npm install firebase @angular/fire font-awesome bootstrap @ng-bootstrap/ng-bootstrap –save

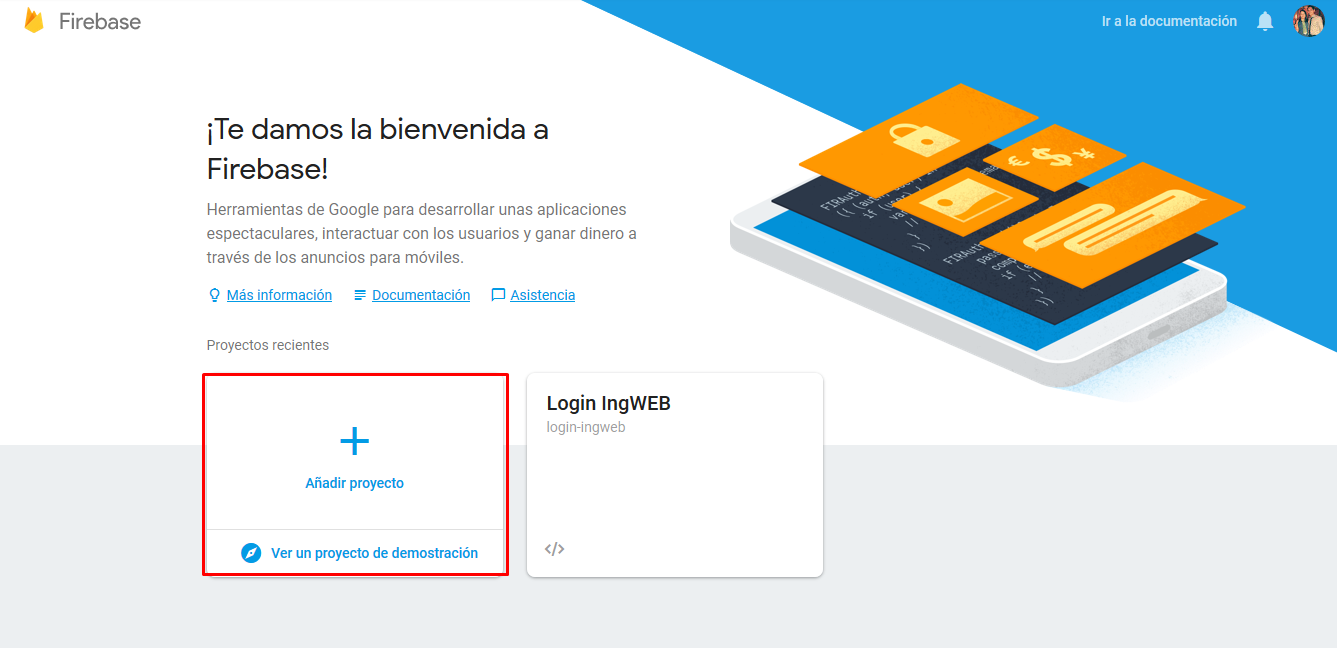
* install firebase @angular/fire: Instala librería de fire base que es la base de datos
* bootstrap @ng-bootstrap: permite modificar CSS y ng-bootstrap permite convertir el css en componentes de angular.
* font-awesome: permite mostrar iconos en la aplicación

