

Memoria práctica 1 PW: Diseño de un sitio web para un museo

Rocío Barragán Moreno

Curso 2023-2024



UNIVERSIDAD DE GRANADA

1. Estructura general de la práctica	3
2. Desarrollo de los distintos sitios web del museo	4
2.1 Encabezado (header) y pié de página (footer)	4
2.1.1 Estructura HTML	4
2.1.2 Estructura CSS	5
2.2 Página de alta de usuarios, altausuarios.html	8
2.2.1 Estructura HTML	8
2.2.2 Estructura CSS	9
2.3 Página de exito de alta de usuarios, alta_exitosa.html	10
2.3.1 Estructura HTML	10
2.3.2 Estructura CSS	10
2.4 Página principal, index.html	11
2.4.1 Estructura HTML	11
2.4.2 Estructura CSS	11
2.5 Página de colecciones, coleccion.html	14
2.5.1 Estructura HTML	14
2.5.2 Estructura CSS	16
2.6 Página de obra, obra1.html	19
2.6.1 Estructura HTML	19
2.6.2 Estructura CSS	20
2.7 Página de visita, visita.html	22
2.7.1 Estructura HTML	22
2.7.2 Estructura CSS	22
2.8 Página de Exposiciones, exposiciones.html	24
2.8.1 Estructura HTML	24
2.8.2 Estructura CSS	24
2.9 Página de información, informacion.html	26
2.9.1 Estructura HTML	26
2.9.2 Estructura CSS	28
2.10 Página de experiencias, experiencias.html	30
2.10.1 Estructura HTML	30
2.10.2 Estructura CSS	31
2.11 Página de contacto, contacto.html	33
2.11.1 Estructura HTML	33
2.11.2 Estructura CSS	33
3. Aspectos innovadores	34
3.1 Transiciones	34
3.2 Cambios de colores progresivos	36
3.3 Borde blanco en título	37

1. Estructura general de la práctica

En esta práctica, se ha propuesto el desarrollo de un sitio web para un museo.

Este desarrollo sólo consistirá en la implementación del diseño en sí, con HTML y CSS, lo que se conoce como el Front End de una app. Es decir, no poseerá ninguna funcionalidad relacionada con el lado del servidor.

Para desarrollar esta práctica, tuve varias alternativas en mente, pero al final me decanté por una que le podría dar un toque interesante y original, y es usar las pinturas que hace mi tía María para representar las distintas obras de arte que hay en el museo, ya que me parece que son bastante bonitas y siempre la he admirado como artista, así que, qué mejor momento que este para mostrar sus obras en un proyecto de clase.

En cuanto el diseño en sí, he intentado ajustarme a todos los criterios que se nos han indicado, que ahora se irán comentando en detalle a lo largo de la memoria, pero he de destacar que me he ceñido a una paleta de colores en sí, que va variando entre el verde y azul claritos, y el cian y violeta. Esto es porque para el fondo de la página web, he utilizado una de las obras de mi tía y esta se combina muy bien con dichos colores.

Con respecto al nombre del museo, me he centrado en las principales temáticas que dibuja mi tía, que son: Animales y Mujeres, de ahí su nombre, WildFemArte.

Cabe destacar también, que en mi directorio pe1, hay 2 carpetas: imágenes, en las cuales se encuentran todas las imágenes utilizadas para la práctica; y estilo, en donde se encuentran todos los ficheros CSS utilizados.

Este es el diseño de mi página principal, index.html, sobre el que se basarán el resto de páginas:



2. Desarrollo de los distintos sitios web del museo

Para esta práctica, se nos pedían varios sitios web que tenían que figurar sí o sí en nuestra página web del museo.

A continuación iré desarrollando una a una su diseño e implementación, haciendo hincapié en el por qué de la estructura que le he dado, el CSS utilizado, etc. Para que así quede claro y reflejado cómo se ha ido estructurando cada una de las páginas.

Para cada uno de los sitios web, he creado un fichero HTML y un fichero CSS, para que así el código quede más organizado y no se haga un solo archivo CSS con muchísimas líneas de código, pues puede quedar poco legible e intuitivo.

Todos los sitios web del museo tienen en común el encabezado (header) y el pie de página (footer), por lo que antes de adentrarme a explicar todas las páginas, me centraré en la explicación de cómo he desarrollado el header y el footer y ya a continuación iré comentando el resto de páginas web

2.1 Encabezado (header) y pie de página (footer)

2.1.1 Estructura HTML

La estructura HTML del header es la siguiente:

```
<header>
  <section id="logo">
    
    <h1>Museo WildFemArte</h1>
  </section>

  <form>
    <h3>Inicio de sesión</h3>

    <label for="Email">Email</label>
    <input type="email" id="Email" name="emailusuario" placeholder="Introduzca su email" required>

    <label for="Contraseña">Contraseña</label>
    <input type="password" id="Contraseña" name="contraseñausuario" placeholder="Introduzca la contraseña" required>

    <input type="submit" id="InicioSesion" name="Iniciar sesión">
    <a href="altausuarios.html">¡Regístrate aquí!</a>
  </form>

  <nav>
    <a href="index.html">Inicio</a>
    <a href="colección.html">Colección</a>
    <a href="visita.html">Visita</a>
    <a href="exposiciones.html">Exposiciones</a>
    <a href="informacion.html">Información</a>
    <a href="experiencias.html">Experiencias</a>
  </nav>
</header>
```

Básicamente, he dividido el header en 3 apartados:

1. Sección de logo, para englobar el logo del museo y su título.

2. Formulario de inicio de sesión, en el que el usuario podrá iniciar sesión en el museo si ya es cliente, o darse de alta si aún no lo es, en el enlace de “Regístrate aquí”.
3. Menú de navegación para ir navegando por las páginas principales del museo.

La estructura HTML del footer es la siguiente:

```
<footer>
  <a id="contacto" href="contacto.html" title="Contacta con nosotros aquí">Contacto</a>
  <a id="practica" href="como_se_hizo.pdf" title="¿Cómo se hizo la práctica?">Cómo se hizo</a>
</footer>
```

2 enlaces con ids para hacer referencia a cada uno de ellos, por un lado a la página de contacto y por otro lado a esta memoria.

2.1.2 Estructura CSS

Para que el logo del museo y el título queden alineados, y en la misma línea, he usado display flex, y align-items center, ya que sino, la imagen quedaba desquadrada. Además a esa sección le he proporcionado un ancho específico para que luego el formulario de registro e inicio de sesión pueda estar en la misma línea que estos, pero más a la derecha

```
#logo {
  display: flex;
  align-items: center;
  width: 70%;
}
```

Para que los elementos del formulario queden uno debajo del otro, he utilizado display flex y flex direction column y así se disponen de esa forma en el contenedor. Además, para que quedase a la altura del título y logo, he ido ajustando los márgenes. También he ajustado el padding para que los elementos no queden tan pegados al borde del formulario y le he dado un ancho no muy grande al formulario ya que prefería que fuese más pequeño.

```
header form{
  width: 19%;
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-left: auto;
  margin-top: -200px;
  margin-bottom: 10px;
  background-color: #rgba(28, 108, 138, 0.5);
  padding-top: 5px;
  padding-bottom: 10px;
  border-style: ridge;
  border-color: #rgba(28, 108, 138, 0.5);
  border-width: 10px;
}
```

Para la estructura del menú de navegación, también he usado display flex, y después justify-content space around, para que los elementos se extiendan por todo el ancho del contenedor y aparecerán así unos al lado del otro.

```
header nav {  
    display: flex;  
    justify-content: space-around;  
    background-color: #rgba(28, 108, 138, 0.8);  
}
```

Cabe destacar que también he establecido un padding de 5px entre cada elemento para que así se encuentren más separados.

```
header nav a {  
    padding: 5px;  
    font-family:'Courier New', Courier, monosp
```

Para el footer simplemente he ajustado el margen de arriba y hacer que los enlaces se encuentren en el centro de la página:

```
footer{  
    margin-top: 50px;  
    text-align: center;  
    background-color: #rgb(207, 224, 240, 0.8);  
}
```

El resto de reglas, tanto del header como del footer, son reglas relacionadas con el estilo de letra, fondo, tamaño de letra, ajustando los márgenes de cada elemento etc. Por lo que no considero muy relevante explicar eso también, ya que lo más tedioso de CSS es situar los elementos en el sitio correspondiente, y una vez hecho eso, es dar el diseño y formato que uno prefiera. Por ello, en todos los CSS explicaré tan solo lo más relevante de cada uno, además que hay algunos ficheros, como este, que me ha ocupado más de 100 líneas, y la gran mayoría es de diseño.

