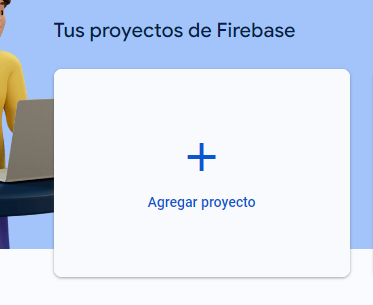
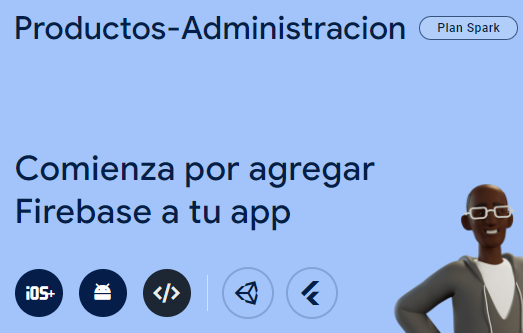
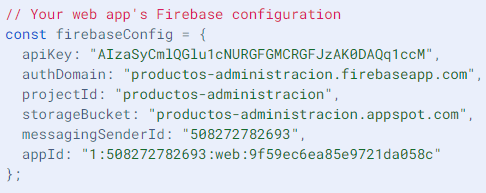
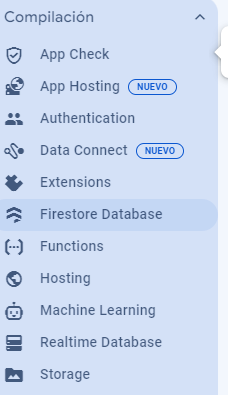
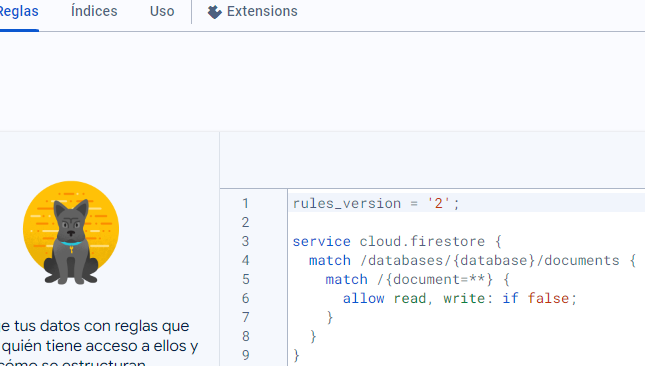
Pasos para crear proyecto

* Instalar ionic, y node.js
* Crear proyecto con ionic start nombre-proyecto. Seleccionamos el framework de angular, y le damos a proyecto blank, que es proyecto vacio.
* Borramos la capeta home, y de la misma manera borramos su ruta en app-routing.module.

Pasos para crear la base de datos en Firebase

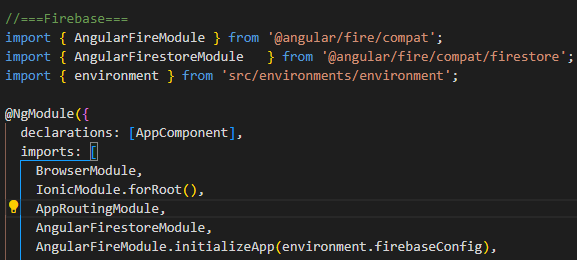
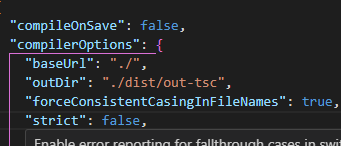
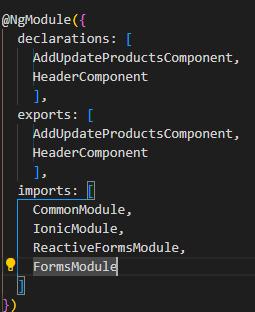
* Vamos a la pagina Firebase Console y creamos una cuenta.
* Luego de ello creamos un proyecto y le asignamos un nombre
*  Luego de ello vamos a crear una app web y le asignamos un nombre.
* Al darle crear nos va a salir el sdk que debemos asignar a nuestro proyecto en ionic. Para ello vamos al archivo environments de nuestro proyecto, y pegamos el siguiente código que nos arroja, en mi caso es el siguiente:
* Salimos de ahí y luego iremos a la pestaña compilación y agregamos el servicio que vayamos a usar en nuestro caso es firestore database.
* Luego nos saldrá un botón para crear la base de datos, damos ahí. Damo siguiente y por último crear. Y se creara la base de datos.
* Para un correcto funcionamiento se recomienda ir a la pestaña reglas, y quitamos las palabras ‘:if false’, que están dentro del fragmento de código que se muestra, esto se hace para que la base de datos permita todo tipo de consulta. Borramos eso junto con los dos puntos que hay después de la palabra write y por último le daremos en publicar. Y ya tendremos nuestra base de datos correctamente creada.

