

## PEC 3: Ud. didácticas 5

### 1. Sobre técnicas de posicionamiento y modos de diseño, responded a las siguientes preguntas:

**1.1.** Explicad brevemente qué es el "normal flow" de un documento e indicad qué métodos pueden alterar el posicionamiento de los elementos, respaldando tu explicación con ejemplos de código.

El 'normal flow' de un documento hace referencia a la forma en la se presentan los elementos de la página web de manera predeterminada. Cada elemento (en general) se predispone con un ancho de toda la página y uno detrás de otro.

Hay métodos para posicionar los elementos a nuestro deseo como son, por ejemplo:

- La propiedad **display**: block, inline, inline-block...
- Flotantes.
- La propiedad **position**: static, relative, fixed, sticky.
- La flexbox o CSS grid.

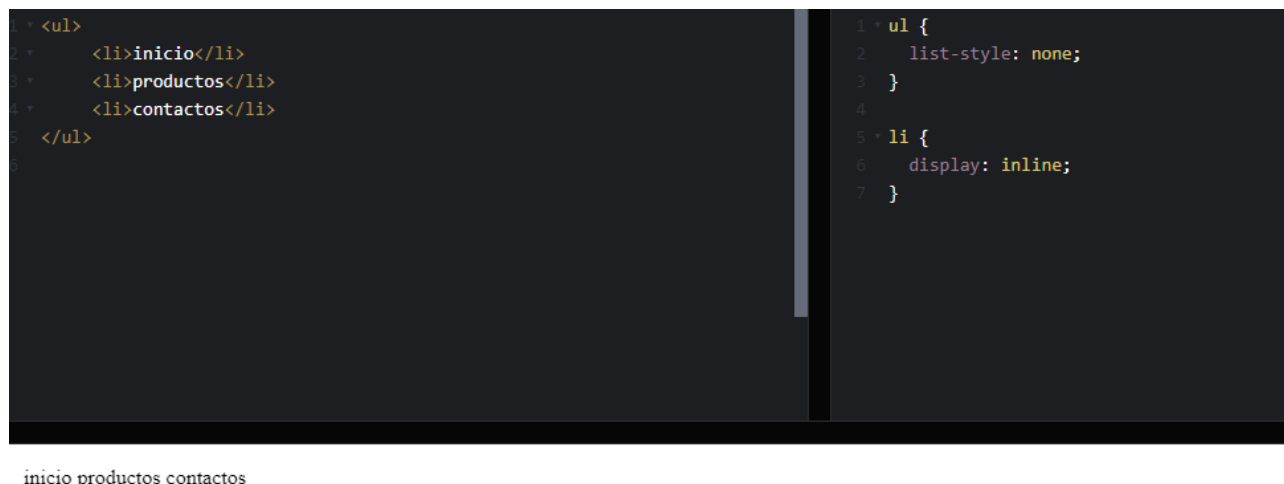
Por ejemplo, si generamos una lista, siguiendo el 'normal flow' cada check se dispondría una debajo de otra:

```
<ul>
  <li>inicio</li>
  <li>productos</li>
  <li>contactos</li>
</ul>
```



Mientras que podríamos modificar este flujo diseñándolo en nuestro documento CSS, para poder ver uno al lado del otro como suele mostrarse en las cabeceras de las páginas web:

```
li { display: inline; }
```



### 1.2. Explicad para qué sirven las siguientes propiedades del módulo CSS Flexbox: flex-direction, justify-content, align-items, flex-grow y order

**Flex-direction.** Establece el eje principal, definiendo así la dirección que se colocan los contenedores:

- Row (predeterminado): de izquierda a derecha.
- Row-reverse: de derecha a izquierda.
- Column: de arriba abajo.
- Column-reverse: de abajo a arriba.

**Justify-content.** Define la alineación a lo largo del eje principal. Ayuda a distribuir el espacio libre adicional sobrante cuando todos los elementos flexibles de una línea son inflexibles o son flexibles, pero han alcanzado su tamaño máximo. También ejerce cierto control sobre la alineación de los elementos cuando desbordan la línea.

