PEC 2

HTML y CSS: Universitat Oberta de Catalunya

Ignacio Casares Ruiz

Diciembre de 2022

Pregunta 1

Sobre la utilización de tablas, responde a las siguientes preguntas:

1. Explica el uso de los elementos `thead`, `tbody` y `tfoot` dentro de una tabla.

- `thead` se utiliza para delimitar las filas que corresponden a la cabecera de la tabla.
- `tbody` por su parte, se usa para delimitar las filas que corresponden al cuerpo de la tabla.
- `tfood` se emplea para englobar aquellas filas que comprenden el pie de la tabla.

La presencia de estas etiquetas permite una interpretación más eficaz por parte de los lectores de pantalla, por lo que son importantes a la hora de mejorar la accesibilidad de la página.

2. Las WCAG 2.1 indican que es importante el uso de la etiqueta `caption` y del atributo `scope`; explica de qué manera su uso mejora la accesibilidad de las tablas.

`caption` es una etiqueta que contiene una descripción de los elementos de que conforman la tabla. Esto permite que los lectores y los usuarios que utilicen lectores de pantalla puedan conocer de forma resumida los contenidos de la tabla de forma rápida para después decidir si examinarla con mayor deteminimiento o no.

`scope` es un atributo que permite definir qué celdas están comprendidas en el elemento `header`. Este puede completarse con los valores `col` o `row`, permitiendo delimitar de forma más eficaz qué filas o columnas son parte del encabezado de la tabla.

Pregunta 2

Sobre la utilización de formularios, responde a las siguientes preguntas:

1. Explica para qué se utilizan los siguientes atributos propios de los formularios:

- `action` y `method`: ambos son atributos del elemento `form`, el primero permite establecer una ruta en la cual se procesará la información proveída por el usuario a través del formulario. En el caso del atributo `method`, permite especificar la forma en la que se enviarán los datos recogidos por el formulario a la URL establecida a través de `action`; puede contener los valores `post`, `get` y `dialog`.

- `required`: se trata de un atributo que se aplica a los diferentes campos que conforman el formulario, obliga al usuario a completarlo para poder procesarlo.
- `minlength` y `maxlength`: son controles de valores que permiten delimitar el número de caractéres presentes en el campo. Se usan como atributos del elemento `input`.
- `pattern`: este control de valor permite limitar la entrada de datos a una serie de características definidas a través de expresiones regulares. Al igual que en el caso anterior, se usan como atributos del elemento `input`.

2. Explica para qué sirve y cómo debe usarse la etiqueta `label` dentro de un formulario. ¿Qué importancia tiene esta etiqueta en relación con la accesibilidad de los formularios?

`label` permite configurar etiquetas con texto descriptivo asociadas a las diferentes entidades del formulario. La asociación se hace mediante dos formas:

- Usando el atributo `for`, cuyo valor debe de ser igual al del atributo `id` del elemento `input` al que se quiere relacionar:

```
<label for="ejemplo">Esto es un ejemplo de etiqueta</label>
<input type="textbox" name="nombre_de_ejemplo" id="ejemplo">
```

- Incluyendo el elemento `input` dentro del elemento `label`:

```
<label>
    Esto es un ejemplo de etiqueta
    <input type="textbox" name="nombre_de_ejemplo" id="ejemplo">
</label>
```

La etiqueta `label` permite una mayor accesibilidad al poder ser identificada por lectores de pantalla y otros tipos de tecnología asistida, y permite a los usuarios acceder al campo a través de la selección de la etiqueta.

Pregunta 3

Sobre CSS, responde a las siguientes preguntas:

1. ¿Qué entendemos por modelo de cajas en CSS? ¿Qué diferencia hay entre el modelo de cajas estándar y el modelo de cajas alternativo? ¿Con qué propiedad podemos activar el modelo de cajas alternativo?

El modelo de cajas CSS hace referencia a cómo se estructuran los elementos, que se conforman como partes de una caja: el margen, el *padding*, el borde y el contenido. Estos

elementos interactúan de forma conjunta de forma que se establece una «caja» que es lo que el usuario ve en la pantalla.

El modelo de cajas se divide en dos tipos: el modelo estándar y el alternativo.

- El modelo estándar es el que forma parte de la mayoría de página de las web. Este está conformado por la asuma de la altura (atributo `height`). la anchura (atributo `width`) más la zona ocupada por el borde y el *padding*.
- En el modelo alternativo, sin embargo, estos parámetros funciona de forma diferente. Los parámetros de anchura y altura no se calculan adicionando las variables de *padding* y borde, sino que conforman una caja por sí mismos.