Instalar librerías de firebase

* Instalar librerías necesarias: **firebase-**con el comando npm i firebase

**Angular/fire-**con el comando npm i @angular/fire

Empezamos a codificar

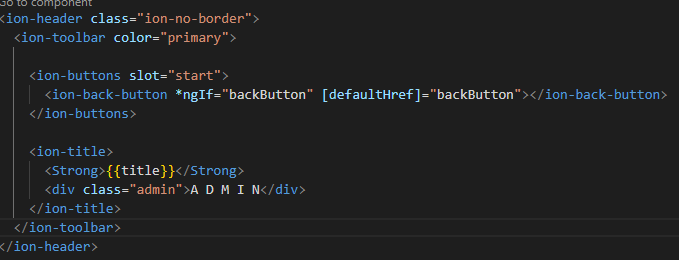
* Vamos al app-module.ts y ponemos los módulos de firebase necesarios, esto se hace con el fin de inicializar firebase, y usar estos módulos a lo largo del proyecto.
* Creamos dos servicios para la base de datos con el comando ionic g service servicios/firebase. Y el otro con nombre utils
* Para un correcto funcionamiento se recomienda ir al archivo ts.config.json y cambiar la propiedad strict de true a false, para tener un correcto funcionamiento con firebase. Así:
* Creamos un module con el comando ionic g module nombre-modulo
* Luego creamos el componente de agregar-y actualizar productos dentro del módulo shared. Con el comando ionic g c shared/components/nombre-componente. Acto seguido crearemos otro componente de header que iremos a utilizar en varias paginas y de la misma manera agregamos los componentes necesarios que miremos que vayamos a usar en varias páginas para reutilizar código.
* Iremos al shared.module.ts y declaramos e importamos los módulos de lo componentes que usaremos(debemos crear el array exports). Así:

Y en el imports debemos importar otros módulos que no serán útiles a la hora de crear nuestros formularios.

* Ahora para usar nuestros módulos simplemente llamamos el modulo del shared. Dentro de cada archivo module.ts de cada página, dentro del array imports.
* Ahora trabajaremos el header que es lo que se vera en todas, las paginas.

Primero creamos una variable de entrada en el component.ts así:

@Input() title?: string;

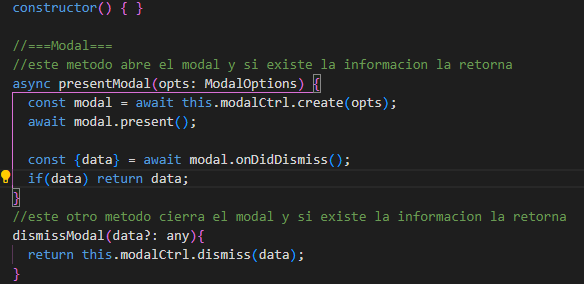
Y en el archivo .html ponemos nuestro header personalizado. Algo así:

* Y para que llame nuestro header debemos simplemente en las paginas en el archivo .html colocar el selector del header, y le pasamos el titulo que corresponda a cada página.