Para finalizar el CSS de header y footer, se nos pedía que tanto index.html como colección.html fuesen responsive, pero yo decidí hacer también responsive el header y el footer:

```
@media (max-width: 1000px) {  
    #logo {  
        flex-direction: column;  
        align-items: center;  
        width: 100%;  
    }  
  
    header form {  
        width: 100%;  
        margin-top: 0;  
    }  
  
    header nav {  
        flex-direction: column;  
    }  
  
    header nav a {  
        padding: 10px 0;  
    }  
  
    #contacto, #practica {  
        margin-left: 10px;  
    }  
}
```

Lo he hecho para que, para pantallas de 1000px o menos, se haga responsive, de manera que el logo y la imagen irán uno debajo del otro, ocupando el 100%, el formulario debajo, ocupando otro 100%.

El menú de navegación quedará de tal forma que cada uno de los elementos quedará debajo del otro y se irá ajustando el padding de cada uno.

Finalmente, los elementos del footer siempre mantendrán un margin left concreto, para que no se solapen.

Por tanto, el diseño del header es el siguiente, tanto a tamaño de pantalla normal, como responsive:



Y el del footer así, tanto normal como responsive:



2.2 Página de alta de usuarios, altausuarios.html

Cuando un usuario no es miembro del museo, en el formulario de Inicio de Sesión del header, tal como se ha indicado en el apartado anterior, hay un enlace llamado “Regístrate aquí” en el que el usuario puede darse de alta para ser un miembro del museo. Para ello, he implementado un formulario que abarca distintos campos a completar.

2.2.1 Estructura HTML

La estructura HTML de esta página es la siguiente:

```
<main>
    <h2>Formulario para ser un miembro de nuestro museo</h2>

    <form action="alta_exitosa.html" method="post">
        <label for="Nombre">Nombre</label>
        <input type="text" id="Nombre" name="nombre" placeholder="Nombre" required>

        <label for="Apellidos">Apellidos</label>
        <input type="text" id="Apellidos" name="apellidos" placeholder="Apellidos" required>

        <label for="Correo">Correo Electronico</label>
        <input type="email" id="Correo" name="correo" placeholder="Correo Electronico" required>

        <label for="Contrasena">Contrasena</label>
        <input type="password" id="Password" name="contras" placeholder="Contrasena" required>

        <label for="Telefono">Telefono</label>
        <input type="tel" id="Telefono" name="telefono" placeholder="Numero de telefono" required>

        <label for="Nacimiento">Fecha de Nacimiento</label>
        <input type="date" id="Nacimiento" name="nacimiento" required>

        <datalist id="conocermuseo">
            <option value="Amigos/as">
            <option value="Familiares">
            <option value="Redes Sociales">
            <option value="Otros">
        </datalist>
        <label for="texto">¿Cómo conociste nuestro museo? </label>
        <input type="text" id="texto" list="conocermuseo" required>

        <input type="submit" value="Registrarse">
    </form>
</main>
```

Se deben completar obligatoriamente campos básicos de registro, ya sea nombre, apellidos, correo, password etc.

Para cada campo he implementado un label con ids únicos.

Se puede apreciar que no he cubierto todos los tipos de labels que pueden haber, pero eso es debido a que, en otro formulario que se verá más adelante (el de experiencias) sí los uso, así están igual de equilibrados.

Este formulario redirigirá, en el action, a una página llamada alta_exitosa.html, que hará referencia a que, si los campos están correctamente completados, el usuario se dará de alta en el museo y por tanto será un miembro de este.

2.2.2 Estructura CSS

Para el diseño de este formulario, quería que estuviese centrado y que además no ocupase todo el ancho de la página, por ello establecí un ancho específico y ajusté los márgenes y padding para que el resultado fuese más estiloso y adecuado.

Usé un margin 20px auto para que, tanto arriba como abajo tuviese un margen de 20px y que con auto centre el formulario en el centro de la página.

```
main {  
    max-width: 600px;  
    margin: 20px auto;  
    padding: 40px;  
    border-radius: 20px;  
    background-color: #rgb(207, 224, 240, 0.8);  
}
```

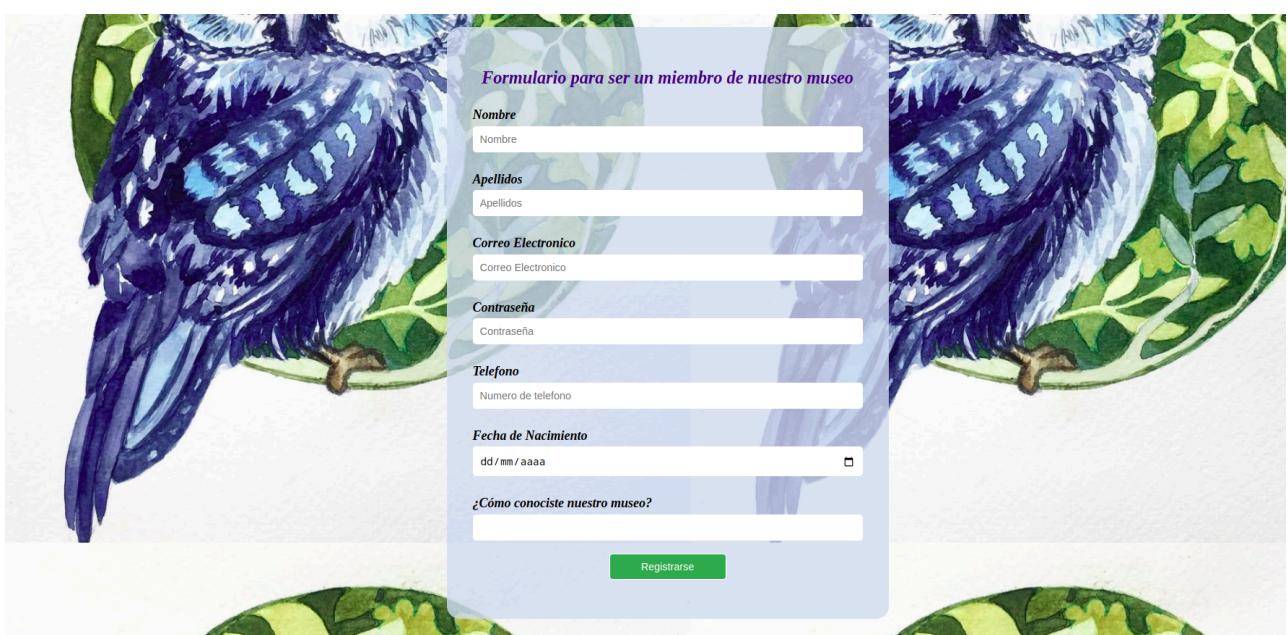
Para que todos los elementos quedasen uno debajo del otro, he usado display flex y flex-direction column:

```
main form {  
    display: flex;  
    flex-direction: column;  
}
```

El resto de reglas CSS están ligadas al estilo de los campos, letras etc, pero en concreto quiero destacar esta regla, ya que lo que hace es que, cuando pasas el ratón por encima del botón “Registrarse” (que es de tipo submit), el color del botón cambia a un verde más oscuro.

```
main form input[type="submit"]:hover {  
    background-color: #green;  
}
```

El diseño de esta página finalmente quedaría así:



2.3 Página de éxito de alta de usuarios, alta_exitosa.html

Esta página simplemente consiste en informar al usuario que se ha dado de alta en el museo, siempre y cuando los datos introducidos sean correctos, y que este ya no sea miembro del museo.

2.3.1 Estructura HTML

La estructura que he seguido para esta página es la siguiente:

```
<main>
  <h2>¡Registro exitoso!</h2>
  <p>Te has registrado correctamente en nuestro museo. ¡Bienvenido/a!</p>
</main>
```

Simplemente un título h2 con un párrafo indicando que se ha registrado correctamente.

2.3.2 Estructura CSS

Para el diseño de esta página, lo único que he hecho es darle un width del 50% y centrar el texto, y a partir de ahí, ir diseñando tanto el título como el párrafo de bienvenida.

```
main {
  text-align: center;
  background-color: #rgb(207, 224, 240, 0.8);
  width: 50%;
  margin-left: 20%;
```

El resto de reglas no son tan relevantes pues es solo diseño de cada componente.

El diseño quedaría de la siguiente manera:



2.4 Página principal, index.html

Esta es la página principal de la página web, por tanto es una página de bienvenida por así decirlo. El requisito que se nos daba para esta página es que fuese responsive, es decir, que se adaptara a todos las pantallas (móviles, tablets, ordenador etc.), no solo a la del ordenador.

2.4.1 Estructura HTML

La estructura que he seguido en esta página es la siguiente:

```
<main>
  <section id="bienvenida">
    <h1>¡Bienvenidos al Museo WildFemArte!</h1>
    <p> En este maravilloso lugar, encontraréis una colección
        única de obras de arte contemporáneo, especialmente dedicadas a celebrar la belleza y
        la diversidad de las mujeres, los animales y los paisajes.<br>
        Espero que disfruten.
    </p>
  </section>

  <section id="obrainvitada">
    <h4>LO MÁS DESTACADO ...</h4>
    <p id="autora"> María BE</p>
    <p id="diseños"> mujeres, animales, paisajes ...</p>
  </section>
</main>
```

He dividido el main en 2 secciones, una sección de bienvenida y otra con la artista más destacada del museo, así como sus principales obras. Dentro de la sección de obra invitada, he puesto dos elementos p con 2 ids para poder editar ambos de manera distinta.

He hecho 2 secciones para distribuirlas en zonas distintas en el main.

2.4.2 Estructura CSS

El diseño de esta página lo he organizado de tal forma que, por un lado he colocado la sección de bienvenida y le he dado estilo, y por otro lado lo mismo pero con la sección de obra invitada.

En la sección de bienvenida, he hecho un display flex para que se coloquen los elementos uno al lado del otro en el contenedor, y le he establecido un width del 50% para que no ocupe todo el ancho de la página. Además, he hecho un text-align center para que los elementos se alineen centralmente.

```
#bienvenida{
  display: flex;
  width: 50%;
  text-align: center;
  margin-left: 100px;
  margin-top: 20px;
  background-color: #rgba(28, 108, 138, 0.75);
  border: solid;
  border-style: dashed;
  border-radius: 30px;
  color: white;
}
```

En la sección de obra invitada, he hecho un display block para que los elementos se coloquen debajo los unos de los otros, y he ido ajustando los márgenes de tal forma que no quede pegada a la sección de arriba, sino un poquito más abajo, y además que quedase más a la derecha. También le he establecido un ancho determinado para que no ocupe todo el ancho de la página:

```
#obrainvitada{  
    width: 30%;  
    display: block;  
    margin-top: 30px;  
    margin-left: 49%;  
    border: solid;  
    border-width: 10px;  
    background-color: #rgb(122, 187, 122, 0.6);  
    border-radius: 30px;  
    border-color: #rgb(122, 187, 122);  
    border-style: dotted;  
}
```

El resto de reglas CSS van relacionadas con el estilo personalizado de cada elemento.

Por último, quedaría indicar cómo he hecho el diseño responsive de esta página:

```
@media (max-width: 1000px) {  
    #bienvenida {  
        width: 100%;  
        margin-left: 0;  
    }  
  
    #bienvenida h1 {  
        font-size: 30px;  
    }  
  
    #bienvenida p {  
        font-size: 20px;  
    }  
  
    #obrainvitada {  
        width: 100%;  
        margin-left: 0;  
        margin-top: 20px;  
    }  
  
    #obrainvitada h4 {  
        font-size: 16px;  
    }  
  
    #autora {  
        font-size: 30px;  
    }  
  
    #diseños {  
        font-size: 20px;  
    }  
}
```

He utilizado media queries, como anteriormente, en el que para pantallas de menos de 1000px el diseño sea adaptable. Lo más destacable es que tanto las secciones de bienvenida como obra invitada ocupan un ancho del 100%, por lo que estarán una debajo de la otra, y les he aplicado también un margin left 0 para que no tengan espacio por la izquierda.

El diseño final de esta página, quedaría de la siguiente manera, tanto normal como responsive:



2.5 Página de colecciones, colección.html

En esta página, se presentan las distintas obras que tendrá el museo. Debe ser responsive también

Para esta página se nos propone un diseño muy específico, en el que, a la izquierda del main, habrá un menú en el que se podrán filtrar las obras en función del autor/a, temática y época. A la derecha, irán todas las obras, de 3 en 3, y al final de la página habrá un botón de siguiente (o anterior en caso de estar en la página siguiente), con el que se avanzará a la siguiente página de colección, que contendrá otra tanda de colecciones.

Para representar las colecciones, he usado un total de 27 obras: 9 de mujeres, 9 de animales y 9 de paisajes.

Esto conlleva a que, en la página original de colecciones, es decir, colección.html, estarán estas 27 obras. Por lo que habrá 2 páginas siguientes: siguiente_generales.html y siguientes_generales2.html.

En el resto de páginas filtradas, he asignado 3 obras de cada tipo a cada filtro, es decir, 3 obras de mujeres para cada autor, 3 obras de animales para cada autor, 3 obras de paisajes para cada autor ... Y así sucesivamente con todos los filtros. Lo que resulta en que cada filtro solo tendrá 9 obras. Por tanto, solo he hecho una página de siguiente para los filtros en el que dichas obras aparecen en orden inverso, y así se notan los cambios de una página a otra.

La página de siguiente la he llamado de tal forma que, si nos metemos en colección_mujeres.html, la página siguiente se llamará siguiente_mujeres.html

Una vez explicado esto, dado que la estructura de todos los documentos es la misma que en colección.html, me ceñiré a explicar esa. Cabe destacar también que he aplicado el mismo CSS a todas las páginas de colecciones, pues al final el diseño es el mismo para todas.

2.5.1 Estructura HTML

La estructura seguida es la siguiente:

```
<main id="colecciones_generales">
  <aside>
    <nav>
      <section>
        <h3>Autores</h3>
        <a href="colección_mariabe.html">Maria BE</a>
        <a href="colección_evabe.html">Eva BE</a>
        <a href="colección_noahbe.html">Noah BE</a>
      </section>

      <section>
        <h3>Temas</h3>
        <a href="colección_paisajes.html">Paisajes</a>
        <a href="colección_mujeres.html">Mujeres</a>
        <a href="colección_animales.html">Animales</a>
      </section>

      <section>
        <h3>Épocas</h3>
        <a href="colección_otoño.html">Otoño</a>
        <a href="colección_primavera.html">Primavera</a>
        <a href="colección_verano.html">Verano</a>
      </section>
    </nav>
  </aside>
```

```

<section id="obras">
    <h2>Obras de arte de toda la colección</h2>

    <section id="todas">
        <article>
            <figure>
                <a href="obra1.html">
                    
                </a>
                <figcaption>
                    <h3>Mujer Libre</h3>
                    <p>Autora: <cite>María BE</cite></p>
                    <time datetime="2020-01-22">Fecha: <cite>22-01-2020</cite></time>
                </figcaption>
            </figure>
        </article>

        <article>
            <figure>
                <a href="obra1.html">
                    
                </a>
                <figcaption>
                    <h3>Café otoñal</h3>
                    <p>Autora: <cite>Eva BE</cite></p>
                    <time datetime="2020-06-27">Fecha: <cite>27-06-2020</cite></time>
                </figcaption>
            </figure>
        </article>
    </section>

```

```

<section id="siguientes">
    <nav>
        <a href="siguientes_generales.html">Siguiente</a>
    </nav>
</section>

```

En la primera foto, tenemos un aside, para hacer referencia a que el menú de filtro de obras irá a un lado del main, en este caso a la izquierda. Dentro de este, con la etiqueta nav, indicaremos que es un menú de navegación en el que se podrá acceder a las obras filtradas por autor, tema y época.

