

1. SCSS: más allá del CSS vanilla

Usar **SCSS** fue un cambio refrescante. Después de haber trabajado un tiempo con Tailwind, regresar a CSS vanilla me parecía tedioso por su falta de estructura. SCSS resolvió ese problema: con variables, mixins y su modularidad, diseñar fue mucho más claro y organizado. Es una herramienta potente que seguramente usaré en más proyectos.

2. Adaptación para dispositivos y lógica dinámica

- **PDFs y experiencia por dispositivo:** Implementé una lógica para que los archivos PDF se puedan **visualizar en desktop** y **descargar en móviles o tablets** usando `iframes`. Esto mejora la experiencia del usuario dependiendo de su dispositivo.
- **Un hero más llamativo:** El diseño del **hero** de la página de inicio se ve mucho más impactante. Sin embargo, el video en 2K que usé plantea un reto: no es adecuado para dispositivos menos potentes. Esto se ha solucionado con **JavaScript** para cargar optimizar la carga del video al igual que los demás componentes de la web. Comprimir el formato `.mp4` y reescalar el video a 1080p fue esencial para una carga veloz.
- **Componentes dinámicos:** Aproveché JavaScript para crear un **header y footer que cambian dinámicamente** según la página. Además, añadí lógica para gestionar PDFs, animaciones como el menú toggle en móviles, y centralicé todo en un archivo `main.js` para mantener ordenado el código.

3. Recursos gráficos y colaboración

Colaborar con mi socia, quien creó todas las imágenes, logos y favicons, fue muy práctico. Tener acceso directo a recursos personalizados eliminó cualquier preocupación sobre derechos de autor y me permitió trabajar con confianza. Este flujo de trabajo será clave en futuros proyectos, como la web de **SuRótulo** que planeo desplegar en 2025.

4. Retos técnicos y aprendizajes

- **Colores dinámicos en SVGs:** Al trabajar con archivos SVG en etiquetas `img`, enfrenté el reto de modificar sus colores. Lo resolví con el atributo `filter`, lo que resultó ser práctico y efectivo.
- **La transición a `@use` en SCSS:** La nueva sintaxis `@use` sustituyó a los `imports` tradicionales y al principio fue un desafío, pero el cambio aporta claridad y modularidad. Para llamar variables, tuve que adaptarme a su nuevo formato (`variables.$nombre`), lo cual me pareció un enfoque más organizado.

5. Conclusión

La experiencia con **SCSS me sorprendió mucho más de lo que esperaba**. Es una herramienta que no solo simplifica el diseño, sino que lo hace más eficiente y claro. Este proyecto también me permitió profundizar en JavaScript y explorar nuevas soluciones para diseño responsivo. A pesar de algunos retos, fue una experiencia enriquecedora que me dejó con ganas de seguir utilizando estas herramientas en proyectos futuros.