**Align-items.** Esto define el comportamiento predeterminado de cómo se distribuyen los elementos flexibles a lo largo del **eje transversal** en la línea actual. Actúa de manera análoga a la propiedad anterior, pero en el otro eje (transversal).

**Flex-grow.** Define la capacidad de crecimiento de un elemento flexible. Acepta un valor sin unidades que sirve como proporción. Determina qué cantidad de espacio disponible dentro del contenedor flexible debe ocupar el artículo.

**Order.** De forma predeterminada, los elementos flexibles se presentan en el orden de origen. Sin embargo, esta propiedad permite controlar el orden en el que aparecen los contenedores flexibles.

### 1.3. Explicad para qué sirven las siguientes propiedades del módulo CSS Grid: grid-template, grid-column, grid-gap, place-items y grid-auto-rows.

**Grid-template.** Esta propiedad es el método abreviado para declarar las propiedades grid-template-rows, grid-template-columns y grid-template-areas en una sola línea de código para definir filas, columnas y áreas de cuadrícula respectivamente.

**Grid-column.** Declaración abreviada para grid-column-start y grid-column-end. Definen la posición y la extensión de un elemento (ítem) dentro del contenedor grid.

**Grid-gap.** Define el espacio entre las filas y/o columnas.

**Place-items.** Establece las propiedades align-items y justify-items en una sola declaración. Controlando así, la alineación transversal y longitudinal.

**Grid-auto-rows.** Especifica el tamaño de una nueva fila creada de manera implícita.

## 2. Sobre diseño responsivo, responded a las siguientes preguntas:

### 2.1. Enumerad al menos cuatro características y/o técnicas que ayudan a realizar un diseño responsivo.

- Imágenes fluidas, permiten adaptar la imagen al tamaño del dispositivo. Aplicar es bastante sencilla basta con usar la propiedad max-width: 100% en la imagen para adaptarla a un tamaño máximo del contenedor donde se ubica.
- Videos de HTML5 responsive, de la misma manera se puede ajustar el video al ancho del contenedor, variando si éste se modifica. Por ejemplo:

```
video {  
    max-width: 100%;  
    height: auto;  
}
```

- Consulta a los media, habilitan un cambio de diseño en función del tamaño de la pantalla del dispositivo, por ejemplo, se puede cambiar un menú horizontal a vertical para uso en móviles ya que a lo mejor el ancho de la pantalla de un dispositivo móvil no te permite leer o tener una experiencia de usuario adecuada. Un ejemplo sería:

```
//Mostraríamos una lista en horizontal cómo pueda ser el menú de navegación  
li { display: inline-block; }
```

```
//Mientras que para pantallas pequeñas se mostraría cada opción una debajo de otra  
@media (max-width: 600px) {  
    li { display: block; }  
}
```

- Utilizar métodos modernos de diseño de páginas web ya que son responsivos por defecto, por ejemplo, diseños flexbox o grid.

### 2.2. Explicad qué es la etiqueta “meta viewport” y qué importancia tiene en el diseño web responsivo.

Nos da información del tamaño de la pantalla del dispositivo para así, poder dar instrucciones sobre cómo controlar las dimensiones y la escala de la página. Por ejemplo:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Content="width=device-width" establece que el ancho de la página sigue el ancho del dispositivo (la pantalla).

### 2.3. Explicad qué es la filosofía Mobile First e indicad cómo se colocarían las media queries en una hoja de estilos que sigue esta filosofía

Se basa en una filosofía que plantea realizar el diseño de cualquier tipo de interfaz digital pensando primero en las pantallas de los dispositivos móviles y los smartphones para luego adaptarlas a las del ordenador.

// Contenido general en versión movil, con un padding superior moderado

```
article {  
    padding-top: 30px;  
}
```

// Contenido general en una versión movil o tablet, con un padding superior amplio

```
@media (min-width: 600px) {  
    article {  
        padding-top: 50px;  
    }  
}
```

// Contenido general en una versión escritorio, con un padding superior muy generoso

```
@media (min-width: 1200px) {  
    article {  
        padding-top: 90px;  
    }  
}
```

## Descripción del caso práctico

### 1. Aspectos generales

#### 1.1 Fuente y algún estilo global

La primera parte del archivo CSS consta del aspecto que tendrán las fuentes en rasgos generales el proyecto.

