

## Tarea nº 2 – Utilización de lenguajes de marcas en entornos web

Lenguajes de Marcas y Sistemas de Gestión de Información

16 DE NOVIEMBRE DE 2022

CIFP CARLOS III - CARTAGENA  
SANTIAGO FRANCISCO SAN PABLO RAPOSO  
2º CURSO DAW



## Contenido

1.- Enunciado.....	2
2.- Items. ....	2
2.1.- Ítem 1 (0,25 puntos).....	2
2.2.- Ítem 2 (0,25 puntos).....	2
2.3.- Ítem 3 (0,25 puntos).....	3
2.4.- Ítem 4 (0,25 puntos).....	3
2.5.- Ítem 5 (0,25 puntos).....	4
2.6.- Ítem 6 (0,25 puntos).....	5
2.7.- Ítem 7 (0,50 puntos).....	5
2.8.- Ítem 8 (0,5 puntos).....	6
2.9.- Ítem 9 (0,25 puntos).....	6
2.10.- Ítem 10 (0,25 puntos).....	7
2.11.- Ítem 11 (1 punto). ....	7
2.12.- Ítem 12 (1 punto). ....	8
2.13.- Ítem 13 (1 punto). ....	8
2.14.- Ítem 14 (1 punto). ....	9
2.15.- Ítem 15 (1,5 puntos).....	11
2.16.- Ítem 16 (1,5 puntos).....	14
Bibliografía. ....	15

# Tarea para LMSGI02.

## 1.- Enunciado.

Diseñar una página web que responda al aspecto y a las especificaciones que se dan a continuación. Cada especificación cumplida sumará la puntuación que aparezca junto a ella. Además, el alumno deberá entregar un documento de texto (.doc, .docx, .odt o .pdf) en el que justificará mediante qué etiqueta, atributos o técnica ha resuelto cada ítem.

## 2.- Items.

### 2.1.- Item 1 (0,25 puntos).

La página debe cumplir con el estándar HTML 5, indicando que está en español y codificada como UTF-8.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
```

Como se puede ver, se ha resuelto utilizando el atributo Lang=es en la etiqueta html, y poniendo como metadato que el conjunto de caracteres (charset) disponibles en el documento HTML es el UTF-8.

### 2.2.- Item 2 (0,25 puntos).

El título de la página será "LMSGI02 - DENTONLINE" y, como metadato, se incluirán palabras clave tales como "dentista", "online", "salud", etc.

```
<meta name="keywords" content="dentista,online,salud">
<title>LMSGI02 - DENTONLINE</title>
```

- La etiqueta **meta** (para especificar metadatos del documento HTML), permiten incluir las “**keywords**” (palabras clave) que pide el enunciado.
- La etiqueta **title** permite poner un título al documento HTML.

### 2.3.- Item 3 (0,25 puntos).

La hoja de estilos se desarrollará en un fichero diferenciado y se enlazará en el código HTML.

El enlace a la hoja de estilos está hecho dentro del head, mediante la siguiente etiqueta:

```
<link rel="stylesheet" type="text/css" href="assets/css/reglas.css">
```

El fichero .css lo podemos encontrar dentro de la carpeta assets/css/reglas.css

### 2.4.- Item 4 (0,25 puntos).

La página se adaptará al ancho de la pantalla en la que se visualice.

Esto se ha conseguido **haciendo que todo el contenido de la página esté dentro de un contenedor que aglutine todos los elementos.**

```
<body>
  <div id="pagina">
    <header class="flex-row flex-ai-end flex-jc-spaceb flex-wrap">
      [...]
    </header>
    <section id="formulario" class="flex-grow">
      [...]
    </section>
    <footer class="color-grey">
      [...]
    </footer>
  </div>
</body>
```

Algunos elementos tienen propiedades específicamente que las hacen comportarse de la manera en que lo hacen para poder crecer. **Por ejemplo:**

- **Para que el formulario crezca**, vemos que el **section tiene asignada la clase flex-grow**, que hace el contenedor del formulario crezca todo lo que pueda.
  - Por su parte, **los inputs de cada campo también tienen aplicada esta clase** para que crezcan todo lo que pueda (según el tamaño del navegador). Podemos ver una muestra de ello visualizando el código de uno de los campos:

```
<div class="campo-formulario flex-row flex-ai-center">
  <label class="lbl-width">Nombre</label>
  <input type="text" id="nombre" class="form-caja-texto flex-grow">
</div>
```

- **La imagen de la silla del dentista** tiene aplicada una regla CSS para establecer un ancho del 100% dentro del contenedor al que pertenece. Además, tiene definidos un tamaño mínimo y un tamaño máximo (el de la propia fotografía).
  - **Esto es para mantener** (más o menos) **la proporción de dimensiones entre la imagen y el formulario**. De esta forma, como el contenedor div del formulario tiene la clase flex-grow, ocupará todo el espacio posible que le permita la imagen, siendo la imagen la primera en crecer (hasta llegar a su tamaño máximo), y el formulario el que crece en segundo lugar después de que la imagen alcanza su tamaño máximo.