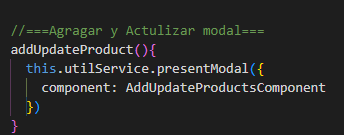
Creación de los formularios

* Para crear los formularios lo haremos con modales, así que iremos al servicio utils y añadiremos el controler del modal. Así:

modalCtrl = inject(ModalController);

* Luego crearemos dos métodos que abre el modal y el otro lo cierra y devuelven la información.
* Dentro de nuestro incio-components.ts vamos a inyectar el servicio llamado utils , así:

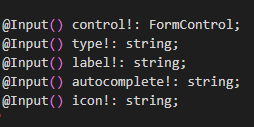
  utilService = inject(UtilsService)



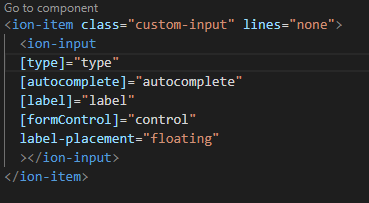
y creamos el método que llama el método que abre el modal:

ahora en nuestro archivo html del inicio creamos un fab, y le ponemos un click, y llamamos el método que acabamos de crear.

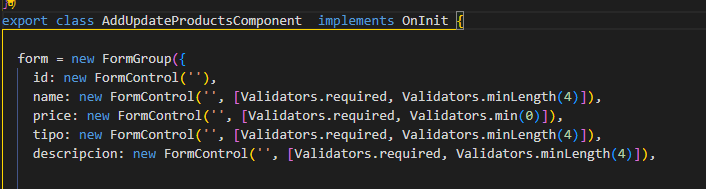
Crear formulario

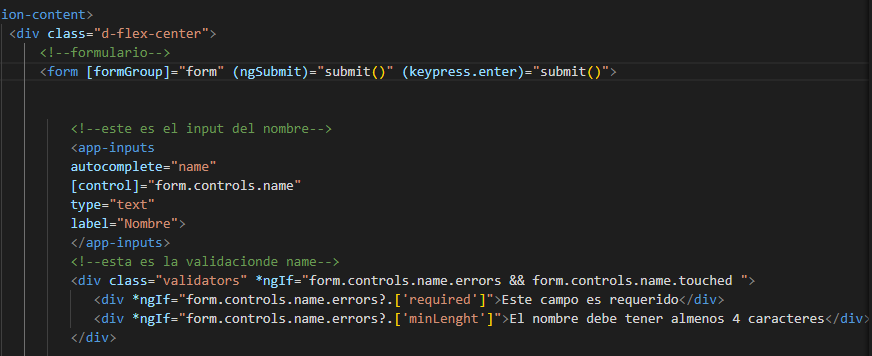
* Para crear el formulario vamos a crear un nuevo modal, que llamaremos inputs, en el cual ira la plantilla nuestra plantilla del formulario, esto con el fin de si queremos hacer un formulario mas adelante podamos reutilizar código.

Entonces en el .ts iría esto. Unas variables de entrada.

Y dentro del html iría esto. Que es nuestros inputs del formulario

Y agregaremos nuestro componente, al shared module para que nos funcione en las otras paginas

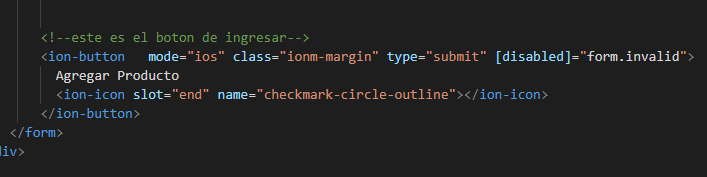
* Luego iremos a nuestro add-update-product .ts y agregaremos estas líneas de código, las cuales nos crea las variables del formulario y además les unas validaciones
* Y en el html crearemos un formulario el cual contendrá nuestros inputs haciendo la inserción del selector de nuestra plantilla del formulario anteriormente creada.



Texto

Descripción generada automáticamente

Y así se repite el código con para los otros campos.

En la última parte ira nuestro botón submit el cual tiene la validación de que se puede oprimir solo hasta que el usuario haya llenado todos los campos.