

| | |
|---|----------|
| Preguntas teóricas | 2 |
| Pregunta 1 | 2 |
| Pregunta 2 | 3 |
| Características de la página web | 5 |
| Distribución de los archivos | 5 |
| Explicación del HTML | 5 |
| Index.html | 5 |
| recipes.html | 5 |
| shop.html | 6 |
| videos.html | 6 |
| Explicación del CSS | 6 |

Preguntas teóricas

Pregunta 1

Sobre técnicas de posicionamiento y modos de diseño, responde a las siguientes preguntas:

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.

Es la manera en la que se presentan todos los elementos de la web si no realizas ningún cambio. Para cambiar tendremos la herramienta de css "display". Tenemos diferentes opciones para display. Entre ellas inline, content, block, flex, grid,... Si vemos el siguiente código, se verá en una especie de bloques distribuidos en dos líneas con tres columnas.

```
.grid-container {  
  display: grid;  
  grid: 150px / auto auto auto;  
  grid-gap: 10px;  
  background-color: #2196F3;  
  padding: 10px;  
}
```

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

flex-direction: Nos permite indicar si el contenido se va a mostrar en forma de columna o en una línea.

justify-content: Podremos indicar si el contenido horizontalmente estará centrado, al principio o al final de un contenedor (entre otras opciones).

align-items: Indicaremos donde estará situado verticalmente el elemento dentro del contenedor.

flex-grow: Podremos indicar cuánto ocupará horizontalmente nuestros bloques.

order: Podremos ordenar los bloques para decir cual irá primero o en cualquier posición.

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: Con esto podremos definir las columnas, líneas y área de nuestro Grid.

grid-column: Podremos decir cuantas columnas ocupará cada bloque.

grid-gap: Definiremos el espacio entre bloques del Grid.

place-items: Podremos definir la posición tanto horizontalmente como verticalmente de cada bloque.

grid-auto-rows: Definiremos la altura de cada línea del Grid.

Pregunta 2

Sobre diseño responsivo, responde a las siguientes preguntas:

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

Hacer grids fluidos.

Poner el max-width al 100%, ya que así siempre se adaptará al 100% de la pantalla.

Hacer diferentes distribuciones para distintos tamaños de pantalla.

Eliminar contenido si la pantalla es más pequeña.

Explica qué es la etiqueta "meta viewport" y qué importancia tiene en el diseño web responsivo.

La etiqueta "meta viewport" le dice al navegador que el ancho de la pantalla debe considerarse el "Full width" de la página. Nos ayudará a la hora de hacer un diseño web responsive ya que podremos saber si la pantalla del usuario es de X tamaño y así acomodar la distribución a eso.

Explica qué es la filosofía Mobile First e indica cómo se colocarían las media queries en una hoja de estilos que sigue esta filosofía

En vez de distribuir la pantalla para ordenadores y después acomodarnos a móviles, primero realizaremos la web para móviles y la escalaremos a ordenadores.

Aquí tendremos un ejemplo de media queries que nos dice que cuando la pantalla sea mayor a 600px, el fondo será de color verde.

```
@media (min-width: 600px) {  
  body {  
    background: green;  
  }  
}
```

Características de la página web

Distribución de los archivos

Los archivos estarán distribuidos en tres carpetas diferentes :

- css: Contiene hojas de estilo, en nuestro caso solo será una.
- fonts: Contiene toda la información y archivos relativos a la fuente de texto de la web.
- img: Dentro de esta tendremos todas las imágenes utilizadas por la página web.
- Y todos los archivos html, en nuestro caso habrán cuatro diferentes.

Explicación del HTML

Index.html

Para empezar tendremos la primera página de nuestra web, que se accede mediante clicar a la imagen de arriba a la izquierda.

Lo primero que encontraremos será la barra de navegación (header), la cual contiene una lista no ordenada con los tres links de nuestra página. Y como el resto de partes de esta página, todos los elementos estarán dentro de un contenedor con la clase "container", que nos facilitará manipular el estilo de la página.

Lo siguiente que encontraremos será un apartado main que tiene un contenedor con la clase "divbuscador", el cual solamente contiene un elemento h1 y un input de tipo texto. Seguido de la anterior parte tendremos otra vez un ".container" que contiene un título h2 y un contenedor con la clase "home-grid". Este segundo contenedor tendrá tres partes: primero cuatro imágenes con un subtítulo (el cual es un link), la segunda parte será un contenedor con un título, dos párrafos y un link y por último tendrá una imagen y un contenedor con la información del "pick of the week".

Después como parte final del main tendremos el apartado newsletter, que básicamente será un título h2, un párrafo y un contenedor con un input de tipo texto y un botón.

Por último tendremos el footer, que será la parte de abajo de la página web. Esta contiene una parte de "follow me on" y mi información (nombre y apellidos).

recipes.html

El header, newsletter y el footer será el mismo que en el archivo index.html.

Después tendremos un contenedor con la clase “container”. Primero tendremos un título h1 junto con un párrafo.

Después tendremos un apartado de filtros con la clase “filtres”, esto contiene diferentes inputs del tipo checkbox y otro input de tipo texto.

A continuación tendremos doce imágenes con un subtítulo que será un link que no llevará a ningún sitio.

shop.html

El header, newsletter y el footer será el mismo que en el archivo index.html.

Tendremos un contenedor con la clase “container”. Primero tendremos un título h1 junto con un párrafo.

Al igual que en la página “recipes.html” tendremos un apartado para los filtros.

También tendremos nueve imágenes con un subtítulo al igual que en los anteriores casos.

videos.html

El header, newsletter y el footer será el mismo que en el archivo index.html.

Tendremos un contenedor con la clase “container”. Primero tendremos un título h1 junto con un párrafo.

En esta página tendremos 9 iframes que contienen un video de youtube y después un subtítulo con un párrafo de explicación.

Explicación del CSS

Primero cabe resaltar que enfocaremos el estilo en realizar la página web responsive, por lo tanto lo enfocaremos en Mobile First. Para eso hemos puesto este breakpoint:

```
@media (min-width: 768px)
```

En cuanto el viewport sea mayor que 768px, pasaremos a la distribución para dispositivos con pantallas mayores.

Todas las páginas tendrán un Grid que cambiará el número de columnas y filas dependiendo del tamaño de la pantalla del usuario.

Toda la página tendrá esta fuente de texto:

```
@font-face {  
  font-family: "Montserrat";  
  src: url("../fonts/Montserrat/Montserrat-VariableFont_wght.ttf")  
  format('truetype');  
  font-weight: 1 999;
```

Alejandro Criado
PEC3

}

Para todos los apartados de la página hemos utilizado tanto Grids como “display flex”.