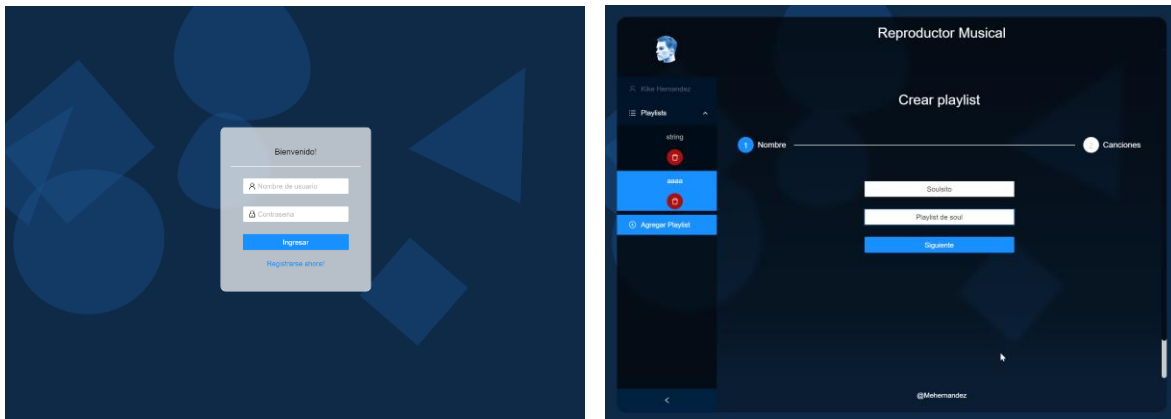


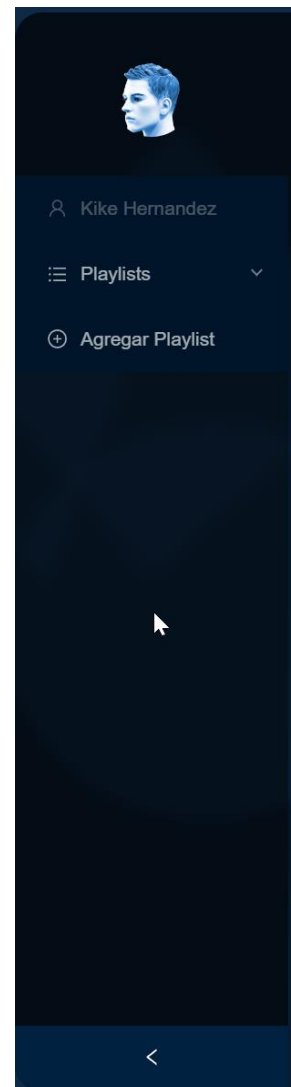
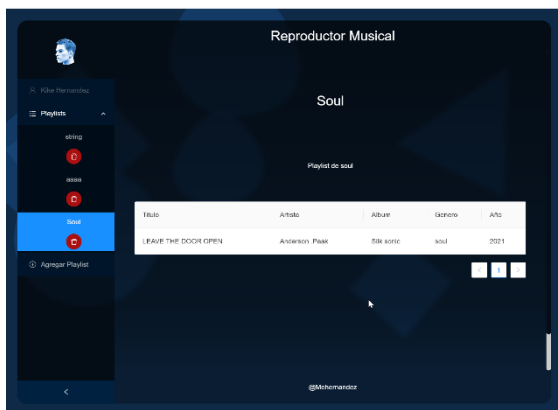
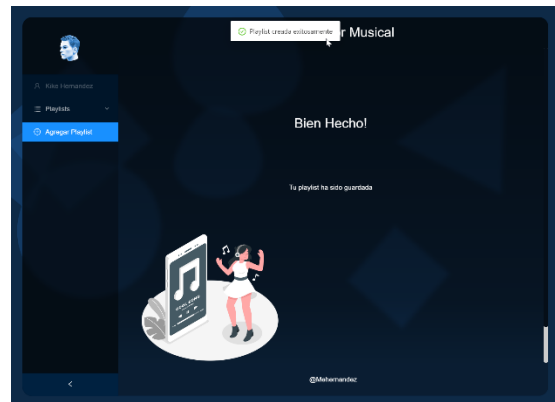
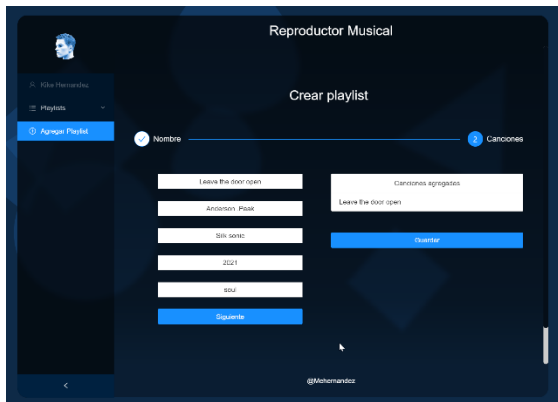
Mockups



Siendo honesto hace años que no uso figma y se me fue tiempo valioso buscando apps para mockups (estos son bocetos muy abstractos pero creo que van de la mano con la idea final que tenía...)

