

PEC 2

JUAN SALINAS GÓMEZ

Pregunta 1. Sobre listas y enlaces:

1.1 ¿En qué consiste el efecto de "faux columns" o columnas falsas? Pon un ejemplo propio de su utilización.

Consiste en establecer un elemento contenedor que contenga todas las columnas que se van a mostrar y que este se estire hasta la altura de la columna más alta del interior. Esto se puede conseguir, por ejemplo, con una imagen de fondo repetida verticalmente en el elemento contenedor. Como el contenedor es tan alto como la columna más alta, su imagen de fondo da la sensación de que todas las columnas son de la misma altura.

```
<div id="contenedor">
  <div id="columna1">Columna 1</div>
  <div id="columna2">Columna 2</div>
  <div id="columna3">Columna 3</div>
</div>

# contenedor {
  float:left;
  width:100%;
}
# columna1 {
  float:left;
  width:30%;
  background:red;
}
# columna2 {
  float:left;
  width:40%;
  background:yellow;
}
# columna3 {
  float:left;
  width:30%;
  background:green;
}
```

1.2 ¿Cuáles son los cinco estados de los enlaces? Crea un ejemplo con HTML y CSS en el que se aprecie el funcionamiento de cada uno de ellos.

- No visitado (*unvisited*). El estado por defecto de un enlace cuando todavía no se ha activado o visitado.
- Visitado (*visited*). El estado de un enlace que el usuario ya ha visitado.

- Seleccionado (*focus*). Se aplica mientras el enlace está seleccionado; por ejemplo, cuando el cursor del usuario del teclado se encuentra sobre el enlace.
- Por encima (*hover*). Se aplica cuando un usuario tiene el puntero del ratón por encima del enlace pero todavía no ha hecho clic sobre él.
- Activo (*active*). Se aplica cuando el usuario activa el enlace, es decir, mientras está haciendo clic sobre él. En algunos navegadores, este estilo también se aplica cuando el enlace se ha abierto en otra ventana o pestaña.

1.3 Pon un ejemplo propio de correcta utilización de cada uno de los tipos de lista, explicando por qué es el tipo más adecuado en cada caso.

- Lista no ordenada:

```
<ul>
  <li>El señor de los anillos</li>
  <li>Guerra mundial Z</li>
  <li>Un lugar tranquilo</li>
</ul>
```

Este es un ejemplo de una lista no ordenada. Es un listado de películas para ver pero no es necesario seguir un orden para verlas, ya que no tienen relación entre sí.

- Lista ordenada:

```
<ol>
  <li>El Silmarillion</li>
  <li>El Hobbit</li>
  <li>La Comunidad del Anillo</li>
  <li>Las Dos Torres</li>
  <li>El Retorno del Rey</li>
</ol>
```

Este es un ejemplo de una lista ordenada. Se trata del listado de libros de “El Señor de los Anillos” y utilizo una lista ordenada porque se debe seguir el orden que se marca de lectura para poder seguir el argumento y la cronología.

- Lista de definiciones:

```
<dl>
  <dt>9:00</dt>
  <dd>Llegada al hotel</dd>
  <dt>9:30</dt>
  <dd>Realizar check-in</dd>
  <dt>10:00</dt>
  <dd>Encuentro en el hall</dd>
  <dt>10:15</dt>
  <dd>Salida hacia el museo</dd>
</dl>
```

Este es un ejemplo de lista de definiciones. He realizado un listado con un itinerario de un viaje. Me parece el más adecuado porque se asocia un nombre (en este caso la hora) con un valor (la actividad que se realiza).

Pregunta 2. Sobre el modelo de cajas:

2.1 ¿En qué propiedad/es del modelo de cajas se pueden aplicar valores negativos? Pon un ejemplo en el que tenga sentido hacerlo.

A los márgenes (propiedad margin).

Un ejemplo puede ser que un elemento esté dentro de un contenedor el cual tenga algo de padding, y queramos que ese elemento esté más pegado al borde del contenedor. Para conseguirlo, le aplicaríamos a ese elemento un margen negativo.



2.2 ¿Para qué sirven las propiedades min-width, max-width, min-height y max-height? Pon un ejemplo propio de su utilización.