En la segunda foto, tenemos una sección cuyo id es obras, haciendo referencia a las obras de arte. Dentro de este, tenemos el título de la colección y a continuación otra sección con el id “todas”, que hace referencia única y exclusivamente a las obras, sin incluir el título principal. Esto lo he hecho de esta manera, debido a que sino, no podía poner el título general arriba y debajo todas las obras de 3 en 3, tal como le comenté en tutoría. Lo intenté de varias maneras y no me salía, así que usted me dijo que lo dejase así ya que ya lo habíamos hablado en la tutoría.

Dentro de esta sección, tenemos distintos elementos article que engloban a cada una de las obras, y en cada uno de estos, englobo a la obra con el elemento figure, que a su vez engloba la imagen, en la cual si clickeas te redirige a obra1.html (se explicará más adelante), y el figcaption con datos relevantes de la obra. Se puede apreciar que solo se ven 2 artículos en la foto, pero en realidad son 9, pero para no hacer una foto demasiado extensa he mostrado solo ese trozo del código.

En la tercera foto tenemos una última sección, dentro de la sección de obras, la cual abarca un elemento nav, para indicar que se navegará a la página siguiente (o anterior, dependiendo de en cuál se esté). También se ha aplicado otra sección ahí para que el enlace a siguiente o anterior quede abajo de las obras.

2.5.2 Estructura CSS

El diseño de esta página quizás haya sido el más tedioso de todos. A continuación, voy a comenzar a explicarlo:

En primer lugar, dado que al main le puse un id llamado “colecciones_generales”, hice un display flex de todos los elementos que contenga, para que se dispusieran uno al lado del otro. De esta forma, me quedaría el aside a un lado y al lado las colecciones, con el título arriba de estas y los enlaces de siguiente y anterior debajo de estas, tal como se explicó antes.

```
#colecciones_generales{  
    display: flex;  
}
```

Ahora vamos con el menú de navegación. Le di un ancho específico, muy estrecho, ya que al ser un aside y un menú, es preferible que fuese más estrecho y las obras ocupasen el resto. El texto de cada sección del menú de navegación quería que quedase centrado en el contenedor nav, por lo que hice text-align center.

También, para cada elemento a de cada sección del menú, quería que estuviesen uno debajo del otro, por lo que utilicé display block.

```
aside{  
    width: 15%;  
    margin-top: 20px;  
    background-color: #antiquewhite;  
}
```

```
nav section{  
    text-align: center;  
    margin-bottom: 100px;  
}
```

```
nav section a{  
    display: block;  
    font-size: 25px;  
    font-style: italic;  
    text-decoration: none;  
}
```

A continuación, con todas las obras. A esta sección le di un width mucho más grande, ya que así se verían mejor.

```
#obras{  
    width: 85%;  
    margin-top: 20px;  
    background-color: #rgb(207, 224, 240, 0.8);  
}
```

A continuación, empecé a modelar la sección de “obras” y también el de “todas”.

Para ello, situé todos sus elementos (título, obras y enlace a siguiente o anterior) centrados con text-align center, display flex para aplicarles flex wrap, para que los elementos puedan estar en múltiples líneas y justify content space around para que se distribuyan el espacio que hay entre ellos, teniendo entre todos el mismo espacio entre ellos. Por tanto, con esto, estoy diseñando una sección flexible que centra el texto y distribuye los elementos en distintas líneas.

```
#obras, #todas{  
    text-align: center;  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: space-around;  
}
```

Ahora, voy a explicar cómo he distribuido las obras de 3 en 3. Para ello he usado la orden flex 1 0 30% en cada artículo. Cada uno de estos números indica un atributo de flex. El 1 indica “flex-grow”, esto quiere decir que el elemento puede crecer pero solo hasta el ancho que tiene disponible. El 0 indica “flex-shrink”, lo que quiere decir que no se va a disminuir más de su tamaño original, será el tamaño mínimo que mantendrá. Y por último 30% hace referencia a flex-basis, que le pone un 30%, por tanto todos ocuparán 30% y se dividirán en líneas de 3.

```
#todas article{  
    flex: 1 0 30%;  
    margin: 10px;  
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);  
}
```

Luego, cada imagen ocupará el 100% de su contenedor padre, o sea de figure, pero como figure es descendiente del artículo, ocupará el 100% de ese 30%, es decir todo el recuadro. También le puse una altura específica para que se viese más bonito. Con object-fit cover me aseguro de que la imagen se ajuste a todo el ancho.

```
figure img{  
    width: 100%;  
    height: 600px;  
    object-fit: cover;  
}
```

Para que los elementos del figcaption se encontraran centrados, usé text-align center

```
figcaption{  
    text-align: center;  
}
```

Dado que el diseño debía ser responsive, añadí un media queries:

```
@media (max-width: 1000px){  
    #colecciones_generales{  
        flex-direction: column;  
    }  
  
    aside{  
        width: 100%;  
    }  
  
    aside nav{  
        margin-top: 20px;  
    }  
  
    #colecciones_generales #obras{  
        width: 100%;  
    }  
  
    #colecciones_generales #obras, #todas{  
        flex-direction: column;  
    }  
  
    #colecciones_generales section article{  
        flex: 1 0 auto;  
        width: 100%;  
    }  
  
    #siguientes a{  
        margin-right: 0;  
    }  
}
```

Con esto, cuando la pantalla sea inferior a 1000px, conseguiré que todas las obras se vean de una en una, una debajo de la otra, pues en pantallas más pequeñas no caben 3 fotos a la vez.

Además, el tamaño base de la imagen será automático dependiendo de la pantalla, pero las otras 2 propiedades de flex se mantienen.

Y esto sería el grosor importante de este CSS, el resto de líneas van enfocadas al diseño y a que se vea más bonito, pero no de situación de elementos.

Finalmente, el diseño se vería de la siguiente manera:



2.6 Página de obra, obra1.html

Tal como he mencionado anteriormente, al hacer click en cualquier obra de las colecciones, se redirigirá a una página que mostrará la obra en grande, y a continuación debajo su ficha técnica y al lado un menú de navegación a obras relacionadas.

Lo que hablé con usted para esto es que, todas las obran redirijan a la misma obra y así se ahorra tiempo, y en los enlaces de interés añadir 3 obras más para que se note que se redirige al clicar en ellas y que funciona.

