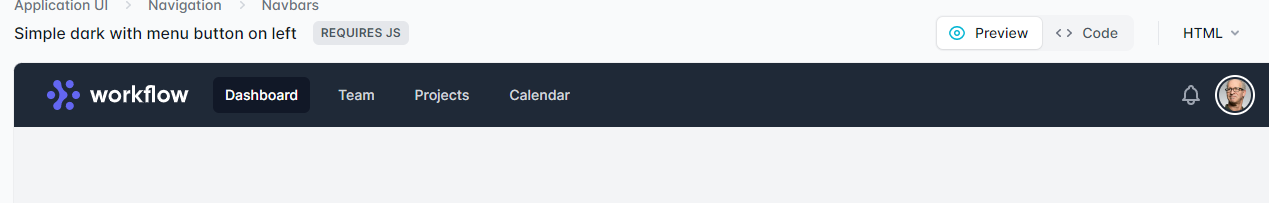
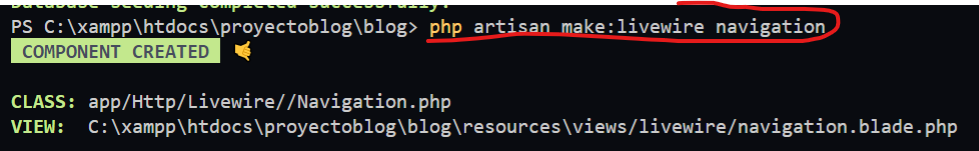
**Crear un menú responsive con Tailwind y Alpine para el blog**

1. Ir a la página oficial de tailwinds y dar clic en compontentes. Nos llevará a: <https://tailwindui.com/>
2. Clic en try it out. Nos llevará a: <https://tailwindui.com/#components>
3. Un menú interesante es el siguiente (pero no el que aparece de primero sino el más de abajo que tiene solo el código para el nav. También lo podemos buscar en la sección de navbars en el link <https://tailwindui.com/components/application-ui/navigation/navbars>):

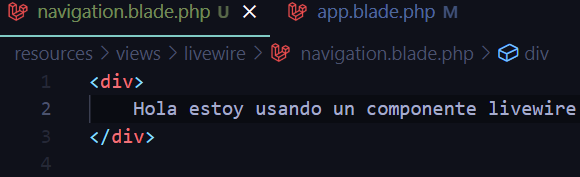


No obstante, tailwind solo provee el estilo css de este componente más no el javascript para su funcionalidad. Para ello, usaremos Alpine, que nos proveerá las funciones javascript requeridas.

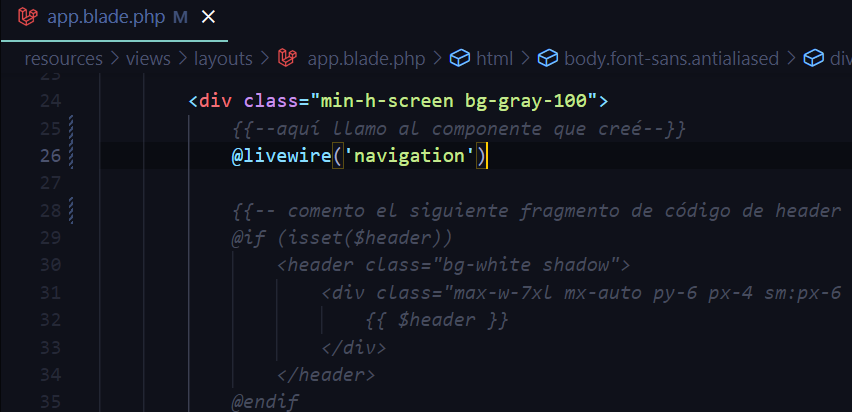
1. Para incluir todo esto en nuestro proyecto blog, es requerido crear un componente de livewire de la siguiente manera:



Como podemos ver se crean archivos tanto en Http como en resources/views. Hacemos un pequeño cambio en la vista:

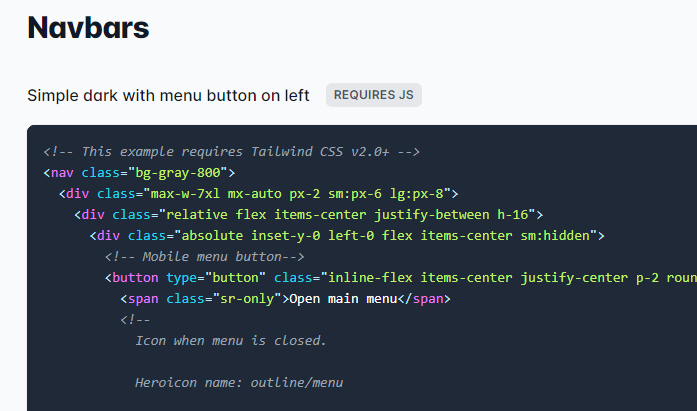


Ahora, es necesario llamar el componente mencionado, en nuestro layout (views/layouts/app.blade.php):