```
#cuerpo img {
  min-width: 300px;
  width: 100%;
  max-width: 640px;
  max-height: 427px;
}
```

- **Para mantener el menú de navegación en su sitio** (en el centro del header), hemos **usado una propiedad flex** que permite dejar la misma distancia de separación entre objetos de un contenedor: **justify-content: space-between**, que está **plasmada en la clase flex-jc-spaceb**.

```
.flex-jc-spaceb {
  justify-content: space-between;
}
```

## 2.5.- Item 5 (0,25 puntos).

Entre la página y los laterales de la pantalla quedará un margen de 64px, de color negro.

La codificación de este ítem se encuentra en el fichero CSS, y es la siguiente:

```
body {
  background-color: black;
  margin: 0px;
}

#pagina {
  background-color: white;
  margin: 0px 64px 0px 64px;
  padding: 8px;
  font-family: Calibri;
  font-size: 1.1em;
}
```

Primero **le establezco a la etiqueta body un margen de 0 píxeles por todos los lados** (arriba, abajo, izquierda y derecha). Y luego a la etiqueta div con identificador “pagina” (que es en realidad el elemento raíz, que contiene al resto de elementos de la página web), es a la que le he aplicado el margen pedido por el enunciado.

## 2.6.- Item 6 (0,25 puntos).

Se mostrará un encabezado haciendo uso de las etiquetas semánticas que HTML5 propone para ello.

Efectivamente, se ha usado la etiqueta header para crear el encabezado de la página web:

```
<header class="flex-row flex-ai-end flex-jc-spaceb flex-wrap">
  <div id="logotipo">
    <span style="font-weight: bolder;">DENT</span>ONLINE
  </div>
  <nav>
    <a href="index.html" class="links flex-grow">inicio</a>
    <a href="servicios.html" class="links flex-grow">servicios</a>
    <a href="clinicas.html" class="links flex-grow">clínicas</a>
    <a href="noticias.html" class="links flex-grow">noticias</a>
  </nav>
  <input type="button" id="btnAreaClientes" class="btn-azul" value="Área de clientes">
</header>
```

## 2.7.- Item 7 (0,50 puntos).

El encabezado contendrá los elementos que se ven en la captura. DENTONLINE a la izquierda, el botón a la derecha y, centrado, un menú con 4 vínculos a páginas del mismo dominio, cuyas URLs relativas puedes inventarte.

En el código del [ítem anterior \(2.6\)](#), se puede ver como están los elementos DENTONLINE, el menú de navegación con 4 vínculos (mediante la etiqueta semántica HTML5 nav) cuyas URL relativa me he inventado (es decir, no funcionarán, ya que esto es solo una maqueta web), y el botón de Área de clientes.

Respecto a **cómo he logrado la distribución de los elementos deseada**, mencionar que he hecho uso de **varias clases flex definidas en el CSS**, pero dos de ellas son protagonistas en el cumplimiento de este ítem:

- **flex-row:** permite especificar que el contenedor al que se aplica, va a tener sus elementos dispuestos según el sistema de posicionamiento Flex, y distribuidos horizontalmente (en fila, de ahí que se llame row).
- **flex-jc-spaceb:** esta clase aplica a los elementos del contenedor un estilo justify-content: spacebetween, que lo que hace es que ajusta los elementos para que haya siempre el mismo espacio entre los elementos que se encuentren en dicho contenedor.

Las otras dos clases flex utilizadas son para otros ítems que tienen que ver con la apariencia conseguida respecto a la ilustración del enunciado de cómo debe quedar.

### 2.8.- Ítem 8 (0,5 puntos).

La imagen deberá referenciar a un fichero ubicado en unas carpetas “assets/img”. Deberá poseer un texto alternativo que describa lo que se ve en ella. Sólo ocupará el área estrictamente necesaria para ser visualizada.