En primer lugar, lo que podemos encontrar es el vínculo a la fuente que se aplica en el proyecto, Montserrat que es extraída de: (<https://fonts.google.com/specimen/Montserrat>), los pesos de fuente que se aplican a lo largo del proyecto son: 400, 700 y 900.

En segundo lugar, podemos encontrar los colores más relevante de la marca y la fuente que se va a emplear a lo largo del proyecto, se declararían como variables:

- `--text-color-dark: #2A2A2A ;`
- `--text-color-contrast:#FFFFFF;`
- `--main-color: #79BAAB;`
- `--main-color-light: #C5E1DB;`
- `--contrast-color: #DC4A23;`
- `--contrast-color-light: #EB9781;`
- `--bg-body: #FFF7E8;`
- `--font-sans: 'Montserrat', sans-serif;`

En tercer lugar, se redefine algunas características del html (\*):

- Se establece el modelo de cajas con el que se va a trabajar: 'border-box'.
- El tipo de fuente y unos atributos de formato básico.
- Eliminamos las viñetas y subrayados predefinidos que podemos encontrar en las listas o enlaces respectivamente.
- Eliminamos el relleno y el margen que establece html de inicio.

Por último, se da formato en términos generales a los diferentes elementos de texto que podamos encontrar en el proyecto (h1, h2, h3, h4, legend, figcaption, p...). Según la parte del proyecto donde vayan apareciendo es posible que pueda verse alguna modificación.

#### 1.2 Estructura (grid, flex y media)

En esta sección se pretende definir la estructura de la página y la disposición de los elementos. Queda fuera de esta estructura la cabecera, newsletter y pie de página porque queremos un ancho máximo del viewer mientras que ya, el contenido estaría acotado a un ancho y centrado.

En primer lugar, a las páginas `re cipe.html` `shop.html` y `videos.html` se les define las diferentes  reas que se ubican en el main. Dichas  reas son: intro (haciendo referencia a la peque a introducci n antes de dar con el contenido), aside (una barra lateral de ayuda para encontrar los items, filtrar), article (donde se encontrar  el contenido principal de la p gina) y la paginaci n. Por otro lado, el  rea 'intro' y 'paginaci n' no se encuentran en la p gina video.

Además de definir las áreas, se define la disposición en las páginas tanto para pantallas grandes como pequeñas. El corte se realiza en un ancho de 600px. Hay que destacar que la sección 'aside' desaparece para pantallas pequeñas.

A continuación, se definen los 'grid' que se han utilizado, aspectos como el espaciado entre las cuadrículas, el ancho de las columnas y sobre todo para la parte de responsive, donde se pretende acomodar el número de columna en el contenido a uno o dos según las especificaciones de cada página.

Por último, definir las características flexbox que aparecen a lo largo del proyecto. Se utiliza para facilitar la disposición de los elementos dentro de contenedores centrándolos.

### 1.3 Header, newsletter y footer

Esta sección en el archivo .CSS contiene los formatos de las partes generales: header, newsletter y footer. Además, propiedades como el ancho del de estas secciones también se define aquí (class="work-area") o el color de fondo en el body y algunas características del main.

**Header**, se estructura en dos partes (logo y navegador). En esta parte del archivo de estilos se define el ancho del logo y aspectos sencillos visuales (fuentes, rellenos, márgenes, colores, sombreado de la cabecera...) y la disposición inline-block del navegador para tener una presentación horizontal. Además, se mantendría fija bajo la propiedad position: 'fixed' y z-index: '2' (para poder mantenerse en un nivel superior al resto del proyecto y se mantenga perfectamente su visibilidad).

Además, la necesidad de fijar el header se ha tenido que colocar un margen superior en el main y adaptarlo para pantallas más pequeñas (ya que el ancho del header también varía, su disposición inline-block cambia a block visualizando las opciones de navegación debajo del logo y no a la derecha como en pantallas más grandes).

**Newsletter**, el contenedor tiene un diseño gráfico particular y específico. El color de fondo viene dado en el contenedor section.news, el ancho es el 100% de la pantalla del dispositivo y una altura fija de 250px.

