede alineado con los siguientes elementos de la lista que están bajo. Dejamos un margen por la parte de bajo con respecto a la lista y finalmente le damos ese color de fondo que vemos en la muestra.

```
.dia1 li,
.dia2 li {
  list-style: none;
  padding: 10px;
}
```

Mismo uso de selector que antes y quitamos los puntos de estilo de lista por defecto de los elementos li y le añadimos un espaciado interior de 10px para obtener ese resultado de separación entre todos los elementos li que vemos en la muestra.

2.4 Estilos empleados en inscripcion.html

```
form h2 {  
    text-align: center;  
    margin: 40px 0 25px;  
}
```

Selector descendiente de etiquetas, con ello centramos el enunciado de este formulario y le damos márgenes fijos con respecto al nav de 40px, 0 por ambos lados y de 25px por la parte inferior.

```
fieldset {  
    width: 60%;  
    padding: 20px;  
    margin: 0 auto 25px auto;  
    background-color: #cae9ab;  
    box-shadow: 3px 3px 6px #216288;  
}
```

Selector de etiqueta y damos ambos fieldset el mismo estilo con esta regla. Le decimos que queremos un ancho del 60% del espacio disponible, hacemos un padding por todos los lados de 20px para separar el contenido de los bordes. Le decimos que no necesitamos un margen superior, usamos auto para derecha e izquierda haciendo que siempre esté el elemento centrado y por abajo con respecto al siguiente fieldset guardaremos 25px de distancia así como con respecto al botón que habrá al final.

```
legend {  
    border: 3px solid #216288;  
    border-radius: 4px;  
    padding: 5px;  
    background-color: #b8d414;  
    font-weight: bold;  
}
```

Selector de etiqueta para darle un estilo concreto a este campo en el que le damos un borde sólido de 3px con el color del patrón de colores de la web, redondeamos los cantos de dicho borde con un radio de 4px, damos un espaciado interior de 5px, le ponemos un color del patrón de la web de fondo y hacemos que la fuente esté en negrita.

```
label {  
    display: inline-block;  
    width: 200px;  
    margin-left: 10%;  
}
```


Usamos el selector de etiqueta y hacemos que todos nuestros label pasen a ser de tipo inline-block dejando de ser elementos de línea y poder trabajar sus medidas, dándoles un ancho fijo de 200px a todas y un margen izquierdo relativo del 10%.

```
label span {  
    font-size: 0.8em;  
}
```

Selector descendiente de etiquetas, para poder hacer más pequeño el texto entre paréntesis que es aclaratorio de que subir un archivo de foto es “(no obligatorio)” para el usuario.

```
input,  
select,  
textarea {  
    width: 250px;  
    margin-bottom: 10px;  
    vertical-align: top;  
}
```

Selector de etiquetas múltiple y hacemos que el ancho sea fijo para todos ellos de 250px, separamos todos por bajo 10px y a pesar de que la propiedad vertical-align no aplica a los tres, aplica a dos de ellos (select, en el caso de múltiple y a textarea) lo he dejado en la misma regla para no hacer otras reglas independientes sólo para esa propiedad que no afecta a los input ni a los select que no sean multiple.

```
input:required:invalid {  
    background-color: ■ rgb(230, 106, 106);  
}  
  
input:required:valid {  
    background-color: ■ rgb(132, 216, 132);  
}
```

Mediante selector de etiqueta y las pseudo clases que vemos, hacemos el requerimiento de color de fondo en función de la validez o no de un input requerido por el enunciado de la actividad. He usado un tono de rojo y verde un poco menos intensos, probé añadiendo un alpha pero no me convencía el resultado obtenido.

```
input:focus,  
textarea:focus {  
    font-weight: bold;  
    color: ■ #216288;  
}
```

Selector múltiple de etiqueta y pseudo clase para cuando tenemos el foco dentro de un input o en el textarea, a la hora de escribir, nuestro texto se escriba en uno de los colores del patrón de la web y en negrita.

```
fieldset p {  
  width: 495px;  
  margin: 0 auto 10px auto;  
  font-size: 0.65em;  
}
```

Selección descendiente mediante etiquetas del único elemento p del formulario, es descendiente para no cambiar los estilos de los p del footer. Le damos un ancho fijo determinado para que quede ajustado bajo el input correspondiente (del que queremos describir la funcionalidad múltiple). Añadimos un margen superior de 0 pues el input ya tiene un margen inferior, derecho e izquierdo auto por si se redimensiona el fieldset siga centrado y por bajo sí le damos un margen de 10px. Reducimos el tamaño de letra pues es un pequeño párrafo explicativo de cómo hacer una selección múltiple en Windows y Mac.

```
.condiciones {  
  width: 100%;  
  margin-top: 20px;  
}
```

Damos estilos concretos a este último label mediante el selector de clase que le hemos puesto, para poder cambiar el ancho fijo previo a que ocupe el 100% del espacio disponible, se nos separe un poco más del resto con un margin top de 20px.

```
.condiciones input {  
  width: 15px;  
  margin-top: 5px;  
}
```

Modificamos el ancho de este input y lo hacemos muy pequeño ya que sólo es un checkbox que queremos nos quede próximo al label, le damos un pequeño margen superior para alinearlos con respecto al texto del label de su derecha.

```
.submit {
  display: block;
  padding: 10px 25px;
  background-color: #b8d414;
  margin: 20px auto 40px auto;
  border: 2px solid #216288;
  border-radius: 4px;
  box-shadow: 1px 1px 3px #216288;
  font-size: 1.1em;
  font-weight: bold;
}
```

Selector de clase, hacemos una regla de estilos un tanto extensa pues el input type submit de envío del formulario he querido destacarlo. Así pues hacemos que sea un elemento de bloque. Añadimos espaciado interior generoso de 10px por arriba y por abajo y de 25px a los lados. Damos color de fondo. Hacemos que los márgenes laterales sean auto para que siempre esté al centro de la página, también le damos 20px por arriba para separarlo más todavía del form y 40px por bajo para separarlo del footer. Añadimos un borde sólido de 2px con el color de patrón de la web. Redondeamos con el mismo radio que los legend a 4px los bordes. Ponemos una sombra pequeña de 1px que salga por la derecha y 1px que salga por bajo y un sombreado de 3px con el mismo color del borde. Aumentamos el tamaño del texto y lo ponemos en negrita.

```
.submit:hover {
  cursor: pointer;
  box-shadow: 3px 3px 3px #b8d414;
  transition: 0.4s;
}
```

Damos un retoque final al botón del submit con este selector de etiqueta y pseudo clase que nos hace que al pasar el cursor del mouse por encima tengamos un puntero con forma de dedo, cambiamos la sombra por una un poco mayor y con el color del background dando un efecto de iluminación y lo hacemos con una pequeña transición para evitar la brusquedad del cambio de color.

3. Capturas íntegras mediante impresión de página en pdf del resultado en mi monitor:



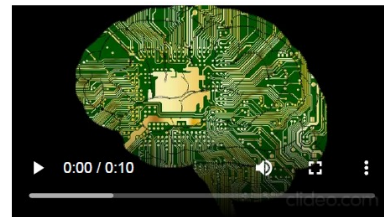
II Jornadas de Informática

Valencia, 3 y 4 de Abril de 2021

[Inicio](#) [Programación](#) [Registro a las jornadas](#)

II Jornadas de Informática

Las Jornadas de la Informática son un espacio de divulgación de temas de actualidad en el que participan empresas, profesionales y estudiantes, ofreciendo una visión de Tendencias tecnológicas actuales, Nuevos ámbitos de aplicación de las Tecnologías de la Información y las Comunicaciones, y Oportunidades de empleo y modelos de empresa actuales.



¿Dónde se realiza?

Auditorio

Plaza de la Misericordia, Valencia, 46115 Tel.

93 555 55 55

info@jornadas.org



¿Cuándo se realiza?

Sábado, 3 de abril, de 9:00 a 20h

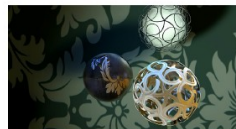
Domingo, 4 de abril, de 9:00h a 18:30h

Entrada libre con inscripción previa

Eventos



Computación Cuántica: la Próxima Revolución de la informática



Animación 3D en la industria del Videojuego



Trailer sobre análisis de dispositivos móviles



Metodologías Ágiles y Transformación Digital



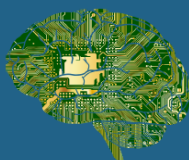
Realidad virtual y aumentada: experiencias asequibles y útiles en la educación



First Lego League

Actividad realizada por José Vicente Falcó Milla, para la asignatura de Lenguajes de Marcas

Primer trimestre 2021/2022



II Jornadas de Informática

Valencia, 3 y 4 de Abril de 2021

[Inicio](#) [Programación](#) [Registro a las jornadas](#)

Programa de las II Jornadas Informática

Todos los talleres de las jornadas				
Fecha	Taller	Ponente	Horario	Ubicación
2 de Abril	Computación Cuántica: la Próxima Revolución de la informática	Javier García	9 a 12h	Aulario I
	Animación 3D en la industria del Videojuego	Vanessa Pérez	12 a 14h	Aulario I
	Taller sobre análisis de dispositivos móviles	Dionisio Vidal	14 a 18h	Aulario II
	Metodologías Ágiles y Transformación Digital	Pablo Fernández	18 a 20h	Aulario II
4 de abril	Realidad virtual y aumentada: experiencias asequibles y útiles en educación	Óscar Sapena	10 a 12h	Aulario I
	La predicción del tiempo	Nuria Medina	12 a 14h	Aulario I
	First Lego League	Pedro Sosa	16 a 18h	Aulario II

La entrada a las actividades es libre con [inscripción](#) previa

Programación

3 de abril de 2021

9h, recepción de los participantes

10 a 14 h, talleres

14 a 16 h, pausa comida

16 a 20 h, talleres

4 de abril de 2021

9h a 14h, talleres

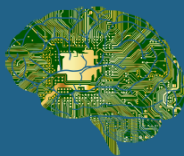
14 a 16 h, pausa comida

16 a 18 h, talleres

18 a 18:30 h, clausura

Actividad realizada por José Vicente Falcó Milla, para la asignatura de Lenguajes de Marcas

Primer trimestre 2021/2022



II Jornadas de Informática

Valencia, 3 y 4 de Abril de 2021

[Inicio](#) [Programación](#) [Registro a las jornadas](#)

Formulario de inscripción

Datos personales

Nombre
Apellidos
E-Mail
Teléfono
Adjuntar foto (no obligatorio) Ningún a...cionado

Perfil de usuario

Explique, brevemente, los motivos para participar en las jornadas
Elija una de las opciones
Elija el taller o talleres a los que desea asistir*

*Windows: Mantén la tecla CTRL para elegir varias. Mac: Mantén la tecla COMMAND para elegir varias.
Elija una de las siguientes opciones
☐ He leído y acepto la política de privacidad y el aviso legal.

[Enviar formulario](#)

Actividad realizada por José Vicente Falcó Milla, para la asignatura de Lenguajes de Marcas

Primer trimestre 2021/2022

4. Validaciones W3C:

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file **index.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ningún archi...seleccionado

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 13 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 21.12.14

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for programacion.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ningún archi...seleccionado

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 10 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 21.12.14

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file inscripcion.html

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ningún archi...seleccionado

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 53 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 21.12.14

Resultados del Validador CSS del W3C para estilos.css (CSS versión 3 + SVG)

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 3 + SVG](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:



```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referen">
  
</a>
</p>
```



```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referen">
  
</a>
</p>
```