```
<div class="flex-ai-start">
  
</div>
```

La imagen se inserta con una etiqueta img, dentro de un contenedor div. Podemos ver que la imagen **está en /assets/img/silladentista.jpg (mediante el atributo src)**, y que posee un **texto alternativo** (establecido mediante el atributo alt) de la etiqueta img.

**Respecto al área que ocupa**, ha sido necesario hacer uso de CSS en la imagen, para que ocupe el 100% de su contenedor:

```
#cuerpo img {
  min-width: 300px;
  width: 100%;
  max-width: 640px;
  max-height: 427px;
}
```

- **El estilo min-width** especifica un tamaño mínimo de la imagen, a partir del cual, no podrá decrecer más.
- **Los estilos max-width y max-height** han sido establecidos a las dimensiones originales de la imagen, para que no ocupe más espacio y se “píxele”, y también por cuestiones de estética.

El contenedor no tiene aplicada la clase flex-grow, mientras que el contenedor del formulario, sí lo lleva, por lo que este empujará al máximo al contenedor de la imagen. **De esta forma, tal y como pide el enunciado, la imagen solo ocupará el área estrictamente necesaria para ser visualizada.** Como mínimo, tendrá un ancho de 300px, y como máximo, el tamaño máximo de la imagen, con el fin de intentar conservar la proporción de espacio que ocupan en la pantalla.

### 2.9.- Ítem 9 (0,25 puntos).

Por su parte, la parte derecha podrá expandirse (ya que la página se adapta al ancho disponible, según ya se ha dicho) para hacer los campos del formulario más anchos.

Esto se consigue **aplicando la clase flex-grow a la etiqueta contenedora del formulario** (en este caso, se ha optado por una etiqueta section de HTML5):



```
<section id="formulario" class="flex-grow">
  [...]
</section>
```

Como ya hemos comentado en el ítem 4, **los inputs de cada campo también tienen aplicada esta clase** para que crezcan todo lo que pueda (según el tamaño del navegador). **Podemos ver una muestra de ello visualizando el código de uno de los campos:**

```
<div class="campo-formulario flex-row flex-ai-center">
  <label class="lbl-width">Nombre</label>
  <input type="text" id="nombre" class="form-caja-texto flex-grow">
</div>
```

#### 2.10.- Ítem 10 (0,25 puntos).

La parte inferior (en gris) debe contenerse en una capa de tipo “**pie**”.

El pie de página está contenido en un contenedor tipo “footer”, propio de la especificación semántica de HTML5:

```
<footer class="color-grey">
  <span style="font-weight: bold;">#169 2022 CIFP Carlos III</span>
  &ndash;
  <a href="avisoLegal.html" class="links">Aviso legal</a>
</footer>
```

Y tiene aplicado un **estilo propio llamado “color-grey”** para establecer el color gris al texto contenido dentro de él.

#### 2.11.- Ítem 11 (1 punto).

El formulario deberá estar definido para enviar, mediante POST, todos los datos que pueden verse. Deberás proveer todo lo necesario para que así sea. La página destinataria de la información será “requestcontactform.php”.

**Para poder enviar los datos del formulario, simplemente he establecido mediante el atributo action el fichero .PHP** al cual habría que mandarle los datos, y **mediante el atributo method, he establecido que fuera el método POST**. Los datos que se enviarían serían cada uno de los campos (inputs y textarea) identificados por su propio identificador.

Al ser solo una maqueta web, este código no enviará ningún dato a ninguna página PHP ya que no existe, es inventada, igual que los enlaces a las páginas del menú de navegación.

```
<form action="requestcontactform.php" method="post">
  <div class="campo-formulario flex-row flex-ai-center">
    <label class="lbl-width">Nombre</label>
    <input type="text" id="nombre" class="form-caja-texto flex-grow">
  </div>
```