2.6.1 Estructura HTML

La estructura es la siguiente:

```
<main>
  <figure>
    
  </figure>

  <section id="contenedor">
    <section id="ficha_tecnica">
      <h2>Ficha técnica</h2>
      <p>Título: <cite>La alegría de la pradera</cite></p>
      <p>Autor: <cite>Eva BE</cite></p>
      <time datetime="2020-04-25">Fecha: <cite>25-04-2020</cite></time>
      <p>Descripción: <cite>Esta obra de arte captura la belleza
        de una pradera en un día soleado. El cielo azul claro, salpicado
        de nubes esponjosas, se extiende sobre un campo verde y vibrante.
        La luz del sol baña la escena, creando un ambiente de tranquilidad y paz.
        Esta pintura fue creada durante la pandemia, sirviendo como un recordatorio
        de la belleza persistente de la naturaleza en tiempos de incertidumbre.</cite>
      </p>
    </section>
    <aside>
      <h2>Otras obras de interés</h2>
      <nav>
        <a href="obra2.html">Corona de corales</a>
        <a href="obra3.html">Melenas vibrante rosácea</a>
        <a href="obra4.html">El pío de la mañana</a>
      </nav>
    </aside>
  </section>
</main>
```

Por un lado, tenemos la imagen que se visualizará en grande, y por otro lado tendremos otra sección que actuará como contenedor para la ficha técnica y el aside que contiene el menú de navegación con las obras de interés. Esto lo he hecho así porque sino no podía poner la imagen, y debajo de esta la ficha técnica y el menú.

2.6.2 Estructura CSS

Para todas las obras, he usado el mismo fichero CSS, obra.css.

Por un lado se nos pide que la imagen ocupe todo el ancho de la página, por lo que le hago un width del 100%:

```
figure img{  
    border: solid 15px #rgb(130, 140, 187);  
    border-style:groove;  
    width: 100%;  
    height: 100%;  
}
```

Por otro lado, para que debajo de la imagen queden alineados la ficha técnica y el menú de obras relacionadas, hago un display flex:

```
#contenedor{  
    display: flex;  
}
```

Dado que el menú debe de ocupar $\frac{1}{3}$ del ancho, le puse un width del 33.33%:

```
aside{  
    text-align: center;  
    width: 33.33%;  
    background-color: #rgb(207, 224, 240, 0.8);  
    padding: 30px;  
}
```

El resto lo ocupará la ficha técnica, que tendrá un ancho del 66.66%:

```
#ficha_tecnica{  
    width: 66.66%;  
    margin-right: 100px;  
    background-color: #rgb(207, 224, 240, 0.8);  
    padding: 30px;  
}
```

El resto del código CSS es estilo propio.

Las distintas páginas de obras quedarán así:

Ficha técnica

Título: La alegría de la pradera

Autor: Eva BE

Fecha: 25-04-2020

Descripción: Esta obra de arte captura la belleza de una pradera en un día soleado. El cielo azul claro, salpicado de nubes esponjosas, se extiende sobre un campo verde y vibrante. La luz del sol baña la escena, creando un ambiente de tranquilidad y paz. Esta pintura fue creada durante la pandemia, sirviendo como un recordatorio de la belleza persistente de la naturaleza en tiempos de incertidumbre.

Otras obras de interés

[Corona de corales](#)

[Melena vibrante rosácea](#)

[El pío de la mañana](#)

2.7 Página de visita, visita.html

En esta página, podremos encontrar un mapa del museo, así como las distintas salas que este tiene para visitar.

2.7.1 Estructura HTML

La estructura es la siguiente:

```
<main>
  <figure>
    <h2>Plano del museo</h2>
    
  </figure>

  <section class="sala">
    <h2>Sala de pinturas modernas</h2>
    <p>Esta sala alberga una amplia colección de pinturas modernas de artistas reconocidos a nivel mundial. Aquí podrás apreciar la evolución del arte moderno a través de las décadas.</p>
  </section>

  <section class="sala">
    <h2>Sala de pinturas abstractas</h2>
    <p>La sala de pinturas abstractas presenta obras que se alejan de la representación literal de la realidad. Prepárate para sumergirte en un mundo de formas, colores y líneas que desafían la percepción.</p>
  </section>

  <section class="sala">
    <h2>Sala de esculturas</h2>
    <p>En nuestra sala de esculturas, encontrarás impresionantes obras tridimensionales de diversos materiales.</p>
  </section>

  <section class="sala">
    <h2>Sala de dibujo libre</h2>
    <p>La sala de dibujo libre es un espacio interactivo donde los visitantes pueden expresar su creatividad. Ofrecemos materiales de dibujo para que puedas crear tu propia obra de arte.</p>
  </section>
</main>
```

De esta manera, dentro del main tendremos una imagen del plano del museo, y las distintas salas de este separadas en secciones, que contendrán el nombre de la sala y una breve descripción de esta. Todas las secciones pertenecerán a una clase llamada “sala” para que así el CSS sea más legible.

2.7.2 Estructura CSS

Para el diseño, he querido que el main ocupe solo un 80%, ya que pensaba que quedaría más estilizado, y con el margin auto, me lo centrará en el centro:

```
main{
  width: 80%;
  margin: auto;
  padding: 20px;
}
```

Por otro lado, dentro del main he querido centrar el plano del museo, por ello usé text-align center

```
figure{  
    text-align: center;  
}
```

El resto de código CSS va relacionado con el diseño de las diferentes secciones de las salas, dándole un toque más personal y a mi gusto.

El diseño quedaría así finalmente:



2.8 Página de Exposiciones, exposiciones.html

En esta página, se mostrarán las diferentes exposiciones temporales del museo, una imagen sobre ellas y una breve descripción de estas, así como otros datos. También se incluirá la fecha que durará la exposición.

2.8.1 Estructura HTML

La estructura de esta página será la siguiente:

```
<main>
  <section id="exposiciones">
    <h1>Exposiciones temporales</h1>

    <article>
      <figure>
        
        <figcaption>
          <p>Autora: <cite>Julia RA</cite></p>
          <p>Temática: <cite> Arte miniatura</cite></p>
          <time datetime="2024-04-17">Fecha: <cite>17 Abril 2024 - 15 Mayo 2024</cite></time>
        </figcaption>
      </figure>
      <section class="informacion">
        <h2>Exposición de figuras miniatura</h2>
        <p>Si eres fan de los objetos miniatura, esta es tu exposición.
          En ella podrá apreciar las maravillosas obras de arte
          en miniatura creadas por la gran artista Julia RA. Además, podrás
          llevarte obsequios personalizados.
        </p>
      </section>
    </article>
```

Tendremos una sección que englobe a todas las exposiciones, las cuales irán separadas en artículos. A su vez, estos artículos se separarán en un figure que contendrá la imagen de la exposición y el figcaption con los datos relativos al autor, temática y fecha de la exposición; y en un section que pertenecerá a la clase “informacion”, el cual contendrá el nombre de la exposición y una breve descripción de esta.

Esto lo he realizado así para poner el figure al lado de la sección de información, de manera que quede la imagen de la exposición al lado de su título y descripción

2.8.2 Estructura CSS

En primer lugar, he establecido el ancho del main a un 80% para que sea más bonito y las exposiciones no ocupen todo el ancho.

```
main{
  background-color: #rgb(207, 224, 240, 0.8);
  border-radius: 20px;
  margin: 10px auto;
  width: 80%;
}
```

A continuación, he hecho un display flex de toda la sección de exposiciones, y he hecho flex-direction column para que se sitúen todos los elementos de esa sección uno debajo del otro, es decir, primero el título, y debajo de este el resto de artículos. Además he centrado todo con align-items center.

```
#exposiciones {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
}
```

Para que en cada artículo se vea a un lado la imagen con sus respectivos datos y al lado el título de la exposición con una descripción de esta, he hecho un display flex de los artículos y un justify-content between para que ocupen el mismo espacio los elementos del artículo.

El resto de código CSS es de estilo propio y personalizado que no es muy relevante de explicar.

El diseño de esta página quedaría así:



2.9 Página de información, informacion.html

En esta página se nos pedía que incluyéramos información acerca del museo, en concreto de su horario, calendario de apertura, precios y ubicación.