El modelo de cajas alternativo se activa mediante la propiedad `box-sizing`.

2. Investiga sobre las funciones `calc()`, `var()`, y `clamp()` de CSS. ¿Para qué las podemos usar? Aporta un ejemplo de código de cada una de ellas dentro de una hoja de estilos.

Las funciones `calc()`, `var()`, y `clamp()` pueden utilizarse para determianr el valor de una propiedad:

`calc()` hace un cálculo del valor a través de una operación aritmética determinada por el desarrollador. Por ejemplo, en el caso siguiente, hará un cálculo del `width` del elemento p que equivaldrá al 100% de la anchura del espacio en el que se encuentre menos 80 píxeles:

```
main p {
    width: calc(100% - 80px);
}
```

`var()` permite hacer una llamada de un valor a una propiedad personalizada. Este tipo de propiedades, a menudo definido al principio del documento CSS permiten establecer valores específicos asignados a una propiedad. Un ejemplo de su uso sería el siguiente:

```
:root {
  --first-color: #16f;
  --second-color: #ff7;
}

#firstParagraph {
  background-color: var(--first-color);
  color: var(--second-color);
}
```

`clamp()` permite asignar tres valores distintos a una propiedad: un valor mínimo, que corresponde al primer valor establecido tras el paréntesis; un valor de preferencia, que corresponde al segundo valor, después de la primera coma; y un valor máximo, que ocupa la tercera posición dentro de los paréntesis. En esta PAC se ha utilizado para que el tamaño del título principal de la página `index.html` varíe pero no sobrepase ciertos valores. El ejemplo es el siguiente, en el que el tamaño de la fuente variará dependiendo del tamaño del dispositivo (viewport: `vw`), pero que quedará siempre dentro de los valores `3.5rem` y `7rem`, al constituir estos el mínimo y el máximo, respectivamente:

```
body.main_page main div.main_title {
    display: inline-block;
    font-size: clamp(3.5rem, 6vw, 7rem);
    font-weight: bold;
    margin-bottom: 0em;
    white-space: nowrap;
}
```

Explicación de las entidades HTML y CSS utilizadas

Entidades HTML

Aspectos generales

En esta sección pasarán a describirse los elementos que son comúnes a todos los archivos `html` sobre los que se ha trabajado en el proyecto.

Todos los archivos `html` sobre los cuales se ha trabajado en la PEC incluyen una etiqueta inicial que describe el tipo de documento (html), definida como `!DOCTYPE html`. Además, en todos los casos se ha especificado el idioma a través de la etiqueta y su atributo `html lang="es"`.

Igualmente, aplicable a todas las páginas del proyecto, se ha descrito un `head` que incluye información sobre el set de caracteres (utf-8), un enlace a la hoja de estilos común del proyecto en mediante el atributo `href` y el valor `"css/styles.css"`, y un título, que varía según la página. En esta PAC se ha añadido, además, un elemento `style` que es el que nos ha permitido importar a través de la propiedad `@import` la fuente requerida en el enunciado del ejercicio a través de Google Fonts. Definiéndose, por tanto, de la siguiente forma:

```
<style>
    @import
    ↪ url('https://fonts.googleapis.com/css2?family=Norican&family=
    Work+Sans:ital,wght@0,400;0,700;1,400;1,700&display=swap');
</style>
```

El resto del contenido se ha comprendido en un elemento `body` en todos los casos. El body de la página principal obtiene una clase `main_page` para poder aplicar las reglas CSS específicas de esta. También se ha hecho así para las subsecciones `header` y `footer`, lo cual se describe en la lista posterior al párrafo siguiente. En el resto no se ha especificado clase para las mismas.

Dentro del `body` se ha dividido el contenido en tres secciones:

- Una sección `header`, que contiene, en todos los casos un menú de navegación definido a través del elemento `nav`. En el caso de la página principal, este ha recibido un atributo `class="main_header`. De forma común en todas las páginas, en esta sección se incluye_
  - Un elemento `div` con la clase `header_title_and_logo` que incluye la imagen (elemento `img`) del logo de la página, contenida en una elemento `figure`.
  - Una lista no ordenada (`ul`) como parte del elemento `nav`, que su vez contiene elementos `li` que conforman los enlaces al resto de páginas. Al enlace `a` a la página en la que se encuentra el usuario se le ha añadido la clase `current`, de forma que este pueda ser más fácilmente modificable a través de reglas CSS.
- Una sección `main`, que corresponde al cuerpo del artículo, en el que aparece la mayor parte del contenido de cada página en todos los casos.
- Una sección `footer` que hace referencia al pie de página, común en todas las páginas, que contiene enlaces vacíos a aspectos legales, la política de privacidad y la política de *cookies*. En la página `index.html` a esta se le ha aplicado el atributo `main_footer`. De forma común a todas las páginas, esta contiene dos div, uno que incluye una clase `author_in_footer` que contiene información sobre el autor de la PEC, y otro que incluye la clase `footer_nav` y que contiene las diferentes enlaces vacíos descritos anteriormente en este punto.