```
<div class="campo-formulario flex-row flex-ai-center">
  <label class="lbl-width">Email</label>
  <input type="email" id="email" class="form-caja-texto flex-grow">
</div>
<div class="campo-formulario flex-row flex-ai-center">
  <label class="lbl-width">Teléfono</label>
  <input type="tel" id="telefono" class="form-caja-texto flex-grow">
</div>
<div class="campo-formulario flex-row flex-ai-center">
  <label class="lbl-width">Mensaje</label>
  <textarea id="mensaje" rows="4" class="form-caja-texto flex-grow"></textarea>
</div>
<div class="campo-formulario form-checkbox flex-row flex-ai-center flex-wrap">
  <input type="checkbox" id="checkbox-infoComercial">
  <label id="lbl-checkbox-infoComercial" for="checkbox-infoComercial">Acepto
recibir informaci&oacute;n comercial</label>
</div>
<div class="submit-formulario flex-row flex-jc-end">
  <input type="submit" class="btn-naranja" value="Enviar">
</div>
</form>
```

## 2.12.- Ítem 12 (1 punto).

Los **campos** de los formularios deben estar **tipificados** en concordancia con su uso (texto, teléfono y email). El campo para el mensaje debe ser multilínea.

En el código adjunto en el ítem anterior 2.11, podemos observar que:

- **El campo email** tiene un input de tipo “email”, según la especificación semántica de HTML5.
- **El campo teléfono** tiene un input de tipo “tel”, según la especificación semántica de HTML5.
- **El resto de campos** tienen establecido el type=”text”, según la especificación semántica de HTML5, y el textarea (caja de texto) del mensaje, como tal, no necesita especificar su tipo como si lo requiere la etiqueta input por ser tan polivalente.

## 2.13.- Ítem 13 (1 punto).

Se hace uso de **identificadores** para los ítems únicos a los que haya que aplicar algún estilo o posicionamiento especial.

Esto se puede ver a lo largo de todo el código.

**Con el atributo id** (aplicable a cualquier etiqueta HTML), **es posible identificar de manera inequívoca un elemento**. Por su parte, CSS nos permite referirnos a estos elementos mediante el selector *#identificador*.

### Ejemplos de este uso:

- La **etiqueta div con identificador “pagina”**, que contiene todos los elementos de la página. Esta etiqueta sería el “elemento raíz” del cual parten el resto de elementos. Por tanto, tiene un estilo específico, y para ello se utiliza un identificador como selector en las reglas CSS.
- El **“logotipo”** (para aplicar el tamaño de fuente deseado a la etiqueta div contenedora del logotipo **DENTONLINE**), etc.
- La **etiqueta div con identificador “cuerpo”**, que tiene aplicado una regla de padding (márgenes del contenedor interiores).
- En general, esta técnica ha sido utilizada **para aplicar margins, paddings o tamaños personalizados** a elementos del documento HTML concretos.
  - **Este es el caso de los identificadores:** “formulario”, “titulo-formulario”, “checkbox-infoComercial” y “lbl-checkbox-infoComercial”,
- También se ha usado para modificar el tamaño de fuente en el caso del label asociado al input **checkbox** del formulario, estableciéndole un tamaño de 0.88em. Esta regla está asociada al identificador “lbl-checkbox-infoComercial”.

**Un caso particular:** para la imagen se ha usado el identificador de la etiqueta div contenedora del cuerpo de la página web, para referirnos a la imagen. Realmente, como la única imagen que aparece en la página es esa, podríamos habernos referido a la etiqueta img directamente, o crear una clase específica para la imagen. Sin embargo, he optado por esta opción para mostrar una forma más de utilizar los selectores CSS.

**También cabe destacar que, por ejemplo, para poner las negritas** en el logotipo de **DENTONLINE**, simplemente he usado el atributo style, aplicando directamente la regla Font-weight: bolder; o en el pie de página, para poner en negrita el Copyright.

### 2.14.- Ítem 14 (1 punto).

Aquellos **ítems que se repitan** deberán ser **agrupados por clases** y sus estilos aplicados a nivel de clase. El “aviso legal” es un vínculo del mismo tipo que los de la parte superior.