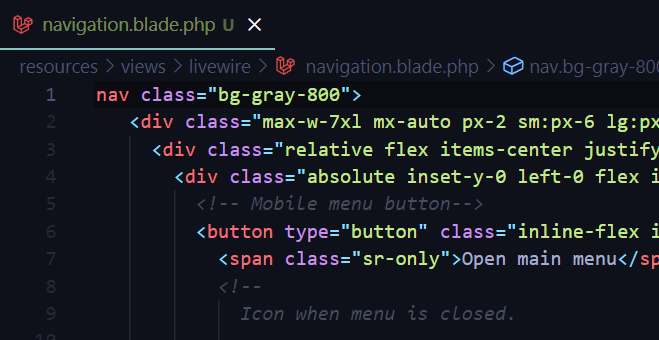


Y observamos el resultado en el navegador web.

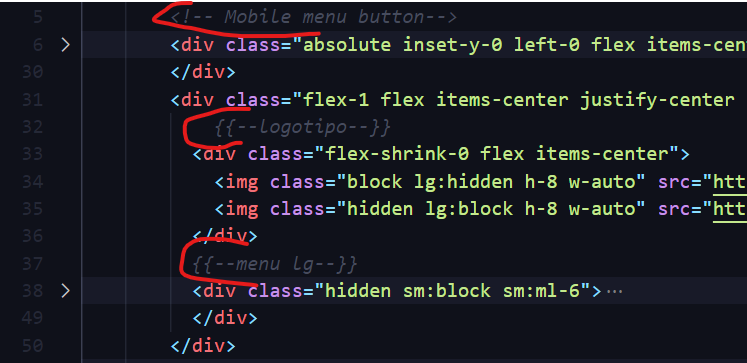
1. Copiamos el código del navbar:

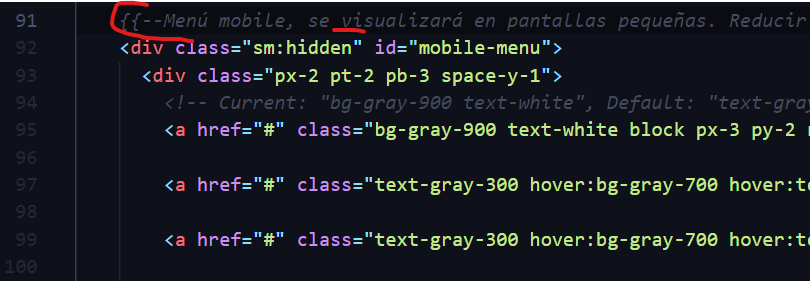


1. Pegamos este código en la vista navigation, y quitamos la primera línea que es un comentario y que puede dar un error:

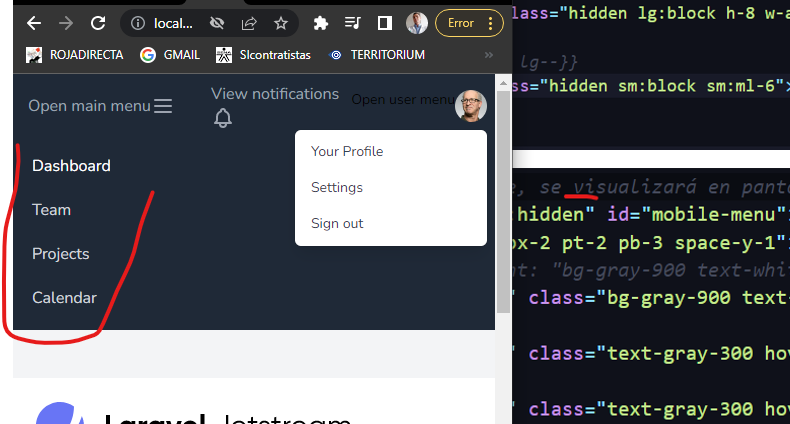


1. Luego es importante, hacer una diferenciación en las diferentes partes de este menú para lograr hacerle modificaciones (consultar el código trabajado). Para eso usamos comentarios. He aquí alguna de las capturas:





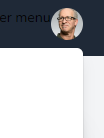
Esto último solo se podrá visualizar en el navegador si reducimos el tamaño del navegador web.



**Añadiendo funcionalidades al menú**

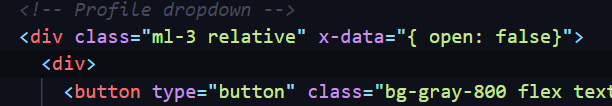
En este caso se requiere Alpine. No es necesario descargarlo, pues al instalar jetstream, este se descargó en el proyecto. Simplemente es usarlo.

