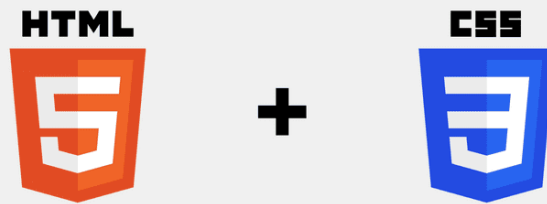


# Guías de estilo HTML y CSS

Joel Acoran Cruz Morales - Curso 2023/2024

---




## Proyecto UFC

Documento para las guías de estilos de HTML y CSS para el proyecto final de 2do curso del Ciclo formativo de grado superior en Diseño de Aplicaciones Web de Joel Acoran Cruz Morales

---

<b>Proyecto UFC</b>	<b>1</b>
<b>HTML:</b>	<b>3</b>
1. Indentación y espaciado	3
2. Nomenclatura de clases e IDs	4
3. Uso de elementos semánticos	5
4. Comentarios	6
<b>CSS:</b>	<b>7</b>
1. Organización del código	7
2. Selección de selectores	8
3. Tamaño y eficiencia	9
4. Comentarios	9



# HTML:

## 1. Indentación y espaciado

- Utiliza sangrías (indentación) consistentes, preferiblemente de 2 o 4 espacios.
- Usa espacios en lugar de tabulaciones para una mejor consistencia en diferentes editores.



```
index.blade.php X
resources > views > index.blade.php > ...
You, 2 minutes ago | 1 author (You)
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <link rel="stylesheet" href="estilos.css">
8   <title>Inicio</title>
9 </head>
10 <body>
11   @include('header')
12   <h2>Bienvenido</h2>
13   
14   @include('footer')
15 </body>
16 </html>
```

*Ejemplo de indentación y espaciado*

## 2. Nomenclatura de clases e IDs

- Utiliza nombres descriptivos y significativos para las clases e IDs.
- Evita nombres genéricos como “div1”, “elemento2”, etc.
- Utiliza nombres en minúsculas y palabras separadas por guiones (kebab-case) para las clases

```
insertarVelada.Blade.php X
resources > views > Velada > insertarVelada.Blade.php > html > body > div#formulario-Velada
You, 2 minutes ago | 1 author (You)
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <link rel="stylesheet" href="{{ asset('estilos.css') }}">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9   <title>Añadir Velada</title>
10  <script src="{{ asset('JavaScript/insertarVelada.js') }}"></script>
11 </head>
12 @include('header')
13 <body>
14 <h2>Añadir Velada</h2>
15
16 <div id="formulario-Velada">
17   <form action="{{ route('velada/insertar') }}" method="POST" class="metodo-post">
18     @csrf
19     <label for="nombre-Velada">Nombre de la Velada:</label>
20     <input type="text" id="nombre-Velada" name="Nombre_Vel"><br><br>
21
22     <label for="fecha-Velada">Fecha de la Velada:</label>
23     <input type="date" id="fecha-Velada" name="Fecha_Vel"><br><br>
24
```

*Ejemplo de nomenclatura de clases e ID*

### 3. Uso de elementos semánticos

- Utiliza elementos semánticos apropiados para mejorar la accesibilidad y la estructura del documento, como <header>, <footer>, <nav>, <article>, <section>, etc.

```
header.blade.php X
resources > views > header.blade.php > ...
You, 1 minute ago | 1 author (You)
1 <header>
2   <h1 class="titulo-principal"><a href="{{ route('index') }}">Proyecto UFC</a></h1>
3   <nav>
4     <a href="{{ route('participantes') }}">Participantes</a>
5     <a href="{{ route('velada') }}">Velada</a>
6     <a href="{{ route('peleas') }}">Peleas</a>
7   </nav>
8 </header>
9
```