Adicionalmente, se han señalado todos los anglicismos con la etiqueta `i`.

Toda slas páginas han pasado el sistema de validación de W3, y se han desarrollado teniendo en cuenta los criterios de accesibilidad propuestos por esta entidad.

Entidades específicas del archivo `index.html`

Entidades del `header`

Al no incluir el logo en el encabezado, no se ha añadido un elemento `div` que lo englobe como ocurre en el resto de las páginas del proyecto.

Entidades del `main`

De forma específica para esta sección, se ha incluido un elemento `div` con la clase `main_title` que corresponde a la imagen del logo del club, así como la palabra `ritmo`, que ha sido contenida en un elemento `span` con la clase `main_words`. También se ha incluído otro elemento `div`, al que se le aplica la clase `subtitle`, que contiene la frase inferior a los dos elementos descritos anteriormente.

Entidades del `footer`

El footer se ha definido con la clase `main_footer`, como ya se ha descrito en la sección de aspectos generales. El resto de entidades es común y según lo explicado en ella.

Entidades específicas del arhcivo `inscripcion-socios.html`

Entidades del `header`

El header es común al resto de páginas exceptuando `index.html`, como se ha descrito en la sección «Aspectos generales».

Entidades del `body`

La sección `main` de esta página abre con un título principal, englobado en una etiqueta `h1`, que describe de forma general la utilidad de la secicón. `A continuación, se ha descrito un formulario, englobado como parte del elemento`form‘ y desarrollado de la siguiente forma:

- La estructura principal se ha organizado en torno a una lista no ordenada (`ul`), dentro de la cual existen elementos `li` que conforman las diferentes secciones de la misma. Cada elemento `li` contiene, a su vez. un elemento `fieldset`, que define el campo con forma de cuadrado en el que se dividen las diferentes partes del formulario. Los *fieldsets* contienen en primer lugar, en todos los casos, un elemento `legend` que sirve para añadir una etiqueta que los define en la parte superior del mismo.
- Dentro de estos *fieldsets* se han definido los diferentes elementos que conforman el formulario en sí:
  - La gran parte de los elementos se han identificado con el elemento `input`, este, a su vez, contiene:
    - Un atributo `type` que define el tipo de elemento: `radio` para los menús con varias opciones, `text` para los campos de texto de diversa índole, `email` para el campo de texto en el que se pide la dirección de correo, `tel` para aquel en el que se solicita el número de teléfono, `checkbox` para aquel en el que se pide la confirmación de la política de privacidad y `date` para aquel en el que se pide la fecha de nacimiento.[1]
    - Un atributo `id` que permite identificarlo para asociarlo a un elemento `label` (descrito posteriormente en esta sección).

[1] Se ha tenido en cuenta que algunos de estos atributos, como es el caso de `tel` o `email` incluyen sistemas de validación por defecto, por lo que no se han añadido de forma manual a través del atributo `pattern` para esos casos.

    - Un atributo `name` que acompañará a los datos que se trasmiten con el formulario y que permitirán identificarlos.
    - Atributos de validación que incluyen:
        - `required`, para aquellos campos en los que se solicita de forma obligatoria la información para poder enviarla al servidor.
        - `minlength`, en los casos en los que se establece un mínimo de caractéres para permitir la validación.
        - `maxlength`, para aquellos en los que se limita la cantidad máxima de caractéres del campo.
        - `pattern`, en los campos en los que se solicita una serie específica de caractéres. En estos casos se han incluido expresiones regulares para definir los valores de los mismos.
- Se ha utilizado el elemento `textarea` para definir el gran cuadro de texto que forma parte de la sección «Otras informaciones».
- Se ha hecho uso del elemento `select` para definir la lista de opciones desplegable de esa misma sección.
    - Las diferentes opciones de la lista han sido definides mediante la etiqueta `option`.
- Se ha empleado el elemento `label` para definir la etiqueta superior a cada campo, que incluye un atributo `for` que sirve para identificar a qué campo se refiere.
- Se han empleado saltos de línea mediante la etiqueta `br` para separar la etiqueta del campo en varios casos.