Diseño final



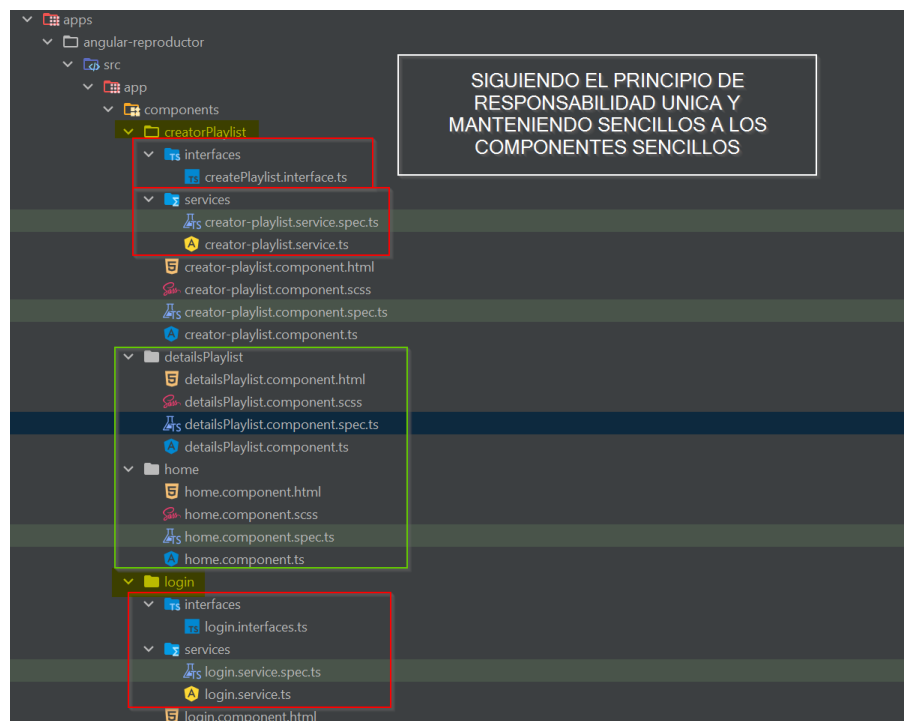


Faltan algunos ajustes de diseño y responsive aun
Pero hasta el momento cumple con lo requerido

Documentación proyecto de Reproductor musical

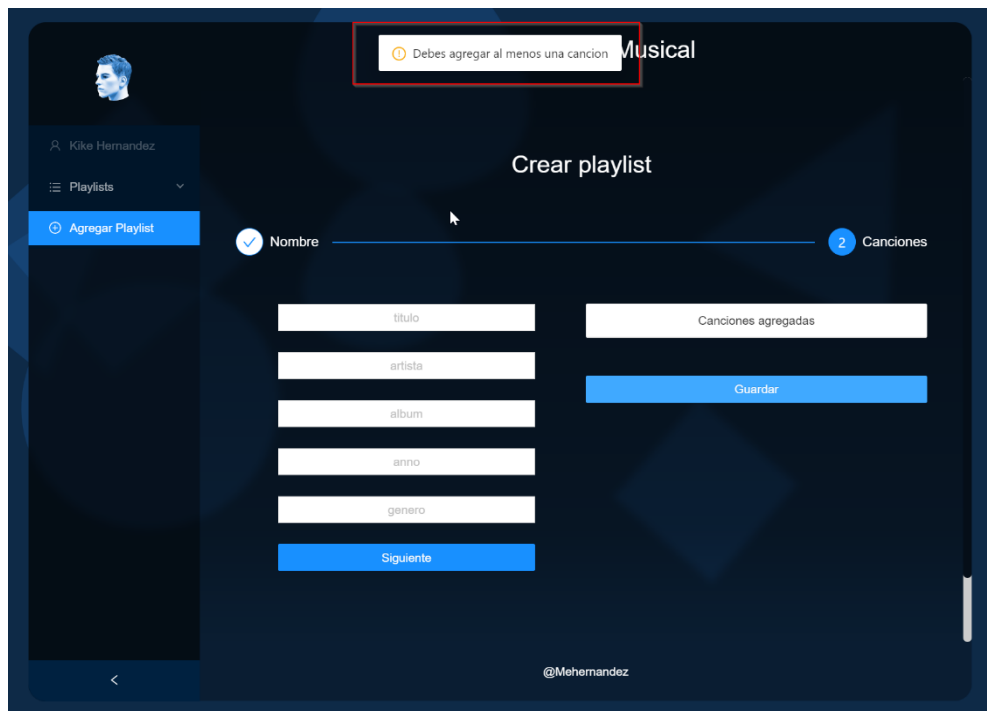
Para realizar esta prueba técnica decidí probar angular 17 y su nueva filosofía de componentes sin módulos (stand alone), también utilicé la última versión de la librería Nx para monorepositorios, el proyecto creado es conocido en la documentación oficial como “Angular monorepo” lo cual permite tener diferentes aplicaciones de diferentes frameworks todas bajo el mismo ecosistema de Nx microfrontend.