En esta sección se encuentra el contenido centrado mediante la propiedad flex con flex-direction: 'column'; título, descripción y formulario. El formulario son dos botones posicionados uno al lado del otro (flex-direction: 'row'). El contenido tiene una position absoluta para poder solapar con un diseño (una imagen con baja opacidad que ocuparía el 100% de la sección (el contenido tendría una capa superior al diseño (z-index: '1' pero menor que la cabecera para ocultarse en el caso de solapar con ella).

**Footer**, se clasifica en dos contenedores separados por un elemento <hr>.

- El primer contenedor tiene una descripción en el lado izquierdo y a la derecha en una lista desordenadas imágenes que llevan a las diferentes redes sociales.
- El segundo contenedor, 'policy' separa a la derecha nombre del estudiante y año del proyecto y a la izquierda los enlaces a los términos de condiciones, política de privacidad, política de cookies.

Otros aspectos particulares que se presentan en este apartado es la disposición en dos líneas del header y el footer. Para dispositivos pequeños se cambian a una disposición flex con dirección en columna. Además, otros retoques para que sea lo más parecido a la plantilla como los altos de los contenedores del footer y algún margen.

## 2. Página de inicio, index.html

El apartado principal 'main' contiene el article, que es donde se ubica el contenido de importante. Este contenido importante se divide en 4 secciones, cada una de ella tendrá la clase .flex-center (para centrar el

contenido), .work-area (menos en la primera sección, para acotar el área del trabajo a un ancho) y una clase numérica (.one, .two, .three, .four) que permita facilitar la creación de un estilo particular de la sección.

**Sección 1.** Se muestra una imagen de background y un formulario con un input para hacer una búsqueda de una receta.

**Sección 2.** Se muestra una etiqueta 'fieldset' para englobar las cuatro imágenes, serían las recetas principales o más actuales de la marca. El 'fieldset' contiene un título mediante la etiqueta 'legend'. Este englobado se aplica en otras partes del proyecto para englobar contenido.

Cada imagen se engloba en una etiqueta 'figure' que nos permite dar una leyenda a través de la etiqueta 'figcaption'. // Se podrían haber utilizado contenedores con imagen y cuerpo, pero opté por este recurso ya que se había visto en el temario.

Otra particularidad es que para pantallas grandes se visualizan cuatro imágenes mientras que para dispositivos pequeños tan solo 2 como se había definido anteriormente en aspectos generales. (no hay salto de 4 -> 3 -> 2, porque viene definido así en el proyecto).

**Sección 3.** En esta sección se añade un background acotado al área de trabajo y un cuadro, en el se engloban a través del 'fieldset' visto en la sección anterior: Título (etiqueta legend), descripción y enlace. Se cuida el centrado, el color del enlace especificado para el proyecto y su forma hover.

**Sección 4.** Esta sección consta de dos contenedores, en el contenedor 1 se muestra un título, descripción y un enlace con forma de botón (no es un botón porque no recoge datos como si de un formulario se tratara sino un enlace al apartado de recetas). El segundo contenedor contiene una imagen.

Para dispositivos pequeños se observa un contenedor encima del otro y no al lado.

### 3. Página de recetas y compra: recipe.html y shop.html

Se diseña un área para ambas pantallas que consta de una parte de introducción (título y descripción), un aside que desaparece en pantallas pequeñas (el alto es el mismo que ocupa el contenido principal más la paginación), el segundo contenido principal (a su vez también se estructura en tres o dos columnas según el tamaño de la pantalla) y una paginación.

El aside contiene diferentes inputs tipo check para poder filtrar, se divide en tres tipos englobados mediante la etiqueta 'fieldset' categories, ingredients y search en la página recipe.html y dos tipos (categories y search) en la página shop.html.

Cada figura está envuelta en un enlace que llevaría a su receta o enlace de compra, se observa el cambio del color del título de la figura con el estado hover.

### 4. Página de videos, vide.html

Consta de un título y una descripción. A continuación, el article, contiene seis contenedores que se estructura con un video de youtube, un título y una descripción. Se estructura en dos columnas para dispositivos de pantalla grande y en una columna para dispositivos pequeños.