

# Proyecto Web Personalizado con Framework CSS

## 1. Framework implementado: Tailwind CSS

¿Qué es?

Tailwind CSS es un framework de utilidades para CSS. En lugar de ofrecer componentes listos como botones o menús, proporciona clases pequeñas y reutilizables (como `flex`, `bg-green-500`, `text-center`) que se combinan directamente en el HTML para dar estilo. Esto permite un control muy fino sobre el diseño y evita escribir CSS adicional.

¿Por qué lo elegí?

Decidí usar Tailwind porque ofrece rapidez al momento de maquetar, ya que con clases predefinidas se puede construir un diseño moderno y responsivo sin necesidad de escribir CSS desde cero. Además, su filosofía de “utility-first” hace que el código sea más consistente y fácil de mantener.

Evidencias del proyecto:

- Navbar translúcida: gracias a las clases `backdrop-blur-lg bg-white/70 fixed w-full z-50`, la barra de navegación queda fija, semitransparente y con efecto de difuminado.
- Hero con overlay: en la sección principal se utilizó `bg-[url('hiking.png')] bg-cover bg-center` con un overlay `bg-black/40` para mejorar el contraste del texto.
- Tarjetas con hover: las tarjetas de “Programas” usan `hover:scale-105 transition`, lo que genera un efecto dinámico cuando el usuario pasa el cursor.

## 2. Framework investigado: Bootstrap 5

¿Qué es?

Bootstrap es uno de los frameworks CSS más populares. A diferencia de Tailwind, ofrece una gran variedad de componentes preconstruidos (navbars, botones, formularios, modales, etc.), además de un sistema de grillas que facilita la creación de diseños responsivos.

¿Qué resuelve?

Bootstrap permite a los desarrolladores crear interfaces completas de manera muy rápida, incluso sin mucho conocimiento previo de CSS, porque muchos elementos vienen listos para usarse y con estilos predeterminados.

## 3. Comparativa

- Facilidad / curva de aprendizaje
  - *Tailwind*: al inicio cuesta memorizar las clases utilitarias, pero una vez entendido se avanza muy rápido.
  - *Bootstrap*: más fácil al comienzo, ya que basta con aplicar componentes prehechos.
- Personalización / theming
  - *Tailwind*: altísima personalización a través del archivo `tailwind.config.js`, donde se pueden definir colores, tipografías y espaciados.
  - *Bootstrap*: personalizable mediante variables Sass, pero su estilo base es más reconocible y puede requerir sobrescribir reglas.
- Tamaño / rendimiento
  - *Tailwind*: cuando se compila en producción elimina las clases no utilizadas, lo que genera un CSS muy pequeño.
  - *Bootstrap*: por CDN se carga todo el bundle, lo que puede ser más pesado, aunque también se puede optimizar con un build personalizado.