2.9.1 Estructura HTML

La estructura de la página es la siguiente:

He dividido la información que se nos pide en varias secciones, las 2 primeras son el horario de apertura, y el calendario de apertura.

```
<main>
  <section class="apartado">
    <h2>Horario de apertura</h2>
    <p>Lunes a Viernes de 9:00h-14:30h y 16:00h-20:00h</p>
    <p>Sábados y Domingos de 9:30h-15:30h y 16:00h-23:00h</p>
  </section>

  <section class="apartado">
    <table>
      <thead>
        <tr>
          |   <th colspan="25">Aperturas parciales o completas y cierres</th>
        </tr>
      </thead>

      <tbody>
        <tr>
          <td rowspan="7"></td>
          <td colspan="7">Abril</td>
          <td rowspan="7"></td>
          <td colspan="7">Mayo</td>
          <td rowspan="7"></td>
          <td colspan="7">Junio</td>
          <td rowspan="7"></td>
        </tr>
      </tbody>
    </table>
  </section>
```

En la sección del horario de apertura, tan solo he puesto el título y 2 párrafos con el elemento p para indicar el horario entre semana en el primero y en el segundo el de los fines de semana.

En la sección del calendario de apertura he hecho una tabla que contiene los 3 meses de Abril, Mayo y Junio. Para hacer esta tabla, he hecho filas, y en la primera de ellas he indicado que habrán 6 elementos: 3 vacíos y 3 con los meses. Los vacíos tan solo se extenderán 7 líneas debido a que así se van generando huecos entre los distintos calendarios de los meses, y los otros se extenderán 7 columnas, puesto que todos los meses tienen 7 días a la semana.

Dentro de este calendario, he puesto varias clases en algunos elementos que me serán bastante útiles para generar la leyenda de aperturas o cierres:

```
<td class="completa">25</td>
<td class="completa">26</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td class="media">22</td>
<td class="media">23</td>
</tr>

<tr>
  <td class="cierre">29</td>
  <td class="cierre">30</td>
  <td></td>
```

La clase completa indicará que esos días se abrirá el museo todo el día, la clase media que solo se abrirá por la mañana, y la clase cierre que cierra ese día.

Dentro de la sección de la tabla, habrá otra sección llamada leyenda, justo debajo de la tabla de aperturas:

```
<section id="leyenda">
  <h3>Leyenda</h3>
  <section>
    <p class="nombre">Cierre completo</p>
    <p id="rojo"></p>
  </section>

  <section>
    <p class="nombre">Apertura de mañanas</p>
    <p id="naranja"></p>
  </section>

  <section>
    <p class="nombre">Apertura todo el día</p>
    <p id="verde"></p>
  </section>
</section>
```

Esta sección incluirá la leyenda de colores en cuanto a los días del calendario. Se dividirá a su vez en 3 secciones, una para cierre completo, otra para apertura de mañanas y otra para apertura de todo el día. En estas secciones encontraremos un párrafo con cada uno de los elementos de las leyendas, junto con otro párrafo que nos representa el color con un id, lo que me será muy útil para dibujar el color de los elementos de las leyendas en el CSS.

Las otras 2 secciones serán una tabla para los distintos precios y otra para la dirección del museo, la cual la he etiquetado con address ya que hay que aplicar la semántica adecuada.

```
<section class="apartado">
  <h3>Precios</h3>
  <table>
    <thead>
      <tr>
        <th></th>
        <th>Menores</th>
        <th>Estudiantes</th>
        <th>Adultos</th>
      </tr>
    </thead>

    <tbody>
      <tr>
        <td>Lunes-Viernes</td>
        <td>5€</td>
        <td>10€</td>
        <td>25€</td>
      </tr>

      <tr>
        <td>Fines de semana</td>
        <td>7€</td>
        <td>14€</td>
        <td>30€</td>
      </tr>
    </tbody>
  </table>
</section>

<section class="apartado">
  <h2>¿Dónde nos ubicamos?</h2>
  <address>C. de Ruiz de Alarcón, 23, Retiro, 28014 Madrid</address>
</section>
</main>
```

2.9.2 Estructura CSS

He adecuado el width del main de manera que solo ocupe el 80% de la página, y además lo he centrado de los lados con auto:

```
main {  
    width: 80%;  
    margin: 10px auto;  
}
```

Por otro lado, para poner los párrafos del horario de apertura uno debajo del otro, he usado display: block

```
.apartado p{  
    display: block;  
}
```

En cuanto al calendario, he puesto que tenga un width del 100%, o sea que ocupará todo el ancho de ese 80%:

```
table{  
    width: 100%;  
}
```

Para colorear los distintos elementos del calendario dependiendo de la clase a la que pertenezcan, he usado background color, siendo el rojo para el cierre en el día, el verde para apertura completa y el naranja para apertura solo de mañanas:

```
.completa{  
    background-color: #rgb(94, 150, 94);  
}  
  
.cierre{  
    background-color: #rgb(187, 80, 80);  
}  
  
.media{  
    background-color: #rgb(187, 127, 17);  
}
```

Para que los elementos de la leyenda se encuentren unos debajo de los otros, he hecho un display flex y luego un align items center para que se centren debajo de la tabla:

```
#leyenda section{
    display: flex;
    align-items: center;
}
```

Otro código relevante de mi CSS es el cómo dibujo los cuadraditos de la leyenda, y es con los siguientes ids, coloreando el borde y el background del mismo color

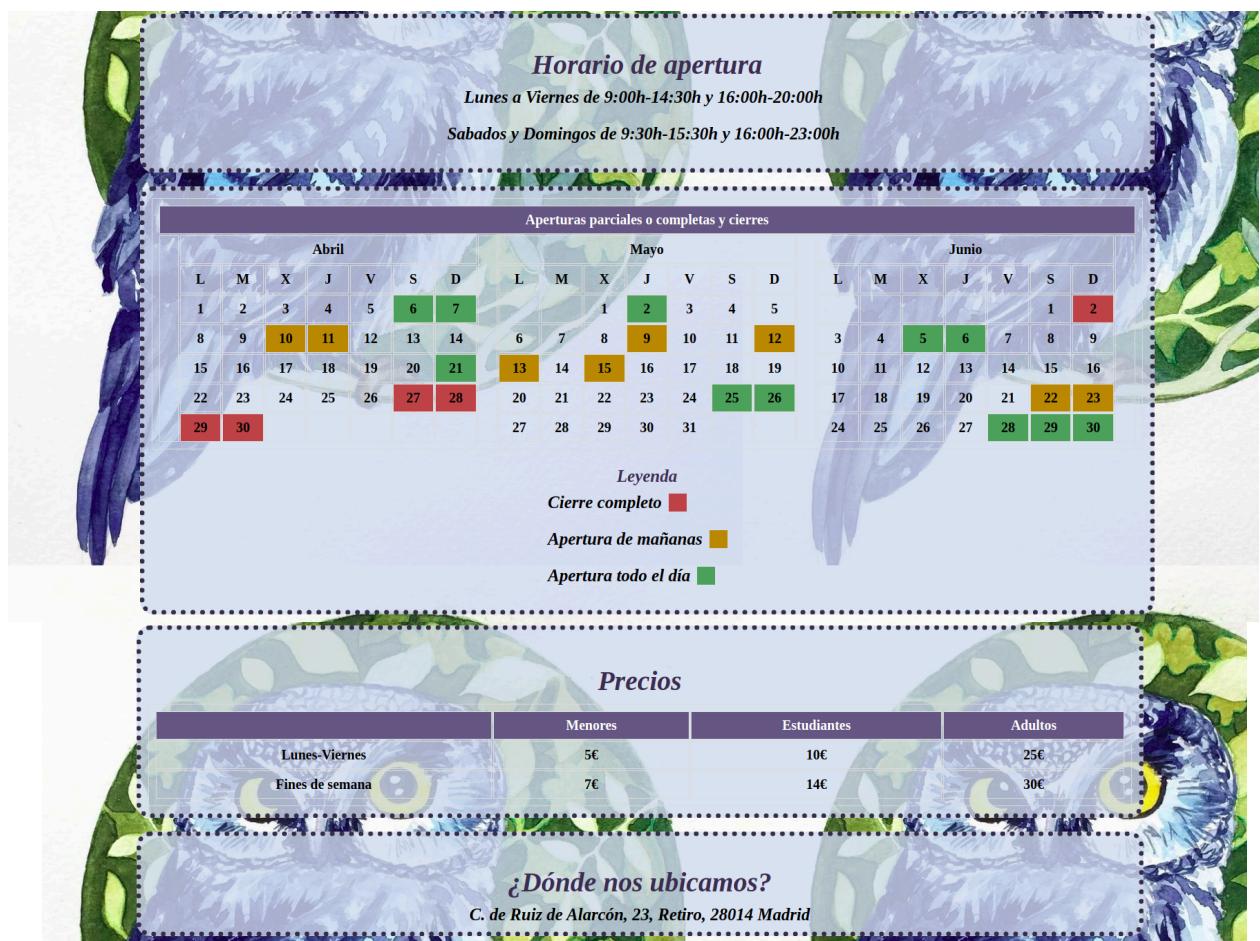
```
#rojo {
    border-color: #rgb(187, 80, 80);
    background-color: #rgb(187, 80, 80);
}

#naranja {
    border-color: #rgb(187, 127, 17);
    background-color: #rgb(187, 127, 17);
}

#verde {
    border-color: #rgb(94, 150, 94);
    background-color: #rgb(94, 150, 94);
}
```