Esto ha sido aplicado a lo largo de todo el documento HTML cuando ha hecho falta aplicar estilos comunes a varios elementos.

- **Todos los elementos que usan el sistema de posicionamiento Flex**, utilizan clases Flex creadas para tal fin (aunque no todas las clases Flex que he creado han terminado siendo utilizadas).
  - **Flex-row:** establece el sistema de posicionamiento Flex y especifica una distribución de elementos en línea o fila (row).
  - **Flex-grow:** permite hacer crecer al elemento al que se le aplique todo el ancho que le permita el contenedor al que pertenezca.
  - **Flex-grow-66:** permite hacer crecer al elemento al que se le aplique un 66% del ancho que le permita el contenedor al que pertenezca.
  - **Flex-grow-33:** permite hacer crecer al elemento al que se le aplique un 33% del ancho que le permita el contenedor al que pertenezca.

- **Flex-grow-80:** permite hacer crecer al elemento al que se le aplique un 80% del ancho que le permita el contenedor al que pertenezca.
- **Flex-grow-20:** permite hacer crecer al elemento al que se le aplique un 20% del ancho que le permita el contenedor al que pertenezca.
- **Flex-ai-center:** permite que los elementos de un contenedor estén alineados verticalmente en el centro con respecto a otros elementos del mismo contenedor que ocupen más de altura. (Propiedad align-items).
- **Flex-ai-baseline:** permite que los elementos de un contenedor estén alineados verticalmente en la línea base de los elementos tipo texto. (Propiedad align-items).
- **Flex-ai-start:** permite que los elementos de un contenedor estén alineados verticalmente en el centro con respecto a otros elementos del mismo contenedor que ocupen más de altura. (Propiedad align-items).
- **Flex-ai-end:** permite que los elementos de un contenedor estén alineados verticalmente en el centro con respecto a otros elementos del mismo contenedor que ocupen más de altura. (Propiedad align-items).
- **Flex-jc-spaceb:** permite que los elementos en línea de un contenedor estén separados entre sí (horizontalmente) a la misma distancia.
- **Flex-jc-spaceb:** permite que los elementos en línea de un contenedor estén alineados a la izquierda. (Propiedad justify-content)
- **Flex-jc-end:** permite que los elementos en línea de un contenedor estén separados entre sí (horizontalmente) a la misma distancia. (Propiedad justify-content).
- **Flex-wrap:** mediante esta propiedad, los elementos "hijos" son obligados a fluir en varias líneas, esto es, permite que los elementos puedan reposicionarse en función del espacio disponible en la ventana del navegador. Por ejemplo, cuando se reduce o se amplía el tamaño de la misma.
- **Clases para dar formatos al texto:**
  - **Links:** para los enlaces del menú de navegación y del aviso legal del pie de página. Además, tiene una especificación de la pseudoclase hover (que permite controlar el comportamiento del link al poner el cursor sobre él).
  - **Campo-formulario:** establece una serie de propiedades (tamaño de fuente, padding-top y padding-bottom) al div que contiene al conjunto de label + input.
  - **Lbl-width:** permite que todos los label del formulario tengan un tamaño fijo de 5.2em para que los inputs crezcan con flex-grow perfectamente alineados entre sí (como si estuvieran dentro de una tabla HTML).
  - **Form-caja-texto:** da formato a los inputs y textarea del formulario.
  - **Campo-checkbox:** da un padding-top y un padding-bottom al checkbox que permite "Aceptar recibir información comercial".
  - **Btn-azul (y pseudoclases hover y active derivadas):** configura todos los detalles para hacer botones azules como el del encabezado de "Área de clientes".

- **Las pseudoclases hover y active** actúan cuando el cursor se posa encima del botón, y cuando el cursor hace clic sobre el botón, respectivamente
- **Btn-naranja (y pseudoclases hover y active derivadas):**