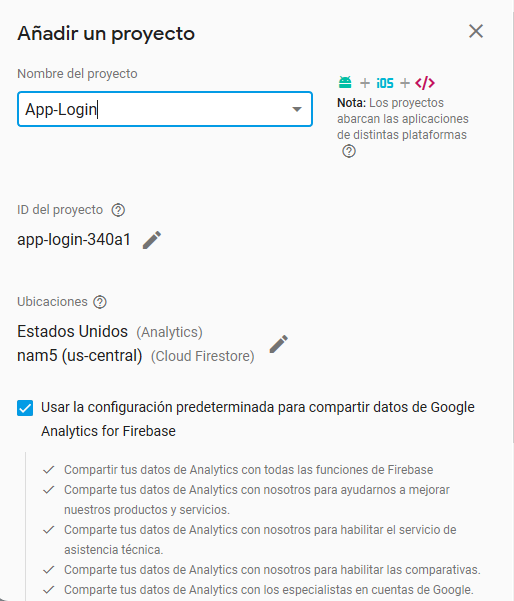
FireBase

<https://console.firebase.google.com/>

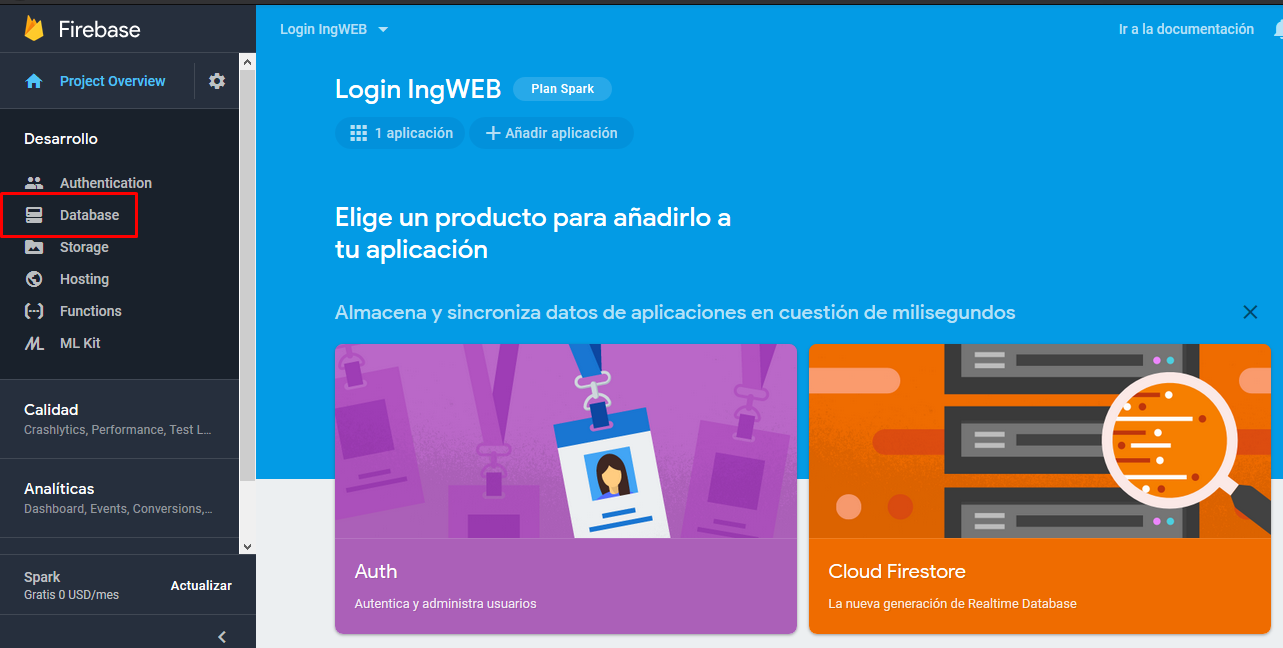
Debemos iniciar sesión con el correo de google.

Seleccionamos la opción de Añadir proyecto.