Dentro de la etiqueta `form` se ha descrito el atributo `action`, pero al no haber desarrollado la parte del código que envía la información a un archivo, se ha establecido una ruta no existente (`/action_page.php`) para este atributo. De forma similar, se ha definido el atributo `method`, con el valor `post`, pero en este caso no se hará uso de esta funcionalidad.

El párrafo que precede a la opción con *checkbox* para confirmar la aceptación de la polítíca de privacidad de ha englobado dentro de un elemento `p`. En este se ha hecho usao de la etiqueta `b` para destacar la palabra «RITMO», así como de varios elementos `span` con la clase `important` con el objetivo de darles el formato en negrita que poseían en el *mockup*.

Se ha usado, igualmente, al final del formulario un elemento `button` para describir el botón de envío de la información recogida en el mismo.

Entidades del `footer`

No existen entidades específicas para esta sección, el 'footer"se ha desarrollado tal y como se ha descrito en el apartado «Aspectos generales» en común con el resto de páginas del proyecto.

Entidades específicas del archivo `practica-el-running.html`

Entidades del `header`

El header es común al resto de páginas exceptuando `index.html`, como se ha descrito en la sección «Aspectos generales».

Entidades específicas del `main`

Esta parte del archivo se ha desarrollado de forma similar a la página principal descrita en la PAC 1 anterior de esta asignatura.

Se han utilizado las etiquetas `h1` y `h2` para identificar los distintos títulos del texto.

Se han empleado etiqeutas `p` para demarcar los párrafos de texto de la sección, así como listas no ordenadas a través de la etiqueta `ul` cuando ha sido necesario. Para el vídeo de youtube situado en la parte central del texto, se ha utilizado la etiqueta `figure`, que incluía en su interior un elemento `iframe` que define los diferentes atributos que permiten incrustar el vídeo de *YouTube*, así como un elemento `figcaption` que ha permitido añadir una corta frase inmediatamente de forma posterior a la posición del vídeo.

Para la frase final en la que se menciona el origen de la información, se ha utilizado un elemento `div` al que se le ha atribuido la clase `authorship`.

Entidades del `footer`

No existen entidades específicas para esta sección, el 'footer"se ha desarrollado tal y como se ha descrito en el apartado «Aspectos generales» en común con el resto de páginas del proyecto.

Entidades específicas del archivo `rutas.html`

Entidades del `header`

El header es común al resto de páginas exceptuando `index.html`, como se ha descrito en la sección «Aspectos generales».

Entidades del `main`

En esta sección, se ha dedsarrollado la caja central con fondo gris, que se ha englobal dentro de un elemento `div` al cual se le ha atribuido la clase `rutas_box` para poder aplicar CSS específico. Dentro, se ha dividido el texto a través de dos etiquetas `span`, a las cuales se les han aplicados las clases `sentence` y `subtitle`, a la frase principal y al subtítulo, respectivamente.

Posteriormente, se han utilizado etiquetas `p` para designar los párrafos posteriores que suceden a esta caja.

A continuación, se ha definido la tabla, de la siguiente forma:

- Se ha empleado el elemento `caption` para designar la frase que aparece en la parte superior de la misma con el texto «Nuestras rutas».
- Se ha hecho uso del elemento `thead` para designar el encabezado de la tabla, que corresponde a los elementos que se hallan en la parte superior a esta, inmediatamente por debajo del elemento `caption`; del elemento `tbody` para designar la parte principal de la misma, donde se halla la mayoría de los datos; y del elemento `tfoot` para designar la parte inferior (pie) de la misma.
- Se han empleado el elemento `tr` para designar las diferentes filas que componen la tabla. Dentro del mismo, se han utilizado:
  - Las etiquetas `th` para designar las celdas que forman parte del encabezado de la tabla. A estas se le han aplicado los atributos `scope="col"` o `scope="row"` dependiendo de si estas formaban parte de las celdas de las columnas o de las filas.
  - Las etiquetas `td` para incluir datos que no forman parte de las celdas principales del resto de la tabla. Para el elemento que corresponde al `tfoot` se ha utilizado el par atributo y valor `colspan="4"` para que este pueda ocupar la totalidad de las celdas de la última fila.
- Las imágenes de la primera fila de la tabla se han englobado dentro de elementos `figure` dentro de cada `th` de esa fila.

