Buenas prácticas:

* [Código limpio](https://stackedit.io/editor#c%C3%B3digo-limpio)
* [Guí­a de estilo, convenciones y buenas prácticas.](https://stackedit.io/editor#gu%C3%AD%C2%ADa-de-estilo-convenciones-y-buenas-pr%C3%A1cticas)
* [Convenciones](https://stackedit.io/editor#convenciones)
* [Espacios en blanco — Identación](https://stackedit.io/editor#espacios-en-blanco-identaci%C3%B3n)
* [Longitud de una lí­nea](https://stackedit.io/editor#longitud-de-una-l%C3%AD%C2%ADnea)
* [Comillas](https://stackedit.io/editor#comillas)
* [Llaves](https://stackedit.io/editor#llaves)
* [Punto y coma](https://stackedit.io/editor#punto-y-coma)
* A[mbito de las variables (Scope) y Hoisting](https://stackedit.io/editor#%C3%A1mbito-de-las-variables-scope-y-hoisting)
* [Declaración de variables](https://stackedit.io/editor#declaraci%C3%B3n-de-variables)
* [Formato de declaración](https://stackedit.io/editor#formato-de-declaraci%C3%B3n)
* [Declaración de variables fuera de las sentencias](https://stackedit.io/editor#declaraci%C3%B3n-de-variables-fuera-de-las-sentencias)
* [Declaración de variables dentro de un contenedor(Reduciendo variables globales)](https://stackedit.io/editor#declaraci%C3%B3n-de-variables-dentro-de-un-contenedorreduciendo-variables-globales)
* [Comparación de variables](https://stackedit.io/editor#comparaci%C3%B3n-de-variables)
* [Otras comparaciones](https://stackedit.io/editor#otras-comparaciones)
* [Evaluación condicional](https://stackedit.io/editor#evaluaci%C3%B3n-condicional)
* [Notaciones cortas](https://stackedit.io/editor#notaciones-cortas)
* [Formato de nombres de variables y funciones](https://stackedit.io/editor#formato-de-nombres-de-variables-y-funciones)
* [En variables](https://stackedit.io/editor#en-variables)
* [En funciones](https://stackedit.io/editor#en-funciones)
* [Nombres y comentarios en ingles](https://stackedit.io/editor#nombres-y-comentarios-en-ingles)
* [Nombres con sentido](https://stackedit.io/editor#nombres-con-sentido)
* [Comentarios](https://stackedit.io/editor#comentarios)
* [Asignaciones y menciones](https://stackedit.io/editor#asignaciones-y-menciones)

Me gusta que mi código sea elegante y eficaz. La lógica debe ser directa para evitar errores ocultos, las dependencias deben de ser mí­nimas para facilitar el mantenimiento… El código limpio hace bien una cosa.

Para crear un código limpio se requiere de mucho esfuerzo, conocimiento, trabajo constante, y no siempre se consigue a la primera. Es por eso que cada lí­nea de código escrita, se debe analizar, sintetizar y volver a reescribir si es necesario para que cumpla con las caracterí­sticas mencionadas anteriormente.

Se recomienda que cuando se haya desarrollado una aplicación, o inclusive aún esté en fase de desarrollo, y has tenido una participación importante en el proyecto; aportando con cientos de miles de lí­neas de código y siguiendo las buenas prácticas de desarrollo, y en algún instante del desarrollo de la aplicación decides abandonar el proyecto; el grado de sufrimiento para el equipo será mucho menor dado que el equipo y quién te sustituya, estarán tranquilos sabiendo que el código que escribiste, sigue las buenas prácticas de desarrollo.

La productividad del equipo aumenta considerablemente y disminuye el sufrimiento dado que seguiste convenciones y buenas prácticas de desarrollo. Y te vas con la conciencia tranquila.

Estructura de la aplicación:

Assets:

Components:

* Se definen los componentes de Header, para mostrar las opciones laterales, el logo y el menú item.
* Nav: Componente que se encarga de renderizar el menú consultado de una fuente de datos externa.
* Input: Se encarga de redenrizar un elemento tipo input.
* Select: Se encarga de rendizar un elemento tipo select con sus opciones respectivas.
* MenuLink: se encarga de renderizar la opción del menú nav.
* Button: se encarga de renderizar el botón.

Config: Definición de variables y constantes utilizadas en los otros archivos.

Services: Definir los diferentes servicios que consumen fuentes de datos externos.

Pages: Se encarga de definir la página dinámica que mostrará cada una de las opciones de acceso en el menú.

Theme.js: Se encarga de construir un objeto con los diferentes colores que se utilizan en componentes y páginas de la aplicación.