Ingresamos el nombre para nuestro proyecto, una vez creado el proyecto debemos crear la base de datos, en la pantalla que se muestra después de de crear el proyecto debemos seleccionar la opción de DataBase







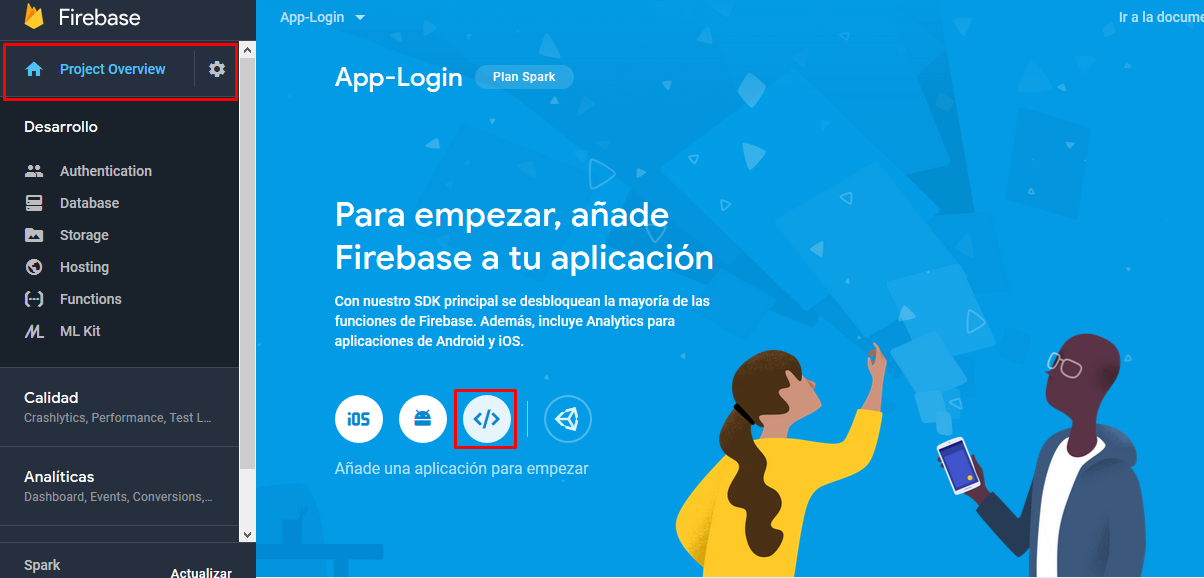
Al Seleccionar el modo de prueba se indica que cualquier usuario podrá leer o escribir en la BD,

Esta BD es NoSQL es decir no trabaja con tabla, trabaja con documentos a manera de un archivo Json, cada registro que ingresamos en la BD queda almacenado con un documento.

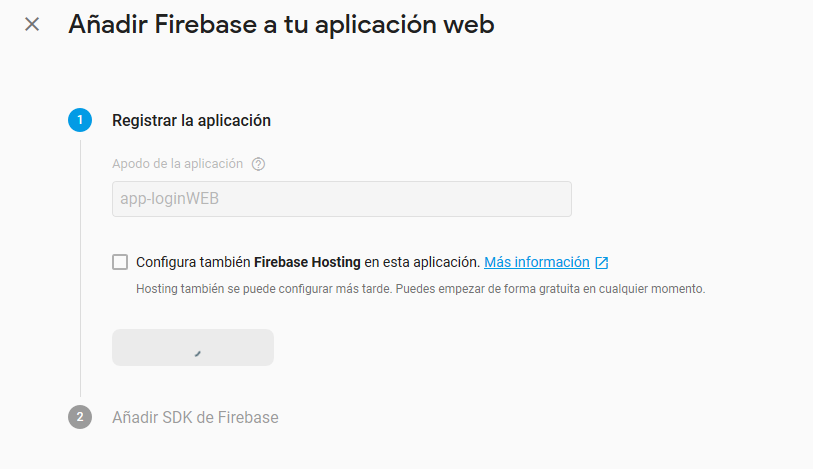
Los documentos se agrupan en una estructura llamada colección.

**Configuracion de la BD en Angular**

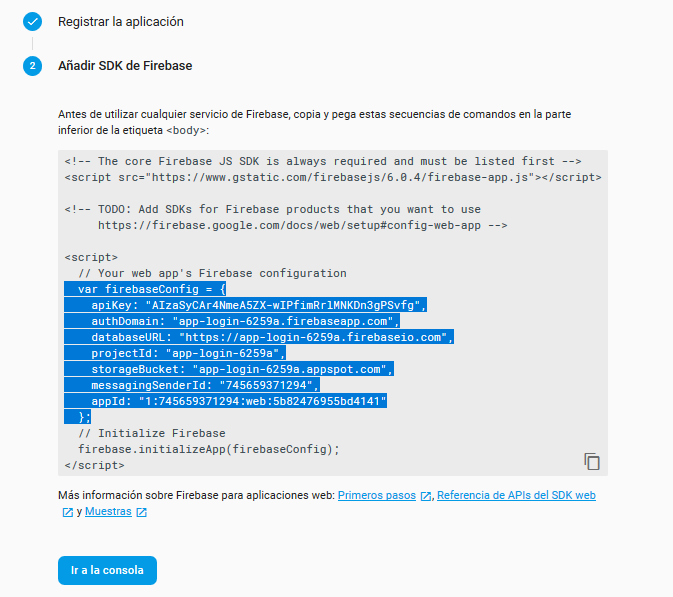
Seleccionamos la opción **Project Overview,** donde se cargara esta pantalla y damos click al icono de **</> WEB**