Estas propiedades sirven para limitar el tamaño de un elemento y que conserve este tamaño.

Se pueden utilizar por ejemplo para que un elemento contenedor contenga el mismo tamaño, aunque se alargue o encoja una pantalla.

Tomando el ejemplo de la pregunta anterior, al incluirle la propiedad `max-width` al contenedor hemos conseguido que el borde no llegara hasta el final de la página, mostrándolo más acortado.

2.3 Describe el funcionamiento de la propiedad `overflow` y sus posibles valores. Investiga posibles usos de interés de esta propiedad.

Controla el comportamiento de los contenidos que no caben en su elemento contenedor. Especifica si recortar contenido, dibujar barras de desplazamiento o mostrar el contenido excedente en un elemento a nivel de bloque.

Dicho de otra forma, permite regular la visibilidad de los contenidos que sobresalen de una caja HTML y si los contenidos que sobresalen se seguirán viendo, si se ocultarán o si aparecerá una barra de scroll en el documento.

Los posibles valores de esta propiedad son:

- `visible` → Valor por defecto. El contenido no es recortado, podría ser dibujado fuera de la caja contenedora si éste sobrepasa las dimensiones de la caja.
- `hidden` → El contenido es recortado (escondido) si sobrepasa los límites del elemento contenedor y no se muestran barras de posición.
- `scroll` → El contenido es recortado y se muestran las barras de desplazamiento verticales y horizontales, se haya recortado contenido o no.
- `auto` → Se crean las barras de desplazamiento necesarias si hay contenido excedente para que éste sea accesible para el usuario.

Pregunta 3. Sobre Accesibilidad:

3.1 ¿En qué consiste el *Quirks mode*? ¿De qué forma hay que tenerlo en cuenta al diseñar nuestras páginas?

Es el modo al que pasa el navegador cuando encuentra una página con un DOCTYPE incorrecto, o sin DOCTYPE. En este modo, el navegador hace todo lo que puede para saber qué serie de reglas debe utilizar para validar el código y repara los errores lo mejor que sabe. Este modo existe realmente para permitir la visualización de las páginas más antiguas, y no se debe utilizar nunca para crear una página nueva.

En principio, este modo no nos afecta si realizamos nuestras páginas respetando los estándares web, sólo afectaría si no respetamos estos estándares o si por alguna razón olvidamos poner la declaración del DOCTYPE al inicio de nuestro fichero HTML, pero para eso tenemos la validación de documentos, que nos avisaría de un error si el DOCTYPE no estuviera.

Aun así, si después de todo esto, por algún error se nos olvida declarar el DOCTYPE o lo declaramos de forma incorrecta, se deberían tener en cuenta algunos aspectos, ya que algunos navegadores muestran ciertas características, como por ejemplo no respetando el padding aplicado con CSS a una imagen o la propiedad `max-width` es ignorada.

3.2 ¿Cuáles son las principales características que debe tener una página web para ser accesible? Pon ejemplos de cada una de ellas.

- Estructura semántica:

Al utilizar bien la estructura semántica en el HTML facilitamos la información de la página. Una correcta estructura semántica es cuando a un documento que usa correctamente las etiquetas HTML le quitamos la capa de diseño, y la estructura resultante tiene sentido por si sola.

La semántica en el HTML y HTML5 se refiere al significado o información extra que otorgan los elementos o etiquetas del lenguaje, información que define o describe el contenido, función o sección que contienen.

Esta información es lo que permite que un documento se pueda rastrear de mejor manera. Ayuda a que el contenido se muestre de manera consistente en diferentes aplicaciones y contextos, además de ser fundamental para otorgar accesibilidad y funcionalidad con tecnología de asistencia como lectores de contenido.

Por ejemplo, el utilizar una lista para realizar un menú de navegación.

- Contenido alternativo:

Se debe añadir contenido alternativo para facilitar la lectura de la página web para la gente que se encuentre en un grupo con discapacidad.

Por ejemplo, el texto alternativo proporciona una alternativa textual a los contenidos que no son texto en las páginas web. Es especialmente útil para las personas que son ciegas y se dependen de un lector de pantalla para acceder al contenido de un sitio web.