#### Utilización de nav para header

```
You, 4 hours ago | 1 author (You)
1 <footer>
2   <div class="container">
3     <p style="color: rgb(255, 255, 255);&copy; {{ date('Y') }} Joel Acoran Cruz Morales - Proyecto de Final de
4     Ciclo de Desarrollo de Aplicaciones Web</p>
5   </div>
6
7   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
8   <script src="JavaScript/Login.js"></script>
9   <script>
10    $(document).ready(function() {
11      const auth_token = sessionStorage.getItem('auth_token');
12      if (auth_token) {
13        const userToken = JSON.parse(auth_token);
14        check_Token(userToken.ID_Usuario, userToken.token, null, '/login');
15      } else {
16        alert("Usuario no autenticado, redireccionando a /login...");
17        window.location.href = '/login';
18      }
19    });
20  </script>
21 </footer>
22
```

#### Utilización de footer para comprobaciones de token

## 4. Comentarios

- Utiliza comentarios claros y concisos para explicar secciones importantes del código, como contenedores de secciones, secciones de estilos específicos, etc

```
25
26 <footer>
27   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
28
29   <!-- Ruta del Script para realizar el login -->
30   <script src="JavaScript/Login.js"></script>
31   <!-- Script para verificar el token de autenticación -->
32   <script>
33     $(document).ready(function() {
34       // Verificar si existe un token de autenticación en la sesión
35       const auth_token = sessionStorage.getItem('auth_token');
36       if (auth_token) {
37         // Obtener y verificar el token de usuario
38         const userToken = JSON.parse(auth_token);
39         check_Token(userToken.ID_Usuario, userToken.token, '/', null);
40       }
41     });
42   </script>
43
44 </footer>
45
46 </html>
```

*Ejemplos de utilización de comentarios para explicar secciones diferentes*

# CSS:

## 1. Organización del código

- Agrupa propiedades relacionadas (por ejemplo, diseño, tipografía, colores) juntas
- Mantén un orden lógico en tus reglas CSS (por ejemplo, posición, caja, tipografía, etc.)

```
# estilos.css ×
public > # estilos.css > header
You, 17 minutes ago | 1 author (You)
1  /* Estilos para el cuerpo */
2  body {
3      margin: 0;
4      padding: 0;
5      background-image: url('img/Wallpaper.jpg');
6      background-size: cover;
7      background-position: center 1%;
8      font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
9      text-align: center;
10 }
11
12 /* Estilos para el encabezado */
13 header {
14     display: flex;
15     align-items: center;
16     justify-content: center;
17     height: 12vh;
18     background-color: rgba(0, 38, 255, 0.1);
19     backdrop-filter: blur(10px);
20 }
```

*Ejemplos de agrupación de propiedades.*

## 2. Selección de selectores

- Prefiere selectores específicos sobre selectores generales para evitar afectar elementos no deseados
- Evita selectores de ID en CSS cuando sea posible, ya que limitan la reutilización y aumentan la especificidad.

```
31  /* Estilos para el contenedor */
32  .container {
33      max-width: 800px;
34      margin: 0 auto;
35      padding: 20px;
36  }
37
38  /* Estilos para los títulos */
39  h1,h2,h3 {
40      color: #ffffff;
41  }
42
43  /* Estilos para el formulario */
44  #formulario-Pelea, #formulario-Velada{
45      color: #ffffff;
46  }
47
48  /* Estilos para los resultados */
49  #resultados {
50      display: none;
51      margin-top: 20px;
52  }
```

*Ejemplos de selectores específicos.*



### 3. Tamaño y eficiencia

- Minimiza la repetición de código utilizando clases reutilizables y heredando estilos.
- Evita el uso excesivo de selectores anidados, ya que puede complicar el mantenimiento del código.

```
21
22 .titulo-principal {
23     margin-right: auto;
24 }
25
```

*Ejemplos de tamaño.*

### 4. Comentarios

- Incluye comentarios para explicar secciones importantes del CSS, especialmente aquellas que pueden no ser obvias a primera vista

```
108 footer::before {
109     content: '';
110     position: absolute;
111     top: 0;
112     left: 0;
113     width: 100%; /* Ancho completo */
114     height: 100%; /* Altura completa */
115     background-color: inherit; /* Color de fondo heredado */
116     filter: blur(10px); /* Desenfoque */
117     z-index: -5; /* Colocado detrás */
118 }
119
120
121
```

*Ejemplos de comentarios para explicar funciones importantes del css.*