# **Documentación de la Arquitectura de Ordenamiento de Código CSS**

## **Enfoque SMACSS**

La arquitectura CSS del proyecto sigue la metodología SMACSS (Scalable and Modular Architecture for CSS), permitiendo un código organizado, escalable y de fácil mantenimiento. Se ha estructurado en diferentes categorías dentro de la carpeta css.

## **Convención de Nombres de Clases**

Estamos utilizando una convención de nombres basada en bloques y modificadores para mejorar la claridad y consistencia del código. Ejemplos:

* **class="menu\_\_container"**
* **class="menu\_\_item-second-text"**

## **Estructura de Archivos:**

**css/**

**│── base/**

**│ ├── reset.css**

**│ ├── typography.css**

**│ ├── variables.css**

**│**

**│── layout/**

**│ ├── home/**

**│ │ ├── header.css**

**│ │ ├── services.css**

**│ ├── form/**

**│ │ ├── form.css**

**│**

**│── modules/**

**│ ├── home/**

**│ │ ├── header-modules/**

**│ │ │ ├── button.css**

**│ │ ├── services-modules/**

**│ │ │ ├── services-description.css**

**│**

**│── state/**

**│ ├── home/**

**│ │ ├── home-states.css**

**│ ├── form/**

**│ │ ├── form-states.css**

**│ ├── menu/**

**│ │ ├── menu-states.css**

**│ ├── product/**

**│ │ ├── product-states.css**

**│**

**│── theme/**

**│ ├── dark-theme.css**

## **Descripción de Carpetas (ejemplos prácticos)**

### **1. base/**

**Contiene los estilos globales del proyecto, incluyendo:**

* reset.css: Restablece los estilos predeterminados del navegador.
* typography.css: Define la tipografía global.
* colors.css: Contiene variables CSS reutilizables como colores.

### **2. layout/**

**Define la estructura de las vistas, organizando los estilos por secciones específicas:**

* layout/home/header.css: Estilos de maquetado para el encabezado de la página de inicio.
* layout/home/services.css: Estilos de maquetado para la sección de servicios en la página de inicio.
* layout/form/form.css: Estilos de maquetado para la página del formulario.

### **3. modules/**

**Define componentes reutilizables dentro de cada vista:**

* modules/home/header-modules/buttons.css: Estilos específicos para los botones dentro del encabezado de la página de inicio.
* modules/home/services-modules/services-description.css: Estilos específicos para la descripción de servicios.

### **4. state/**

**Define los estados de los elementos en cada vista:**

* state/home/home-states.css: Define estados específicos de la página de inicio.
* state/form/form-states.css: Define estados para la página del formulario.
* state/menu/menu-states.css: Estados específicos de la página de menú.

### **5. theme/**

**Define estilos de temas visuales como tipografía y colores.**

* theme/dark-theme.css: Estilos relacionados con la presentación visual o aspecto del proyecto.

## **Beneficios de Esta Estructura**

* Modularidad: Facilita la reutilización de código y la escalabilidad.
* Mantenibilidad: Separa las responsabilidades para un mejor control de estilos.
* Flexibilidad: Permite realizar cambios sin afectar otras partes del proyecto.
* Claridad: Facilita la colaboración en equipo al seguir una estructura bien definida.

**Esta documentación servirá como referencia para la organización del código CSS en el proyecto. Cualquier actualización en la arquitectura deberá seguir estas convenciones.**