- Definir la interacción

Se trata de comprobar si el aspecto de un componente tiene sentido en relación con su función y si se ha usado correctamente una metáfora visual para el componente con el que se ha utilizado.

Por ejemplo, podría ser el ejemplo de los apuntes de las pestañas para navegar por las diferentes páginas de un sitio web, o la metáfora visual de una lupa para saber que es ahí donde debemos ir para hacer una búsqueda en la web.

Explicación de las entidades HTML y CSS utilizadas

En el documento de texto donde habrás respondido a la parte teórica de esta PEC, deberás añadir un apartado con la explicación de las entidades HTML y CSS utilizadas:

1. Sobre las entidades HTML: debe explicarse por qué se han usado las distintas etiquetas.

`<nav>` → Utilizado para agrupar el listado que marca el menú principal de navegación.

``, `` → Usado para definir el menú principal, en el primer caso de la página “artículo”, y para listas no ordenadas en el resto.

``, `` → Para hacer listas ordenadas.

`<dl>`, `<dt>`, `<dd>` → Para crear las listas de definición.

`<div class="contenedor">` → He usado esta clase para agrupar todo el contenido de la página y poder darle un estilo genérico con CSS.

`<header>` → Lo he usado para definir la cabecera de esta página, pero creo que su uso en este caso no es correcto.

`<h1>` → Para los títulos de primer nivel.

`<main>` → Utilizado para indicar el contenido más importante o principal de la página. Le he dado un id para poder tratarlo luego con CSS.

`<blockquote>` → Este elemento de la página me parecía una cita larga, y me ha parecido la mejor manera de etiquetarlo. También con el atributo id para hacerlo único.

`<p>` → Los he usado para distinguir los párrafos. Además, a varios les he añadido el atributo class para darles estilos distintos del resto.

`<q>` → Le he utilizado para darles estilos a citas cortas.

`` → Lo he usado con el atributo class para darle estilos, que se van a repetir en varias partes de la página, a una parte de texto. En un caso en “artículo” lo he usado con el atributo id para darle estilos a las comillas y que se mostraran como en el ejemplo.

`<cite lang="en">` → Para marcar pequeñas partes del texto que estaban en otro idioma.

`<a>` → Para los enlaces. Va con el atributo href para indicar el sitio con el que enlaza, y en algunos casos le he añadido el atributo Lang para indicar que es en otro idioma.

`<h2>` → Para los títulos de segundo nivel.

`<abbr title="...">` → Para la expansión de las siglas que se muestran en el texto. También lo he usado con el atributo lang porque el texto era en inglés.

`` → Para hacer énfasis en el texto y que se muestre en cursiva.

`` → Para insertar una imagen. Se le ha añadido los atributos de alt para mostrar el texto alternativo a la imagen, el title para que al pasar el ratón por encima se muestre una pequeña descripción y width y height para las medidas de la imagen.

`<aside>` → Me ha parecido que esta parte está relacionada con el contenido principal que la rodea, pero se puede considerar aparte de dicho contenido.

`<footer>` → Para indicar el pie de página.

`` → Para remarcar el texto en negrita.

* En un par de casos he insertado el código en el HTML para crear las comillas (“ ”) sin tener que utilizar etiquetas porque me parecía más costoso usar una etiqueta y luego darle estilo a un texto que no parecía importante.

2. Sobre el CSS: debe explicarse todo el CSS utilizado, incluyendo cuándo se ha utilizado y por qué y para qué se aplican los selectores y propiedades definidos.

```
body {  
  background-color: #f7e990;  
  margin: 0;  
  padding: 0;  
  font-size: 16px;  
  font-family: 'Times New Roman', Times, serif;  
  line-height: 1.2;  
}
```

Para el body he usado background-color para darle el color de fondo a la página. El margin y el padding los he puesto a 0 para que se aplique el color a toda la página y la ocupe toda el body. He aplicado el font-size para que este tamaño se use en toda la página y el font-family para usar la letra serif (o con rabito). Por último, el line-height para separar un poco las líneas.

```
.contenedor {  
  background-color: white;  
  width: 90%;  
  max-width: 970px;  
  margin: 20px auto;  
  padding: 35px;  
}
```