1. Vamos a darle funcionalidad a…

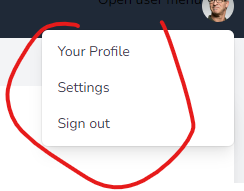


* 1. Esta sección se llama Profile dropdown:
  2. Alpine tratará el div respectivo como un componente. Para lograrlo, es necesario agregarle al div la propiedad x-data, de la siguiente manera:

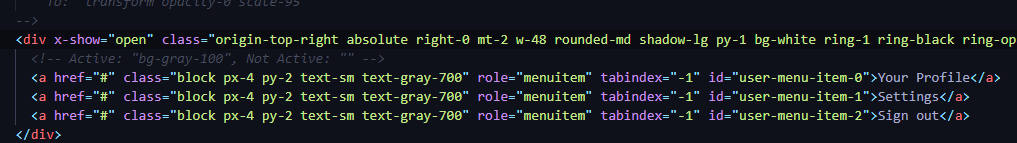
Se opta por definir una variable que se llamará open y tendrá el valor de false.

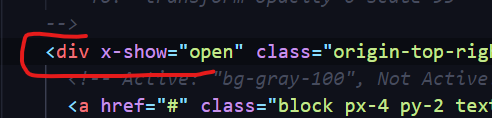


* 1. Queremos ocultar lo que se despliega de ese menú:

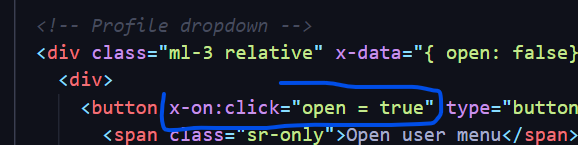


Para ello ubicamos la sección respectiva en el código y ponemos el código x-show en el div:

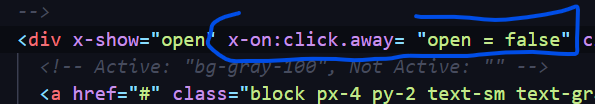




* 1. No obstante, el menú se oculta y ya no podemos verlo aun dando clic en el botón. Entonces, vamos a agregar un evento al botón, para que el ocultar o mostrar se maneje con clics. Los eventos los añadimos con x-on:



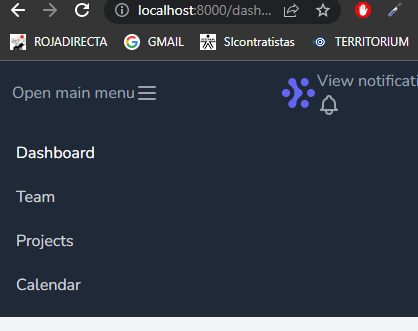
* 1. El menú se abrirá pero no se volverá a cerrar. ¿Cómo resolverlo? Tenemos que crear un evento para que al dar clic por fuera del menú, este desaparezca. Vamos al div donde están los elementos del menú y agregamos el siguiente x-on:



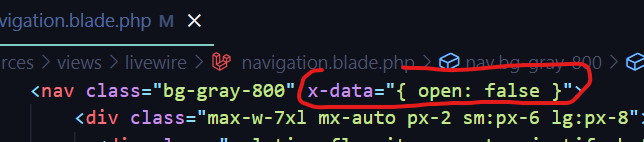
Al decir click.away estoy indicando que el evento se da al dar clic por fuera del div donde agregué dicho evento.

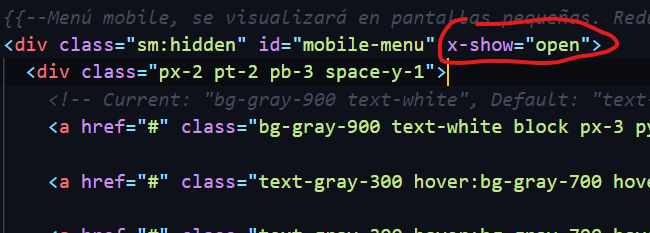
**Tip:** Podríamos confundirnos y escribir clic en lugar de click. Es importante notar la diferencia de escritura y hacerlo correctamente.

1. Queremos modificar la siguiente sección (aplica solo para tamaño móvil):

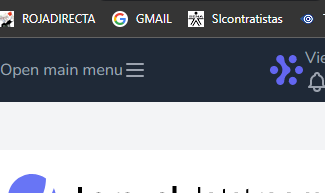


* 1. Para lo cual hacemos modificaciones tanto el nav principal como el div para el menú mobile:

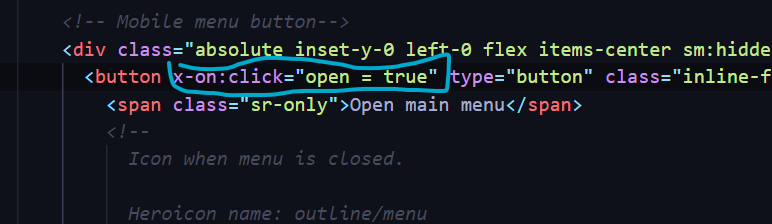




* 1. El menú respectivo desaparece, pero no vuelve a aparecer



* 1. Ahora, modificamos el botón (el icono de 3 rallitas), para que reaccione a un clic y así el menú se despliegue. Es el botón del menú de mobile:



9.10. Luego otra vez de vuelta en el menú mobil, le indicamos que al dar clic fuera de él, el menú se cierre: