

1. Introducción

El proyecto desarrollado consiste en una plataforma web llamada **SporyHigh**, centrada en la difusión de contenido musical y noticias relacionadas.

El sitio incluye varias páginas: **Inicio**, **Playlists**, **Artistas**, **Noticias** y **Contacto**, todas ellas diseñadas con HTML5 y CSS3.

El objetivo principal ha sido aplicar de manera práctica los conceptos aprendidos durante la unidad, tales como:

- Uso de etiquetas semánticas
- Maquetación con Flexbox y Grid
- CSS responsive mediante media queries
- Uso de variables CSS
- Formulario con validación
- Menú hamburguesa
- Accesibilidad básica

El proyecto combina estructura semántica, diseño responsive y estilos modernos para crear una experiencia clara y fácil de usar.

2. Conceptos aplicados del AV

2.1 Uso de etiquetas semánticas HTML5

En el proyecto se han utilizado las etiquetas semánticas:

- `<header>`
- `<nav>`
- `<main>`
- `<section>`
- `<article>` (usado en la página de inicio)
- `<footer>`
- `<figure>` y `<figcaption>` (usado en el hero del index)

Ejemplo utilizado en proyecto

```
<!-- CABECERA -->
<header class="site-header">
  <div class="header-inner">
    <a href="index.html" class="logo" aria-label="Volver al inicio">
      <span class="logo-main">SPORY</span><span class="logo-highlight">HIGH</span>
    </a>

    <!-- BOTÓN HAMBURGUESA -->
    <button class="nav-toggle" aria-label="Abrir menú">
      <span></span>
      <span></span>
      <span></span>
    </button>

    <nav class="main-nav" aria-label="Navegación principal">
      <ul>
        <li><a href="index.html" class="active">Inicio</a></li>
        <li><a href="playlists.html">Playlists</a></li>
        <li><a href="artistas.html">Artistas</a></li>
        <li><a href="noticias.html">Noticias</a></li>
        <li><a href="contacto.html">Contacto</a></li>
      </ul>
    </nav>
  </div>
</header>
```

2.2 Maquetación con Flexbox

Utilizando en múltiples partes:

Cabecera:

```
.header-inner {
  max-width: var(--max-width);
  margin: 0 auto;
  padding: 0.8rem 1.5rem;
  display: flex;
  align-items: center;
  justify-content: space-between;
}
```

Tarjetas de destacados:

```
.featured-content {
  display: flex;
  gap: 1rem;
  align-items: center;
}
```

Hero principal:

```
.hero-card {
  background: var(--color-negro);
  color: var(--color-blanco);
  border-radius: 1rem;
  padding: 1.8rem 2rem;
  display: flex;
  gap: 1.5rem;
  align-items: center;
  justify-content: space-between;
  box-shadow: var(--sombra-suave);
  max-width: 950px;
}
```

Todo esto visible en el archivo styles.css

2.3 Maquetación con CSS Grid

Utilizado en:

- Playlists
- Artistas
- Noticias
- Destacados del index

Ejemplo:

```
/*  
  ARTISTAS  
*/  
  
.artists-full-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(230px, 1fr));  
  gap: 1.5rem;  
  align-items: stretch;  
}
```

Otro ejemplo:

```
.playlist-full-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));  
  gap: 1.5rem;  
}
```

2.4 Uso de Variables CSS

Variables declaradas en :root al principio del CSS (styles.css)

```
:root {  
  --color-verde: #70F97A;  
  --color-negro: #000000;  
  --color-blanco: #ffffff;  
  --color-gris: #d9d9d9;  
  
  --max-width: 1100px;  
  --radio-borde: 1rem;  
  --sombra-suave: 0 12px 35px rgba(0, 0, 0, 0.15);  
  --transicion: 0.25s ease;  
}
```

2.5 Diseño responsive con Media Queries

Uso tres breakpoints:

```
@media (max-width: 750px) {  
  .news-card-large {  
    flex-direction: column;  
  }  
  .news-card-large img {  
    width: 100%;  
    height: auto;  
  }  
}
```

```
@media (max-width: 900px) {  
  .hero-card {  
    flex-direction: column;  
    text-align: center;  
  }  
}  
  
@media (max-width: 820px) {  
  .nav-toggle {  
    display: flex;  
  }  
}
```

2.6 Menú hamburguesa (responsive)

```
.nav-toggle {  
  display: none;  
  flex-direction: column;  
  gap: 5px;  
  width: 35px;  
  height: 30px;  
  background: transparent;  
  border: none;  
  cursor: pointer;  
}
```

Con el CSS de arriba a la derecha, el segundo y código en JavaScript

```
<script>  
  document.querySelector(".nav-toggle").addEventListener("click", () => {  
    document.querySelector(".main-nav").classList.toggle("nav-open");  
    document.querySelector(".nav-toggle").classList.toggle("open");  
  });  
</script>
```

2.7 Formulario con validación HTML5

```
<div class="contact-card">
  <form action="#" method="POST" class="contact-form">

    <label for="nombre">Nombre completo</label>
    <input type="text" id="nombre" name="nombre" required placeholder="Tu nombre..">

    <label for="email">Correo electrónico</label>
    <input type="email" id="email" name="email" required placeholder="tugmail@ejemplo.com">

    <label for="asunto">Asunto</label>
    <input type="text" id="asunto" name="asunto" required placeholder="Motivo principal del mensaje">

    <label for="mensaje">Mensaje</label>
    <textarea id="mensaje" name="mensaje" rows="6" required placeholder="Escribe tu mensaje aqui..."></textarea>

    <button type="submit" class="btn btn-primary">Enviar mensaje</button>

  </form>
</div>
```

2.8 Sombras, border redondeados y estética

```
.artist-card-large {
  background: var(--color-blanco);
  border-radius: var(--radio-borde);
  padding: 1.8rem 1.5rem 2rem;
  text-align: center;
  box-shadow: var(--sombra-suave);

  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: flex-start;
  height: 100%;
}
```

2.9 Botones personalizados y hover

```
.btn {
  display: inline-block;
  border: none;
  border-radius: 999px;
  padding: 0.7rem 1.6rem;
  font-weight: 700;
  cursor: pointer;
  transition: var(--transicion);
}

.btn-primary,
.btn-secondary {
  background: var(--color-verde);
  color: var(--color-negro);
  box-shadow: var(--sombra-suave);
}

.btn:hover {
  transform: translateY(-1px);
}
```

2.10 Optimización tipográfica

```
.main-nav a {  
  font-size: 0.95rem;  
  text-transform: uppercase;  
  letter-spacing: 0.07em;  
  padding-bottom: 0.15rem;  
  border-bottom: 2px solid transparent;  
  transition: var(--transicion);  
}
```

2.11 Imágenes responsive

```
img {  
  width: 100%;  
  display: block;  
  object-fit: cover;  
}
```

2.12 Accesibilidad básica

aria-label en botones hamburguesa

aria-label en nav

alt en todas las imágenes

Semántica adecuada