En la página del artículo he creado un contenedor genérico par contener todo el documento. El background-color lo he usado para que todo el contenedor se vea con el fondo en blanco. width: 90% para que no ocupe todo el espacio, max-width: 970px; para que como máximo el contenedor tenga un tamaño de 970px, margin: 20px auto; con 20px arriba y abajo para separarlo del menú de navegación, padding: 35px para separar del borde todos los elementos que están dentro del contenedor.


```
header {
  background-color: #e1d1c1;
  font-family: Arial, Helvetica, sans-serif;
  padding: 20px 20px 5px 30px;
}
```

background-color: #e1d1c1; → para que el header tenga un color distinto de fondo
font-family: Arial, Helvetica, sans-serif; → el texto se mostrará con este tipo de fuente
padding: 20px 20px 5px 30px; → para “colocar” el texto dentro del header

```
header > h1 {
  margin-top: 0px;
  font-size: 26px;
}
```

header > h1 → he usado este selector para aplicar estilo únicamente al h1 que es hijo del header

margin-top: 0px; → para quitar el margen superior que separa el texto del borde de la caja del header

font-size: 26px; → para que el texto sea más grande

```
header > p {
  margin: 0;
  font-size: 17px;
  font-weight: bold;
}
```

header > p → con este selector quería darle estilos únicamente al párrafo hijo del header

margin: 0; → eliminar los márgenes del párrafo para que no haya tanta separación

font-size: 17px; → tamaño de la fuente

font-weight: bold; → poner la fuente en negrita

```
#navegador {
  text-align: center;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  font-size: 20px;
  margin: 60px auto;
  padding: 0px;
}
```

#navegador → selector de id para un elemento único de la página, en este caso el navegador

text-align: center; → centrar el texto

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; → tipo de fuente del navegador o menú principal

font-size: 20px; → tamaño de la fuente

margin: 60px auto; → hacer que los “botones” del navegador se separen de la parte superior e inferior

padding: 0px; → utilizado para centrar los botones en la página

```
#navegador > li {
  list-style-type: none;
```

```
display: inline;
}
```

#navegador > li → selector de los elementos li que son hijos del id navegador
list-style-type: none; → eliminar los picos de la lista
display: inline; → propiedad para poner los elementos de una lista en horizontal

```
#navegador li > a {
  text-decoration: none;
  padding: 30px 70px;
  background: rgb(251,251,251);
  background: linear-
gradient(180deg, rgba(251,251,251,1) 0%, rgba(211,211,211,1) 50%);
  color: black;
}
```

#navegador li > a → selector de los enlaces que son hijos de una lista, que a su vez esta es hijo del id navegador (para seleccionar los enlaces del menú de navegación)
text-decoration: none; → para quitar el subrayado que muestran los enlaces
padding: 30px 70px; → para aumentar el tamaño de los botones
background: rgb(251,251,251);
background: linear-gradient(180deg, rgba(251,251,251,1) 0%, rgba(211,211,211,1) 50%);
 → estas dos propiedades las he usado para crear el gradiente de los botones
color: black; → color del texto del botón

```
#articulo {
  border-top-left-radius: 7px;
  border-bottom-left-radius: 7px;
}

#recursos {
  border-bottom-right-radius: 7px;
  border-top-right-radius: 7px;
}
```

Utilicé ids para los elementos de la lista que iba a ser el menú de navegación para poder darles luego la forma redondeada a los bordes de los botones de las esquinas y que el central siguiera con los bordes rectos. Eso lo conseguí con la propiedad border-radius

```
#navegador li > a:hover {
  color: white;
  background: #FAC754;
}
```

#navegador li > a:hover → usé este selector para los enlaces hijos del elemento li que a su vez son descendientes del id navegador. Con a:hover se cambia el color del botón cuando el ratón está por encima

color: white; → al pasar el ratón por encima se pone el texto en blanco
background: #FAC754; → al pasar el ratón por encima se pone el fondo amarillo

```
#descripcion {
  margin-top: 30px;
```

```
margin-bottom: 50px;
font-weight: bold;
color: #6a6957;
line-height: 1.5em;
}
```

