

Documentación del proyecto Shopi

ALBERT ALARCÓN MARTÍNEZ



Uso de Tailwind CSS en el Proyecto



Configuración en `tailwind.config.cjs`

El archivo de configuración de Tailwind `tailwind.config.cjs` define las reglas básicas del framework en el proyecto:



Modo Oscuro

```
darkMode: 'class',
```

- **Modo oscuro activado manualmente** mediante la clase `dark`.
- Permite alternar entre `light` y `dark` aplicando `document.documentElement.classList.add('dark')`.



Rutas de los archivos CSS

```
content: [  
  "./index.html",  
  "./src/**/*.{js,ts,jsx,tsx}",  
],
```

- **Tailwind escanea los archivos** dentro de `src/` y `index.html` para optimizar los estilos generados.



Personalización del tema

Fuente global personalizada

```
fontFamily: {  
  sans: ['Poppins', 'sans-serif'],  
},
```

- **Fuente primaria:** `Poppins`, garantizando un diseño limpio y moderno.

Colores personalizados

```
colors: {  
  primary: '#3ea987', // Verde principal  
  secondary: '#49caa1', // Verde más claro  
  accent: '#60a5fa', // Azul para detalles  
},
```

- **Colores principales para la interfaz** (`primary`, `secondary`, `accent`).
- **Facilita la coherencia en toda la UI.**

Bordes redondeados personalizados

```
borderRadius: {  
  'x1': '1rem',  
  '2x1': '1.5rem',  
},
```

- Define radios de borde más grandes (`x1` y `2x1`) para un diseño más **moderno y suave**.



Estilos Globales en `Pages/App/App.css`

```
@tailwind base;          /* Estilos base de Tailwind */  
@tailwind components;    /* Componentes de Tailwind */  
@tailwind utilities;     /* Utilidades de Tailwind */
```

- **Organización estándar de Tailwind** para manejar la carga de estilos.



Variables CSS personalizadas

```
:root {  
  --primary-color: #3ea987;  
  --secondary-color: #000000;  
}
```

- Permite **usar los colores personalizados en todo el CSS**.
- **Ejemplo de uso:**

```
background-color: var(--primary-color);
```



Comportamiento del scroll

```
html {  
  scroll-behavior: smooth;  
}
```

- **Habilita el scroll suave** al navegar en la página.

Estilos base del **body**

```
body {  
  font-family: 'Poppins', sans-serif;  
  @apply bg-white dark:bg-gray-900 text-gray-900 dark:text-white transition-  
  colors duration-200;  
}
```

- **Usa Tailwind para:**
 - Establecer el fondo blanco (**bg-white**) en modo claro y **dark:bg-gray-900** en modo oscuro.
 - **Colores del texto:** **text-gray-900** en claro y **dark:text-white** en oscuro.
 - **Transiciones suaves** al cambiar de tema.

Uso de **flex** y **grid** en el Diseño

Sistema de Layout con **flex**

Tailwind se usa ampliamente para **estructurar los componentes con flex**:

```
<div className="flex flex-col min-h-screen">
```

- **flex**: Distribuye los elementos en fila/columna.
- **flex-col**: Organiza los hijos en **columna**.
- **min-h-screen**: Hace que ocupe **toda la pantalla**.

Ejemplo de uso en un contenedor de productos:

```
<div className="flex flex-wrap gap-4">
```

- **flex-wrap**: Permite que los elementos pasen a la siguiente línea si no caben.
- **gap-4**: Añade un espacio uniforme entre los elementos.

Sistema de Layout con **grid**

Para el **listado de productos y otras secciones** se usa **grid**:

```
<div className="grid gap-4 grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 p-4">
```

- **grid**: Activa CSS Grid.
- **gap-4**: Espaciado uniforme entre elementos.
- **grid-cols-1 sm:grid-cols-2 lg:grid-cols-4**:
 - **Móvil**: 1 columna (**grid-cols-1**).
 - **Pantallas medianas (sm)**: 2 columnas (**grid-cols-2**).
 - **Pantallas grandes (lg)**: 4 columnas (**grid-cols-4**).
- **Esto hace el diseño 100% responsive automáticamente.**

🌙 Modo Oscuro con Tailwind

El modo oscuro está habilitado con la configuración:

```
darkMode: 'class',
```

y se usa en los estilos:

```
@apply bg-white dark:bg-gray-900 text-gray-900 dark:text-white;
```

- **Ejemplo aplicado al <body>**:
 - En modo claro → **bg-white text-gray-900**
 - En modo oscuro → **bg-gray-900 text-white**
 - **El cambio es automático según la clase dark en <html>**.

📱 Estilos para Inputs (**range**) en modo oscuro

```
.dark input[type="range"]::-webkit-slider-runable-track {  
  background: #374151;  
}
```

- **Se cambia el color de la barra en modo oscuro** para adaptarse mejor a la UI.

🌀 Estilización de **Swiper.js**

🎮 Botones de navegación personalizados

```
.swiper-button-next,  
.swiper-button-prev {
```

```
    color: white !important;
}
```

- Cambia el color de los botones de navegación (`next` y `prev`) a blanco.

🎯 Paginación personalizada

```
.swiper-pagination-bullet {
  background: white !important;
}

.swiper-pagination-bullet-active {
  background: var(--primary-color) !important;
}
```

- Bullets (• • •) en blanco por defecto.
- Bullet activo usa el color primario (`var(--primary-color)` → `#3ea987`).

📦 Componentes Reutilizables con `@apply`

El uso de `@apply` en Tailwind permite **evitar repetición de estilos**:

```
body {
  @apply bg-white dark:bg-gray-900 text-gray-900 dark:text-white transition-
  colors duration-200;
}
```

Esto evita escribir manualmente:

```
body {
  background-color: white;
  color: #111;
}
body.dark {
  background-color: #1a1a1a;
  color: white;
}
```

- Más limpio y fácil de mantener.

🏆 Conclusión

✓ Principales ventajas del enfoque con Tailwind

- ✓ **Flexibilidad total:** Se usa `flex` y `grid` según cada caso.
- ✓ **Modo oscuro optimizado:** Cambios suaves entre `light` y `dark`.
- ✓ **Configuración clara:** Tema personalizado con colores y fuentes en `tailwind.config.cjs`.
- ✓ **Diseño adaptable:** `grid-cols` y `flex-wrap` garantizan una UI **responsive**.
- ✓ **Carga optimizada:** Solo se generan los estilos utilizados gracias a la configuración de `content`.
- ✓ **Código más limpio:** Uso de `@apply` para evitar repetición de estilos.