Elegí una distribución de carpetas similar a la que se usa en las versiones actuales cuando se desea trabajar “con módulos” o “modularizado” permitiendo así que la aplicación sea mucho mas escalable con el tiempo y sea más sencillo agregar o mantener funcionalidades, en esta versión de angular cada funcionalidad no podrá tener su modulo (porque angular 17 no usa módulos) pero cada componente si tendrá su carpeta de services, interfaces, etc... esto facilita el mantenimiento ya que en esas carpetas se encontraran los servicios o interfaces que necesita solo ese componente para funcionar, lo cual hace mas sencillo el debug y el refactor, si un servicio se usa en mas de 2 componentes se podría considerar pasarlo a la carpeta shared, aunque una de las ventajas de no trabajar con módulos son las importaciones, ya que al no haber módulos no debemos preocuparnos por llamadas en loop por dos módulos cruzados.



La documentación de Ant design aun no es muy clara para las versiones nuevas, sin embargo, en vista a la complejidad de manejarla dentro de un monorepositorio decidí centralizar los módulos dentro de un modulo custom en la carpeta shared y dejarlo disponible para los componentes de la app que lleguen a necesitar de él.

Se implemento un interceptor tanto para el manejo y alertas de errores como para la customización de cabeceras en las peticiones HTTP, el login quedo completamente funcional y también se implemento un guard el cual valida el acceso a la ruta home que es la protegida como tal.



Se implemento la inyección de dependencias de la nueva manera y también de la manera a la que estamos acostumbrados, de igual manera las directivas NgFor y NgIf, se implementó un nuevo pipe (algo sencillo como un uppercase solo con fines de probar lo que más se pudiera en la versión 17),

La estructura de proyecto implementada permite que el proyecto sea escalable y sencillo a la hora de mantener, separando los servicios y componentes compartidos de los servicios y componentes específicos minimizando así impactos grandes a la hora de corregir un bug o implementar una nueva funcionalidad.

Siempre es recomendable manejar el despliegue continuo para este tipo de proyectos, eso hace que se optimicen lo procesos de desarrollo, también ayuda bastante a generar flujos de trabajo personalizados y automatizar los despliegues de la manera en que mas le convenga a la aplicación, manejar el etiquetado a la hora de versionar el código también ayuda mucho a optimizar despliegues ya que evita que se realicen dichos despliegues por cambios menores que probablemente no ameriten un despliegue inmediato.

El compilador usado es Esbuild lo cual hace que el runtime y la compilación sea mucho más rápida, sin embargo, es bueno considerar que a pesar de que la app funciona bien, se pueden hacer unos refactor para que funcione aún mejor.

Por ultimo no está demás mencionar que al ser un mono repo las apps no se manejan como si fueran los frameworks normalmente, si es cierto que Nx es bastante cómodo, pero por ejemplo para instalar la librería de componentes de Ant design es necesario configurarla desde el inicio del proyecto, es bueno tener claro desde un inicio que es lo que se va a usar, pues hacer un cambio de este tipo a medio desarrollo puede impactar en gran manera la app ya que si se genera errores en la librería de componentes prácticamente la app se cae, también es recomendable considerar políticas para las librerías ya que algunas no se instalan normalmente con el NPM sino con el comando de la librería nx, aquí un ejemplo de lo que se tuvo que realizar a la hora de instalar la librería de componentes:

Crear el proyecto usando el generador Nx:

```
nx g @nrwl/angular:app reproductor
```

Agregar ng-zorro-antd al proyecto:

```
nx g ng-zorro-antd:ng-add --project reproductor
```

Nx tiene en cuenta la carpeta de donde se lanza el comando por lo cual hay que estar atento a la hora de crear cualquier servicio, pipe, componente, etc...

Quedo atento a cualquier duda o comentario sobre el repositorio expuesto, seguiré haciéndole refactor de algunas cosas que se pueden optimizar, creo que no se me escapa nada importante, sin embargo, si llegan a tener alguna duda no duden en hacérmela saber.

Miguel Hernandez
Desarrollador Frontend