#descripcion → para seleccionar el id

margin-top: 30px; → para aplicarle a este elemento margen superior y separarlo del elemento superior

margin-bottom: 50px; → para aplicarle a este elemento margen inferior y separarlo del elemento inferior

font-weight: bold; → aplicar negrita al texto

color: #6a6957; → color del texto

line-height: 1.5em; → para separar las líneas (ampliar el interlineado)

```
.primerPrfo{
  text-indent: 2em;
}
```

.primerPrfo → selector de clase

text-indent: 2em; → para indexar la primera línea del párrafo

```
.primerPrfo a {
  text-decoration: overline underline;
  color: blue;
}
```

.primerPrfo a → selector para los enlaces descendientes de la clase

text-decoration: overline underline; → aplicar subrayado y línea por encima al texto del enlace

color: blue; → color del texto

```
.primerPrfo a:visited {
  color: #651091;
}
```

.primerPrfo a:visited → selector de los enlaces descendientes de la clase. El atributo :visited se usa para cuando el enlace ya se ha visitado

color: #651091; → color para los enlaces ya visitados

```
cite {
  font-size: 1.2em;
  font-style: normal;
}
```

cite → selector de la etiqueta cite

font-size: 1.2em; → aumentar tamaño del texto a los elementos con esta etiqueta

font-style: normal; → eliminar el estilo cursiva que se le aplica a la etiqueta cite

```
h2 {
```

```

    color: #683b04;
    border-bottom: #683b04 dashed;
    margin-top: 40px;
}

```

h2 → selector de las etiquetas h2

color: #683b04; → aplicar color al texto de las etiquetas h2

border-bottom: #683b04 dashed; → aplicar un borde inferior de un color y un estilo simulando un subrayado como punteado

margin-top: 40px; → aplicar margen para separación del elemento superior

```

h2 + p abbr, .primerPrfo > abbr {
    font-weight: bold;
}

```

h2 + p abbr, .primerPrfo > abbr → (h2 + p abbr) selector de la etiqueta abbr que es descendiente del párrafo que es hermano adyacente del título de segundo nivel (están al mismo nivel de jerarquía h2 y p). (.primerPrfo > abbr) selector de los abbr que son hijos de la clase. Con la coma se agrupan los selectores que comparten regla.

font-weight: bold; → texto en negrita

```

#listaAxiomas {
    list-style-type: none;
    border-left: solid #683b04;
    margin: 30px 95px;
}

```

#listaAxiomas → seleccionar id

list-style-type: none; → eliminar los picos de la lista

border-left: solid #683b04; → hacer visible el borde izquierdo (con solid) y darle color

margin: 30px 95px; → desplazar la lista hacia la derecha y separarla de los elementos que la rodean

```

.nombre {
    font-style: italic;
    font-size: 1.1em;
}

```

.nombre → selector de clase. Esta clase la usé para darle estilo a dos nombres que en la imagen tenían un formato similar

font-style: italic; → poner texto en cursiva

font-size: 1.1em; → aumentar el tamaño del texto

```

q {
    quotes: '\201C' '\201D';
    font-style: italic;
    font-size: 1.05em;
}

```

q → etiqueta para una cita corta. En el código la usé para añadirle comillas y hacer el texto cursiva

quotes: '\201C' '\201D'; → añadir comillas (" ")

font-style: italic; → texto en cursiva

font-size: 1.05em; → aumentar el tamaño del texto

```
.resaltado {
  color: #683b04;
  font-weight: bold;
}
```

Esta clase la usé para darle estilo a las diferentes partes del texto que aparecen en color marrón en negrita. Con el color puse el texto en el color y con el font-weight la negrita.

```
#producto {
  border: solid #cdcdcd;
  border-width: 0 3px;
  padding: 30px;
  margin: 30px 95px 50px 95px;
}
```

Este id lo puse para la parte del texto que aparece entre comillas más grandes más centrado en la imagen. Con border conseguí mostrar los bordes en el color indicado, con border-width sólo se muestran los bordes laterales al estar el valor del superior/inferior a 0, con el padding el texto se separa del borde y con margin se centra todo el elemento

```
#iniComilla {
  float: left;
  height: 40px;
  font-size: 30px;
  margin-top: -1px;
  margin-bottom: -30px;
  padding-right: 10px;
}
```

Este id es para insertar las comillas de la izquierda más grande de lo que en realidad son y en otra posición distinta a la suya, ya que irían más arriba. Con float:left la comilla se sitúa a la izquierda del texto ocupando ese espacio, height para el tamaño que va a ocupar la comilla, font-size para el tamaño del texto (la comilla), con margin-top y margin-bottom se juega con la posición de la comilla, para que está más arriba o abajo, y con padding right se separa la comilla del texto que tiene a su derecha.

```
#finComilla {
  float: right;
  height: 40px;
  font-size: 30px;
  margin-top: -1px;
  margin-right: 560px;
}
```

Este id hace lo mismo que el iniComilla pero para la comilla del lado derecho. Se ha aplicado el margin-right para acercarla a la parte derecha del texto, ya que si no se quedaría muy separada.

```
#imgCarta {
  display: block;
  margin: 50px auto 20px auto;
}
```

Este id es para identificar la imagen de la carta. Con la propiedad display:block hace que el comportamiento del elemento sea como un bloque, y se han aplicado unos márgenes para que la imagen quede centrada.

```
#pieFoto{
  text-align: center;
  margin: 30px 95px;
}
```

Se usa este id para el texto que se muestra debajo de la foto. text-align es para centrar el texto en la página y el margin para que se vea lo más parecido posible a como está en la imagen de ejemplo.

```
#enlacesMas {
  border: 0.7px dotted #683b04;
  margin: auto 210px;
  padding: 15px 30px;
  font-family: 'Times New Roman', Times, serif;
  font-size: 18px;
}
```

Para dar estilo a este id, que engloba la parte de los enlaces del final. Se encuadra dentro de un cuadrado con la propiedad border, y con dotted se hace que la línea sea punteada. El margin es para que se centre todo el cuadro en la página y el padding para separar el borde del texto. Con los atributos de font se cambia la familia de la fuente (font-family) y el tamaño (font-size)

```
#enlacesMas a {
  text-decoration: none;
}
```

Seleccionar los enlaces descendientes del id. Se le quita el subrayado que se muestra a los enlaces con el valor none de la propiedad.

```
aside > ol {
  font-size: 16px;
}
```

Aquí podría haber aplicado el selector de id en lugar del de etiqueta. Selecciona los elementos que sean listas ordenadas (ol) e hijos de la etiqueta aside. font-size para aumentar el tamaño de la fuente.

```
footer {
  color: #525348;
  margin: auto 320px auto 280px;
}
```

Para aplicar color al texto y márgenes a la etiqueta footer y que se vea lo más similar al del ejemplo.

```
#imgAutora {
  float: left;
  margin: 35px 30px 30px 80px;
}
```

Esta imagen le puse id para poder aplicarle estilos solo a este elemento. Con el float:left la imagen se pone a la izquierda del texto y con el margin se aplican márgenes para que quede lo más alineada posible con el texto.

```
footer q {
  font-style: normal;
  font-size: 1em;
}
```

Selecciona la etiqueta q que es descendiente del footer y le quita la cursiva que le aplica cite al texto y le aumenta el tamaño.

```
footer abbr {
  text-decoration: underline dotted;
}
```

Selecciona la etiqueta abbr que es descendiente del footer y le pone un subrayado punteado al texto.

```
#general {
  margin: 30px;
}
```

Aplica un margen de 30 píxeles a los 4 márgenes. Esto es para centrar los elementos en la página de recursos.

```
.recurso {
  border: #f7e990 solid;
  margin: 20px 0px;
  padding: 20px 30px;
  max-width: 495px;
}
```

Selecciona las clases recurso y les aplica un borde (border), un margen superior e inferior (margin), un padding para separar el texto de los bordes, y un tamaño máximo (max-width) para que la caja no supere ese tamaño.

```
ol.recurso {
  list-style: lower-latin;
}
```

Cambia el pico de la lista ordenada y muestra letras en lugar de números.

```
.revistas {
  list-style-type: none;
  padding: 5px 30px;
  list-style-image: url("../img/usabilidad-icone.png");
}
```

Quita el pico de la lista de la clase revistas, le aplica un padding para desplazarla y le pone la imagen indicada en el valor de la url como imagen del pico.

```
.revistas li {
  padding: 5px 20px;
}
```

Para desplazar los elementos li de la clase revistas.

```
ul.recurso {
  list-style-type: square;
}
```

Le pone un cuadrado como pico a la lista no ordenada con la clase recurso.

```
ul.recurso li {
  padding: 5px;
}
```

Desplazar los elementos de la lista no ordenada.

```
#fRecursos, #fTecnicas {
  text-align: center;
}
```

Centra el texto de los footer con los id indicados. Con la coma (,) se aplica la regla a los distintos elementos.

```
a {
  text-decoration: none;
  color: #683b04;
}
```

Le quita el subrayado a todos los enlaces y les cambia el color al indicado.

```
footer cite {
  font-style: italic;
  font-size: 1em;
}
```

Cambia el estilo y el tamaño de la fuente a las etiquetas cite que sean descendientes de la etiqueta footer.

```
#pagRecursos {
  max-width: 720px;
  background-color: white;
  width: 90%;
  margin: auto;
  padding: 20px 35px;
}
```

Aplica el estilo general a la página recursos. Ancho máximo (max-width), color de fondo (background-color), con width se indica que no ocupe todo el espacio del que dispone, solo el 90%, con margin: auto se centra en la pantalla y con padding se separa el contenido de los bordes.

```
a abbr {
  text-decoration: none;
}
```

Se le quita la decoración (subrayado) a todas las etiquetas abbr que desciendan de un enlace.

```
#pagTecnicas {
  background-color: white;
  width: 90%;
  max-width: 920px;
```



```
margin: 20px auto;
padding: 30px 60px 15px 60px;
text-align: justify;
line-height: 1.3em;
}
```

Estilo general para toda la página de técnicas.

```
#pagTecnicas > h1 {
  text-align: right;
  margin-right: -45px;
}
```

Texto alineado a la derecha y margen en negativo para que sobrepase el padding del contenedor aplicado al título de primer nivel que es hijo del id indicado.

```
dd {
  margin-bottom: 30px;
}
```

Se le aplica un margen inferior a todas las descripciones de la lista de definiciones para separarlo de los elementos inferiores.

```
dt {
  font-size: 1.4em;
  color: #683b04;
  border-bottom: 3px dotted;
  display: inline;
  font-weight: bold;
  line-height: 2.5em;
}
```

Estilos para los términos de la lista de definiciones.

font-size: 1.4em; → tamaño del texto
color: #683b04; → color del texto
border-bottom: 3px dotted; → borde inferior punteado
display: inline; → colocar lista en horizontal
font-weight: bold; → negrita
line-height: 2.5em; → separación línea

```
dd::first-line {
  text-indent: 2em;
}
```

Desplazar la primera línea de la descripción de la lista como si fuera una tabulación

```
#requisitos {
  list-style-type: none;
  padding-left: 35px;
  margin-bottom: -15px;
}
```

Elimina el pico de la lista con el id indicado y le aplica un padding izquierdo y un margen inferior

```
#requisitos li {
  background: url(../img/check-icone.png) 5px no-repeat;
  padding: 0 50px;
  margin-right: 100px;
  line-height: initial;
}
```

Aplica una imagen como pico en la lista, y el padding y margen indicados. El line-height no hace nada, lo estuve probando para intentar desplazar el texto como en el ejemplo pero no pude.

```
dl + img {
  display: block;
  margin: 0 auto;
}
```

Aplica el mismo estilo a las imágenes que son hermanos adyacentes de las listas de definiciones. Margen superior/inferior a 0 y laterales automáticos para que se centre. Con el display: block hace que el comportamiento del elemento sea como un bloque.

```
dl + img + p {
  text-align: center;
  margin-top: 5px;
  font-size: 15px;
}
```

Aplica el mismo estilo a los párrafos que son hermanos adyacentes de las imágenes que a su vez son hermanos adyacentes de las listas de definiciones.