El resto de código es estilo de los elementos.

El diseño de esta página quedaría así finalmente:



2.10 Página de experiencias, experiencias.html

En esta página, albergarán tanto un formulario para añadir tu opinión con respecto a tu experiencia en el museo, como un apartado de las experiencias del resto de usuarios que son miembros y lo han visitado.

2.10.1 Estructura HTML

La estructura de esta página es la siguiente:

Por un lado, tenemos el formulario de experiencias de usuario:

```
<main>
  <section id="tu_opinion">
    <h2>¿Quieres contar tu experiencia en el museo?</h2>

    <p>
      En este formulario podrás contar tu experiencia, además de
      valorar el museo con diversas preguntas a las que tendrás que responder.
      La puntuación de la experiencia en el museo y su comentario, serán visibles
      por el resto de usuarios
    </p>

    <form action="" method="post">
      <label id="recomendar" for="experiencia">¿Qué tan satisfactoria te ha parecido la
      | experiencia del 1 al 10?
      </label>
      <input type="number" id="experiencia" min="1" max="10" required>

      <fieldset id="radio">
        <legend>¿Recomendarías este museo a conocidos o familiares?</legend>
        <label for="recomendacion1">Sí, he disfrutado mucho la experiencia.</label>
        <input type="radio" id="recomendacion1" name="rec" value="Sí">

        <label for="recomendacion2">No, no ha sido muy de mi agrado.</label>
        <input type="radio" id="recomendacion2" name="rec" value="No">
      </fieldset>

      <fieldset id="checkbox">
        <legend>Por favor, selecciona cuál o cuales han sido tus exposiciones temporales favoritas</legend>
        <label for="miniatura">Exposición de figuras miniatura</label>
        <input type="checkbox" id="miniatura" value="miniatura">

        <label for="lego">Exposición de LEGO</label>
        <input type="checkbox" id="lego" value="lego">

        <label for="coolman">Exposición COOLMAN COFFEEMAN</label>
        <input type="checkbox" id="coolman" value="coolman">
      </fieldset>

      <label id="opinion" for="valoracion">Danos tu opinión más detallada, así como mejoras o sugerencias</label>
      <textarea id="valoracion" rows="10" cols="50" required></textarea>

      <input type="submit" value="Enviar">
    </form>
  </section>
```

Esta sección con id tu_opinion contiene el título del formulario y a continuación un párrafo acerca de los datos que se mostrarán al público acerca de la valoración.

A continuación, tenemos el formulario con otros tipos de labels que aún no se habían cubierto, como por ejemplo radio o checkbox, o incluso textarea.

Para radio y checkbox he establecido un fieldset para que se recojan todas las opciones oportunas en él, así como el título de ese fieldset con legend.

Para que el usuario pueda enviar su opinión más detallada, he establecido un textarea con 10 filas y 50 columnas, de manera que sea lo suficientemente extenso como para enviar una opinión.

Por otro lado, tenemos la sección de opiniones, que contendrá las distintas opiniones de los clientes separadas por artículos, mostrando tanto la puntuación del museo como su comentario enviado. Esta parte es totalmente simulada y tan solo sería un ejemplo de cómo quedaría cuando se desarrolle en el lado del servidor.

```
<section id="opiniones">
  <h2>Opiniones de nuestros clientes</h2>

  <article class="opinion_cliente">
    <h3>Dani</h3>
    <p><span class="label">Puntuación:</span> 9</p>
    <p><span class="label">Opinión:</span> Mi experiencia en el museo ha sido maravillosa, sin duda volvería a repetirla.</p>
  </article>

  <article class="opinion_cliente">
    <h3>María</h3>
    <p><span class="label">Puntuación:</span> 8 </p>
    <p><span class="label">Opinión:</span> Ha sido muy agradable venir aquí. El trato es excepcional
       y la verdad que he salido muy contenta.</p>
  </article>

  <article class="opinion_cliente">
    <h3>Gonzalo</h3>
    <p><span class="label">Puntuación:</span> 9</p>
    <p><span class="label">Opinión:</span> Un lugar muy agradable y acogedor al que, sin duda, debes ir siquieres
       disfrutar de un rato agradable y con una gente maravillosa. ¡Qué felicidad!</p>
  </article>
</section>
</main>
```

He decidido hacer esta estructura ya que puedo colocar en la misma línea la sección del formulario y la sección de las opiniones ya mandadas, y dentro de cada una, que los elementos se vean a lo largo, uno debajo del otro.

2.10.2 Estructura CSS

Para el diseño de esta página he hecho lo siguiente:

Por un lado he establecido el width del main a un 80%, así se ve más centrado, y también las 2 secciones del main que estén una al lado de la otra

```
main{
  width: 80%;
  margin: 20px auto;
  display: flex;
  justify-content: space-between;
}
```

Luego, para que los elementos del formulario se encuentren uno debajo del otro, he usado display flex y flex-direction column

```
form{  
  display: flex;  
  flex-direction: column;  
}
```

El resto de código CSS va relacionado con el estilo, ya sea el padding de los elementos, alineación, color etc.

El diseño de esta página quedaría de la siguiente manera:

¿Quieres contar tu experiencia en el museo?
En este formulario podrás contar tu experiencia, además de valorar el museo con diversas preguntas a las que tendrás que responder. La puntuación de la experiencia en el museo y su comentario, serán visibles por el resto de usuarios

¿Qué tan satisfactoria te ha parecido la experiencia del 1 al 10?

¿Recomendarías este museo a conocidos o familiares?
Sí, he disfrutado mucho la experiencia.
No, no ha sido muy de mi agrado.

Por favor, selecciona cuál o cuales han sido tus exposiciones temporales favoritas
Exposición de figuras miniatura
Exposición de LEGO
Exposición COOLMAN COFFEEMAN

Danos tu opinión más detallada, así como mejoras o sugerencias

Enviar

Opiniones de nuestros clientes

Dani
Puntuación: 9
Opinión: Mi experiencia en el museo ha sido maravillosa, sin duda volvería a repetirla.

María
Puntuación: 8
Opinión: Ha sido muy agradable venir aquí. El trato es excepcional y la verdad que he salido muy contenta.

Gonzalo
Puntuación: 9
Opinión: Un lugar muy agradable y acogedor al que, sin duda, debes ir siquieres disfrutar de un rato agradable y con una gente maravillosa. ¡Qué felicidad!

2.11 Página de contacto, contacto.html

En esta página se mostrarán los datos de, en este caso, la desarrolladora de la aplicación web, es decir yo.

2.11.1 Estructura HTML

La estructura de esta página es la siguiente:

```
<main>
  <section class="contacto">
    <h2>Contacto</h2>
    <p><span class="label">Desarrolladora de la página web:</span> Rocio Barragán Moreno</p>
    <p><span class="label">Email:</span>rociobarragan@correo.ugr.es</p>
    <p><span class="label">Ciudad:</span>Granada</p>
    <p><span class="label">Institución:</span>Universidad de Granada</p>
    <p><span class="label">Asignatura:</span>Programación Web</p>
  </section>
</main>
```

Simplemente hay una sección de contacto que incluirá un título, y varios párrafos indicando distintos datos de la desarrolladora. He añadido elementos span con una clase para poder poner los títulos de los datos de un color y la información restante de otro.

2.11.2 Estructura CSS

Para estilizar el diseño de esta página, he puesto que el ancho de la sección ocupe tan sólo un 50% y que se alinee al centro. Además he aplicado text-align center para que los elementos de dentro de la sección queden centrados en la sección.

```
.contacto {
  width: 50%;
  text-align: center;
  margin: 10px auto;
  padding: 20px;
  background-color: #rgb(207, 224, 240, 0.8);
  border-radius: 20px;
}
```

El resto de código es en cuanto al estilo de los elementos.

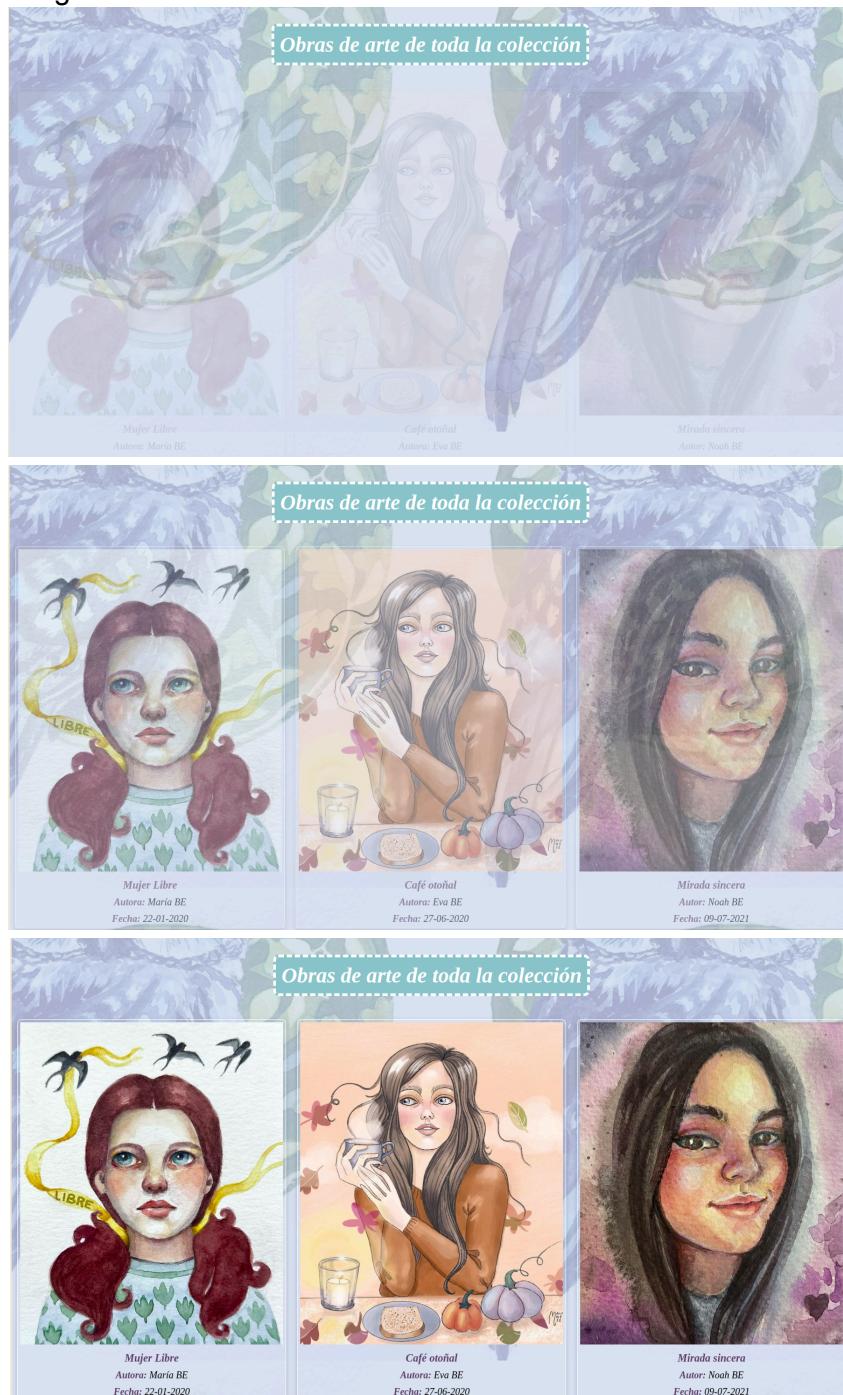
3. Aspectos innovadores

En este apartado describiré varios aspectos innovadores que he incluido en la práctica para darle un toque de originalidad.

3.1 Transiciones

He usado las transiciones de elementos tanto en las colecciones como en el alta exitosa de usuarios.

La transición de las colecciones consiste en que al abrir cualquier página de colecciones, todas las obras aparezcan poco a poco, desde ser invisibles hasta ser completamente vistas. Sería algo así la transición:



Para esto, he usado el siguiente código CSS:

```
@keyframes fadeIn {
    0% {opacity: 0;}
    100% {opacity: 1;}
}

#todas {
    opacity: 0;
    animation: fadeIn 1s ease-in forwards;
}
```

Con keyframes definimos la propiedad, en la que opacidad 0 representa la transparencia y 1 la total visibilidad del elemento,

Esto se aplica a todas las obras, o sea a la sección todas, y la aplica con duración de 1 segundo, ya que sino puede resultar bastante agobiante o cansado para el usuario. Con ease-in indica que la animación comienza lentamente y luego va más rápido, aunque con 1 segundo no se nota apenas, y forwards indicará que se quedará con la última forma de la transición, o sea, con la imagen completamente visible.

Después, en alta exitosa de usuarios, cuando este se ha registrado correctamente, he añadido otra transición:



El código utilizado es el mismo que el anterior.

3.2 Cambios de colores progresivos

Para el título del museo, he aplicado que vaya cambiando de color progresivamente, pero variando entre los tonos oportunos acorde con la paleta de colores que he usado, azul, celeste y violeta.

Sería algo así:



Estos cambios los he hecho con el siguiente código:

```
@keyframes colorChange {
    0% { color: #rgb(60, 110, 110); }
    50% { color: #rgb(139, 120, 139); }
    100% { color: #rgb(69, 168, 168); }
}

#logo h1 {
    font-family: 'Indie Flower', cursive;
    font-size: 80px;
    font-weight: bold;
    color: #rgb(60, 110, 110);
    -webkit-text-stroke-width: 2px;
    -webkit-text-stroke-color: #white;
    animation: colorChange 5s infinite;
}
```

Se puede apreciar que va cambiando de color progresivamente, y se aplica al h1. Al inicio de la animación (0%) será azul, a la mitad (50%) será violeta y al final celeste (100%).

3.3 Borde blanco en título

Otra innovación que he añadido, que no es tan original como las anteriores, es que el borde de las letras del título del museo se vea en blanco. Lo hice sobretodo para que, cuando solape con la imagen de fondo, se viesen más claras las letras



Esto lo he hecho con la regla -webkit-text-stroke, aplicando width de 2px y color blanco:

```
#logo h1 {  
    font-family: 'Indie Flower', cursive;  
    font-size: 80px;  
    font-weight: bold;  
    color: #rgb(60, 110, 110);  
    -webkit-text-stroke-width: 2px;  
    -webkit-text-stroke-color: white;  
    animation: colorChange 5s infinite;  
}
```