## 2.15.- Ítem 15 (1,5 puntos).

La estructura de capas diseñada por el alumno es coherente y permite llevar a cabo la composición visual requerida.

Esto se puede observar analizando el código. La estructura de capas ha sido realizada realizando una descomposición por capas de la ilustración propuesta en el enunciado.

1. **Descomposición de primer nivel:** nos damos cuenta de que en primera instancia hay un primer contenedor que contiene al resto de contenedores. Es el contenedor raíz.



2. **Descomposición de segundo nivel:** una vez dentro del contenedor raíz, podemos observar tres capas bien diferenciadas: header (encabezado), cuerpo y footer (pie de página).

The screenshot shows the top section of the DENTONLINE website. On the left is the logo 'DENTONLINE'. In the center is a navigation menu with links: 'inicio', 'servicios', 'clínicas', and 'noticias'. On the right is a button labeled 'Área de clientes'. Below the navigation menu is a large image of a dental chair. To the right of the image is a 'Formulario de contacto' (Contact Form) with fields for 'Nombre', 'Email', 'Teléfono', and 'Mensaje'. There is also a checkbox for 'Acepto recibir información comercial' and an 'Enviar' button. At the bottom of the page, there is a footer with the text '© 2022 CIFP Carlos III – [Aviso legal](#)'.

3. **Descomposición de tercer nivel:** entramos dentro de cada uno de los elementos identificados en la descomposición anterior, y localizamos los “elementos atómicos”:
  - a. **Encabezado (header):** lo descomponemos horizontalmente en tres piezas fundamentales: “logotipo”, menú de navegación y botón de “Área de clientes”.

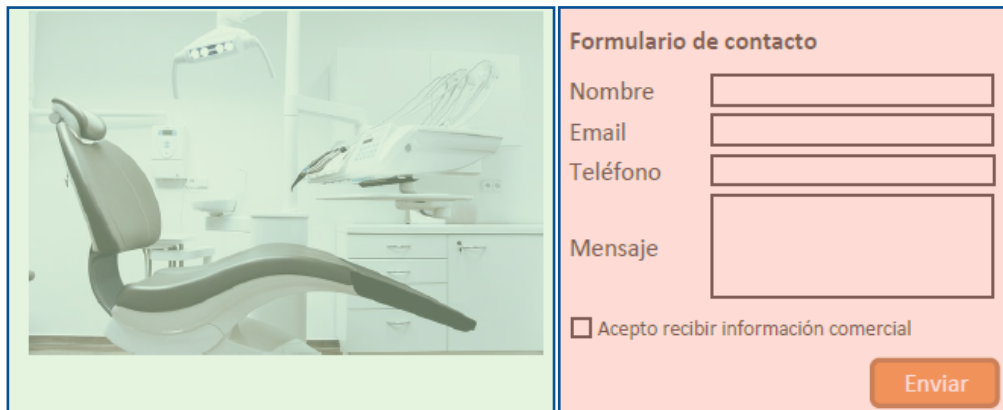
The diagram illustrates the horizontal decomposition of the header. It is divided into three distinct sections: the 'DENTONLINE' logo on the left, the navigation menu ('inicio', 'servicios', 'clínicas', 'noticias') in the center, and the 'Área de clientes' button on the right.

**Dentro del menú de navegación, tendríamos “elementos atómicos”,** que serían los distintos vínculos que se presentan.

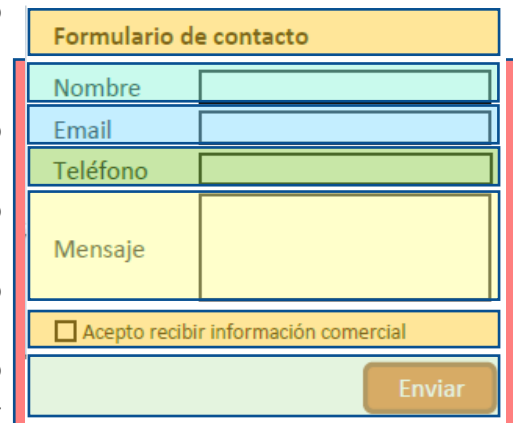
Por otra parte, el “logotipo DENTONLINE” es ya un elemento atómico, y el botón de Área clientes también lo es.

No pertenece a la etiqueta nav, puesto que tiene un formato diferente y, mientras que el menú de navegación ha de estar centrado según los requisitos, el botón Área de clientes ha de estar situado a la derecha, guardando mismo espacio entre este y el menú de navegación y el menú de navegación y el logotipo de “DENTONLINE”.

- b. **Cuerpo:** en este caso hacemos una descomposición horizontal.



- A la izquierda nos encontramos con un elemento atómico, **la imagen**.
- A la derecha (**el formulario**), nos encontramos con varios elementos atómicos que se pueden descomponer verticalmente en varias subcapas:
  - Una capa para el título “Formulario de contacto”.
  - Una capa para el formulario:
    - Una subcapa para el campo “nombre” (label + input).
    - Una subcapa para el campo “email” (label + input).
    - Una subcapa para el campo “teléfono” (label + input).
    - Una subcapa para el campo “mensaje” (label + textarea).
    - Una subcapa para el checkbox.
    - Una subcapa para el botón “Enviar”.
- c. **Pie de página (footer)**: nos encontramos ante un único elemento atómico. El formato de negrita del copyright se ha hecho con una etiqueta span (elemento HTML en línea).



2.16.- Ítem 16 (1,5 puntos).

La apariencia de la práctica desarrollada por el alumno se asemeja a la mostrada en este enunciado (colores, tamaño de los textos, etc.)

Esta sería la apariencia conseguida para la página web. En comparación con la del enunciado:



## Bibliografía.

A continuación, presento la relación bibliográfica que he consultado para la realización de este trabajo.

- [Apuntes y recursos de la plataforma de educación a distancia.](#)
- [10.2. Metadatos \(Introducción a XHTML\) \(uniwebsidad.com\)](#)
- [Flexbox, modelo de caja flexible en CSS - Eniun](#)