Entidades del `footer`

No existen entidades específicas para esta sección, el ‘footer“se ha desarrollado tal y como se ha descrito en el apartado «Aspectos generales» en común con el resto de páginas del proyecto.

Entidades CSS

Aspectos globales

Con el objetivo de facilitar la lectura de esta sección, en esta parte del documento se comentarán aquellas reglas CSS que se han considerado relevantes para con los materiales aprendidos a través de las unidades precedentes y relacionadas con la práctica en concreto.

De forma similar a las entidades HTML, el documento CSS `styles.css` ha pasado el filtro de validación del sistema ofrecida por la W3, y estas se han aplicado teniendo en cuenta los criterios de accesibilidad propugnados por esta institución.

Imagen 1: Sello de validación que aparece en la web tras la comprobación

El archivo CSS se ha incluído en una carpeta `css` a la que se accede a través de la raíz del directorio principal del proyecto.

Este archivo se ha organizado de forma que la información sea más sencilla de localizar, para este fin, se han utilizado comentarios que organizan la información por archivos, ámbitos de aplicación y temática. Un ejemplo de esto es:

```
/* Global rules start here */
```

Respecto al orden en el que aparecen. Se han dispuesto en primer lugar dos *queries* de tipo `media` con el objetivo de adaptar ciertos elementos a dispositivos de menor tamaño.[2] A continuación, se definen las reglas que afectan a los aspectos más comunes de todos los archivos `html` que conforman el proyecto, para más tarde pasar a aquellas que se aplican de forma local en cada archivo por separado. Ejemplos de reglas que se han definido como aquellas que abarcan aspectos globales son las siguientes:

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

em {
    font-style: normal;
    font-weight: bold;
}
```

[2] Esto no formaba parte de los requisitos del enunciado de la PAC, pero se ha hecho empleo de ello como forma de practicar el uso de estas reglas *at.*

En las declaraciones globales se han incluido aquellos aspectos que formaban parte de los requisitos exigidos por el enunciado de la práctica y que se han considerado que tenían una aplicación global y casi global.[3]

En la siguiente sección, que incluye reglas aplicables al elemento `body` pero igualmente de forma global se ha incluido la regla que permite la aplicación de la fuente requerida en el desarrollo del enunciado, entre otras:

```
body {
    font-family: 'Work Sans', sans-serif;
    line-height: 1.5em;
    margin: 0;
}
```

De manera similar, se han utilizado unidades `rem` para definir los valores relacionados con la propiedad `font-size`, tal y como se pedía en el enunciado de la práctica.

Entidades CSS globales relevantes en la sección `header`

Uno de los requisitos de la práctica era crear un navegador como parte del `header` que estuviera estructurado a través de un elemento `ul` pero que mostrase el contenido en una misma fila. Para conseguir este objetivo, se ha utilizado la propiedad `display`, y se han modificado `list-style-type` y `padding-inline-start` para que esta no muestre el espacio correspondiente a las viñetas de las lista:

```
body header ul {
    list-style-type: none;
    display: flex;
    padding-inline-start: 0;
    margin-top: 0;
}
```

La imagen del logo que aparece en el `header` en todas las páginas menos en `index.html` se ha dispuesto a través de la propiedad y valor `display: inline-block` de forma que esta aparezca junto al texto anexo. Esto se ha realizado mediante la siguiente regla:

```
body header figure {
    width: 2.5em;
    display: inline-block;
    margin-left: auto;
    margin-right: 0.5em;
}
```

[3] Para aquellos casos en los que se han identificado como de aplicación «casi global» se han creado reglas más específicas (siguiendo los criterios de especificidad CSS aprendidos en módulos anteriores de la asignatura) para modificar esos elementos.

Que también se ha aplicado, precisamente, al texto anexo:

```
body header .header_title {
    display: inline-block;
    font-size: 3rem;
    font-weight: bold;
}
```

Entidades CSS globales relevantes en la sección `main`

En esta sección se han aplicado multitud de reglas para adaptar los distintos elementos de forma similar a lo mostrado en los *mockups*, pero se destacará la siguiente, que hace uso de la propiedad `display` para conseguir que las imágenes y sus descendientes aparezcan centrados:

```
body main figure>* {
    margin: 0 auto;
    display: block;
    max-width: 100%;
    text-align: center;
}
```

Entidades CSS globales relevantes en la sección `footer`

Se han aplicado aquellos cambios necesarios exigidos por el enunciado del ejercicio, como la aplicación de un color específico para el fondo de esta sección. Un ejemplo entre el resto de reglas aplicadas en esta sección para conseguir esos objetivos es el siguiente:

```
body footer {
    width: 100%;
    background-color: rgb(0, 0, 0);
    color: white;
    margin: 3em auto 0 auto;
    padding-top: 1rem;
    padding-bottom: 4rem;
}
```

Entidades CSS relevantes en la página principal

Con el objetivo de adaptar el fondo a los criterios establecidos en el desarrollo del enunciado de la PAC, se utilizaron diferentes propiedades que ayudarían a que este se adaptase de forma dinámica al tamaño total de la pantalla, sin necesidad de realizar *scrolling* y de forma fluida. Estas fueron las siguientes:

```css
body.main_page {
    background-image: url('../img/home-hero.jpg');
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center center;
    background-attachment: fixed;
    display: grid;
    color: white;
    height: 95vh;
    width: 95vh;
    grid-template-rows: auto 1fr auto;
}
```

Es dentro de esta sección donde se hizo uso de `clamp()` para establecer un título cuyo tamaño variaría de forma dinámica adaptándose al tamaño del dispositivo, pero que tendría acotados ciertos valores mínimos y máximos para otorgar cierta cohesión al elemento:

```css
body.main_page main div.main_title {
    display: inline-block;
    font-size: clamp(3.5rem, 6vw, 7rem);
    font-weight: bold;
    margin-bottom: 0em;
    white-space: nowrap;
}
```

En la página de `rutas.html` se ha hecho uso del par propiedad-valor `display: grid` con el objetivo de alinear los elementos de forma que estos se adaptasen a la visualización del *mockup*. La regla utilizada para este fin es la siguiente:

```css
body main div.rutas_box {
    max-width: 100%;
    background-color: rgba(0, 0, 0, 0.1);
    text-align: center;
    display: grid;
    padding: 1em 0 1.5em 0;
}
```

Para esa misma página, en la elaboración de la tabla, se ha hecho amplio uso de pseudo-selectores con el objetivo de definir las diferentes propiedades que permitirían la adaptación al modelo *mockup*. Algunos ejemplos:

```css
body table th[scope="col"] {
    font-weight: normal;
}
```

```
body table th[scope="col"]:first-child figure {
    display: inherit;
}
```

Dentro de la página en la que aparece el formulario, se han utilizado selectores específicos para desarrollar los bordes en forma similar a lo mostrado en el *mockup*, a través de propiedades como `border-top` y `border-bottom` aplicadas a través de selectores de aplicación a menudo muy concreta:

```
body table caption {
    border-top: solid black 1px;
    padding: 0.5em 0 0.5em 0;
}

body table thead {
    border-top: solid black 1px;
    border-bottom: solid black 1px;
}
```

Para conseguir el fondo gris que solo se aplica a la mitad de las filas de la tabla, se ha hecho uso del pseudo selector `nth-child(odd)` de la siguiente forma:

```
body table tbody tr:nth-child(odd) {
    background-color: rgba(0, 0, 0, 0.1);
}
```

Por otro lado, específicamente en la sección en la que se toman datos personales del solicitante como el nombre, apellidos y el teléfono, se ha hecho empleo de pares propiedad-valor como `display: flex`, `flex-direction: row` y `flex-wrap: wrap` con el objetivo de adaptarlo al sistema de dos columnas requerido en el enunciado de la PEC:

```
fieldset.data {
    justify-content: space-between;
}

fieldset.data ul {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    align-content: center;
    justify-content: space-between;
    margin: 0 auto;
}
```

También se alteró la propiedad `width` de las celdas de esta sección para que pudiesen compartir el espacio con la otra columna de forma equilibrada:[4]

```
fieldset.data li {
    width: 50%;
}
```

Igualmente, se han definidos bordes para indicar al usuario que los campos que ha rellenado cumplen o no los requisitos de validación especificados a través de los diferentes elementos HTML que conforman la tabla:

```
input:valid {
    border: 1px solid palegreen;
}

input:invalid {
    border: 1px solid lightpink;
}
```

El resto de modificaciones corresponde a aspectos de posicionamiento, a menudos tratados a través de propiedades como `padding`, `line-height` o `width`, entre otras.

[4] En esta sección se aplicó una `media` *query* con el objetivo de mostrar solo una columna al hacer uso de la web a través de dispositivos móviles. De nuevo, esto no formaba parte de las exigencias del enunciado de la PEC.