

Práctica Formativa 2 / Mejorando nuestro portfolio con CSS

Tomaremos como referencia nuestra entrega anterior y haremos los siguientes añadidos y mejoras. La idea es pegarle una vuelta y una mejora a su trabajo para que puedan tener un portfolio mucho más profesional.

Si ya se cumplieron algunos de los requisitos obvio lo pueden dejar como están, pero se valorará también si hubo una mejora de estos.

Mejora de contenido

- Incluyan una mayor descripción de sí mismos, de los servicios que ofrecen, etc. Por ejemplo, una sección **Qué hago** con varios bloques y una sección **Sobre mí**.
- Tomen ideas de otros portfolios y amplíen su contenido en general. Este portfolio será su carta de presentación al internet y será lo que vean los reclutadores a la hora de ver y examinar sus perfiles. Cuanto más profesional y completo, más chances de destacar.

Mejora de estilos

- **Manejo de tamaños en los contenedores.** Deberán usar unidades relativas `%`, `flex` o `vw` para sus contenedores padre principales en el width y unidades como `px` o `vh` en el height, si es que trabajan con height. El uso de **Flex** soluciona mucho estos problemas del espaciado y el posicionamiento. **Usen flex.**
- Se valorará muy positivamente el correcto manejo de **elementos en bloque y elementos en línea**. Deben conocerlos bien para luego poder posicionar toda su página. No usen más saltos de línea como `
`.
- **Buen conocimiento de box model** Deberán poder manejarse cómodamente con márgenes, bordes y padding. Deberán mostrar su conocimiento para acomodar su contenido y su espaciado en el documento.
- **Centrado de elementos:** Con text-align para textos, y `margin: 0 auto;` o `flex`.
- **Manejo de fuentes:** Agregado de un `font-family` propio, descargado o de fonts google y un `font-size` que emplee unidades relativas en versión mobile.
- **Uso de iconos:** Usando font-awesome o la fuente que se prefiera. No hace falta saturar todo de íconos pero estaría bien tener un par, por ejemplo en el botón enviar del formulario, etc.
- **Manejo de enlaces con pseudoclases:** Se pueden eliminar sus estilos, pero deberían tratarse con `a:link`, `a:visited`, `a:hover`, `a:active`.
- **Usar una lista para el nav:** Usaremos dentro de nuestro contenedor `nav` una lista que mostraremos horizontalmente y que dejaremos a nuestro gusto, no se olviden usar `list-style-type` si quieren sacar los caracteres de lista.

- **Mejorar el estilado de nuestra tabla:** Esta deberá mostrar una personalización más allá del estilo por defecto. También deberán usar la siguiente pseudoclase para darle un estilo al fondo de cada fila:

```
/* Filas pares */
tr:nth-child(even) {}

/* Filas impares */
tr:nth-child(odd) {}
```

Ahora que van a incluir una descripción mayor en su web, la tabla simplemente puede listar las tecnologías que manejan o que pueden llegar a manejar.

- **Uso de z-index:** Con un uso es suficiente, deben mostrar su conocimiento del z-index en donde ustedes consideren, como ubicar un texto encima de un contenedor.
- **Uso de combinadores CSS:** Usen en su código varios, *combinador descendiente*: `div p {}` y *combinador directo*: `div > p`. De esta manera se podrán evitar usar en exceso ids y clases.
- **Manejo de opacidad:** Deben mostrar el uso de opacidad en algún elemento de su portafolio. Este puede usarse, por ejemplo, para oscurecer algún fondo y resaltar el texto.

Responsividad

Deberán hacer que su página sea responsiva y recolocar los elementos acorde al tamaño de la pantalla. Su proyecto debe llevar 4 tamaños máximos de pantalla.

```
/* Tablet horizontal */
@media (max-width: 1080px) { }
```

```
/* Tablet vertical */
@media (max-width: 768px) { }
```

```
/* Mobil 1 */
@media (max-width: 480px) { }
```

```
/* Mobil 2 */
@media (max-width: 375px) { }
```

Recordatorios

- Sean prolijos la estructura de su proyecto, un archivo `index.html`, otro archivo `README.md`, una carpeta `css` con su hoja de estilos `css/styles.css` y una carpeta `assets` con sus imagenes `assets/img/miimagen.jpg`.
- Es importante que sean estrictos y prolijos con los nombres de los ficheros, desde sus archivos `html` y `css` hasta sus imagenes.
- Comprueben que tienen bien las rutas, sepan navegar entre carpetas. `../` nos permite retroceder una carpeta atrás. Para avanzar podemos poner directamente el nombre de la carpeta o `./`, por ejemplo

desde `index.html` la ruta podría ser `css/styles.css` o `./css/styles.css`. Recurren a la ayuda de navegación que les va tirando Visual Studio Code para encontrar los archivos. *Recuerden que cuando vayan a poner una ruta, fíjense desde que carpeta están, si en la raíz o en una carpeta superior y naveguen desde ese lugar.*

- Recuerden tener un código limpio, bien comentado y bien indentado. Debe ser limpio, claro y fácil de entender. Separen los distintos bloques de su CSS con caracteres bien distinguidos como por ejemplo:

```
/*//////////////////// AJUSTES GENERALES////////////////////*/  
  
/* _____Portada_____ */
```

- Incluyan en un archivo `README.md` donde hagan una pequeña descripción de su proyecto usen `#` para el título. Este se puede crear por defecto cuando se crea un nuevo repositorio en github.
- No es obligatorio, pero se valorará también incluir una imagen de muestra en nuestro README, este puede ser una captura de pantalla de nuestra web, por ejemplo, `sample.png` y se puede enlazar en nuestro README como ``

Recursos

- En la web, los tiempos de carga son cruciales, recuerden no subir archivos muy pesados, idealmente menos de 500kb cada imagen. Pueden usar [compresores de imagen](#) o herramientas para recortar, comprimir y redimensionar como [picresize](#).
- [Documentación Markdown](#).
- [Guía visual flexbox](#).