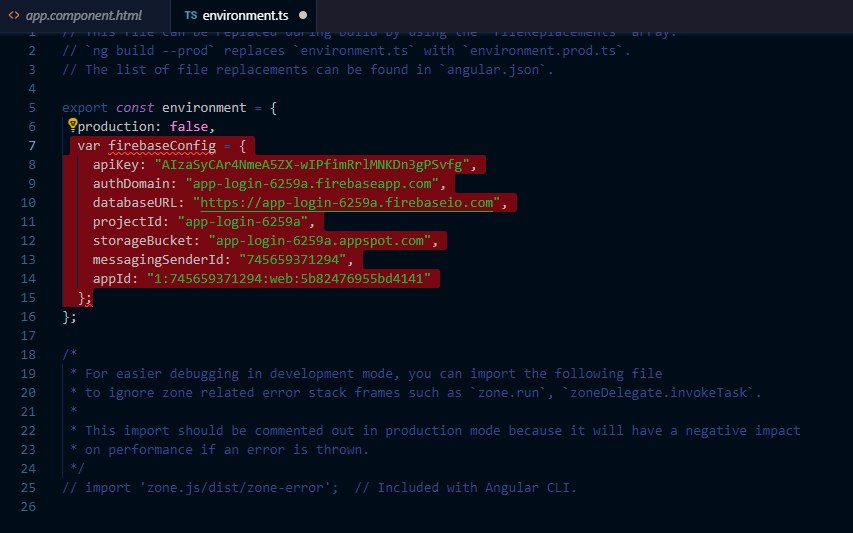
Registramos la aplicación

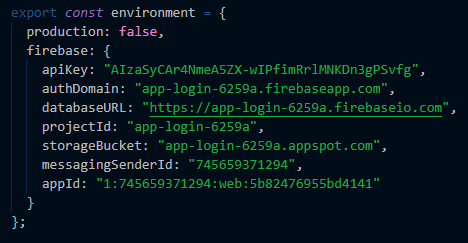


Y nos mostrara la siguiente pantalla, donde aparecerán las credenciales de la BD



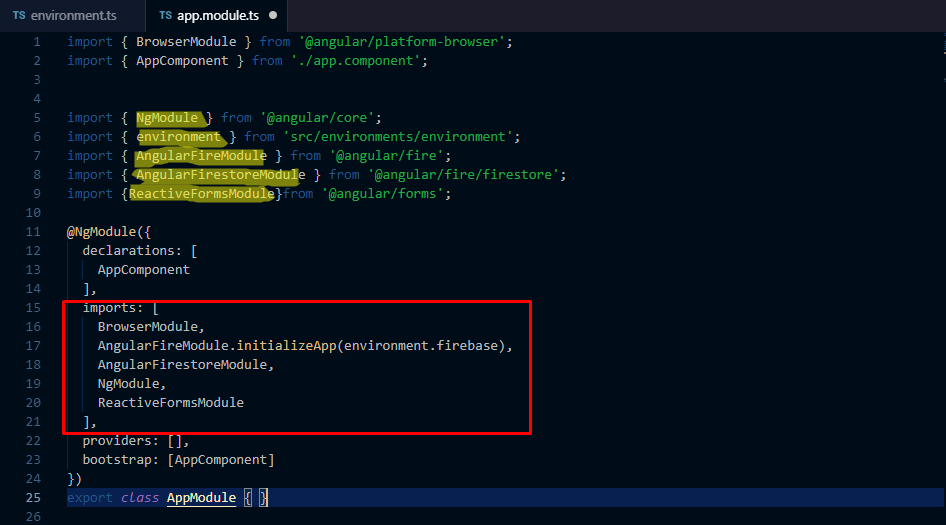
Estas credenciales las pegamos en un archivo de la aplicación que se llama **environment**





Luego de agregar este código, debemos configurar en el app.modules.ts los modulos de firebase.

Debemos agregar los imports que nos permitirá hacer uso de los modulos de firebase, bootstrap y ReactiveForms, adicional en la sección de imports debemos agregar la referencia a los import agregados.



Creamos los diferentes componentes a utilizar



En nuestro formulario agregamos un **formGroup,** en el cual entraremos a realizar las validaciones de los campos**.**

Hay dos maneras de hacer formularios en Angular, template y reactive. Las diferencias son:

Template Driven

fácil de usar

adecuado para escenarios simples.

similar a AngularJS

two way data binding (usando [(NgModel)] syntax)

código de componente mínimo

seguimiento automático del formulario y sus datos (manejado por Angular)

test unitarios difíciles de implementar

Reactive Forms

más flexible, pero necesita mucha práctica.

adecuado en escenarios complejos

No se realiza ningún data binding (immutable data model preferido por la mayoría de los desarrolladores)

Más código de componente y menos marcado HTML

Se pueden hacer transformaciones reactivas tales como

Manejando un evento basado en debounce time.

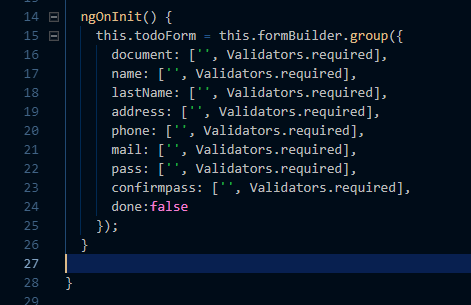
Manejo de eventos cuando los componentes son distintos hasta que se modifican.

Adición de elementos dinámicamente

test unitarios fáciles de usar

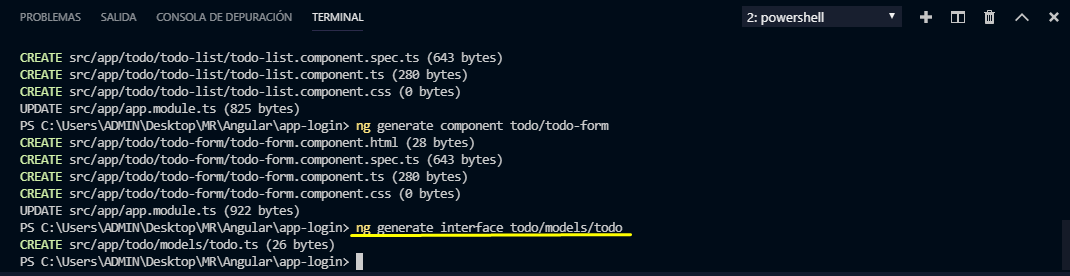
FormGroup -> es un cojunto de FormControls, el estado de este objeto depende del estado de todos sus objetos, es decir, si uno de los FormControl es inválido, el grupo entero es inválido.

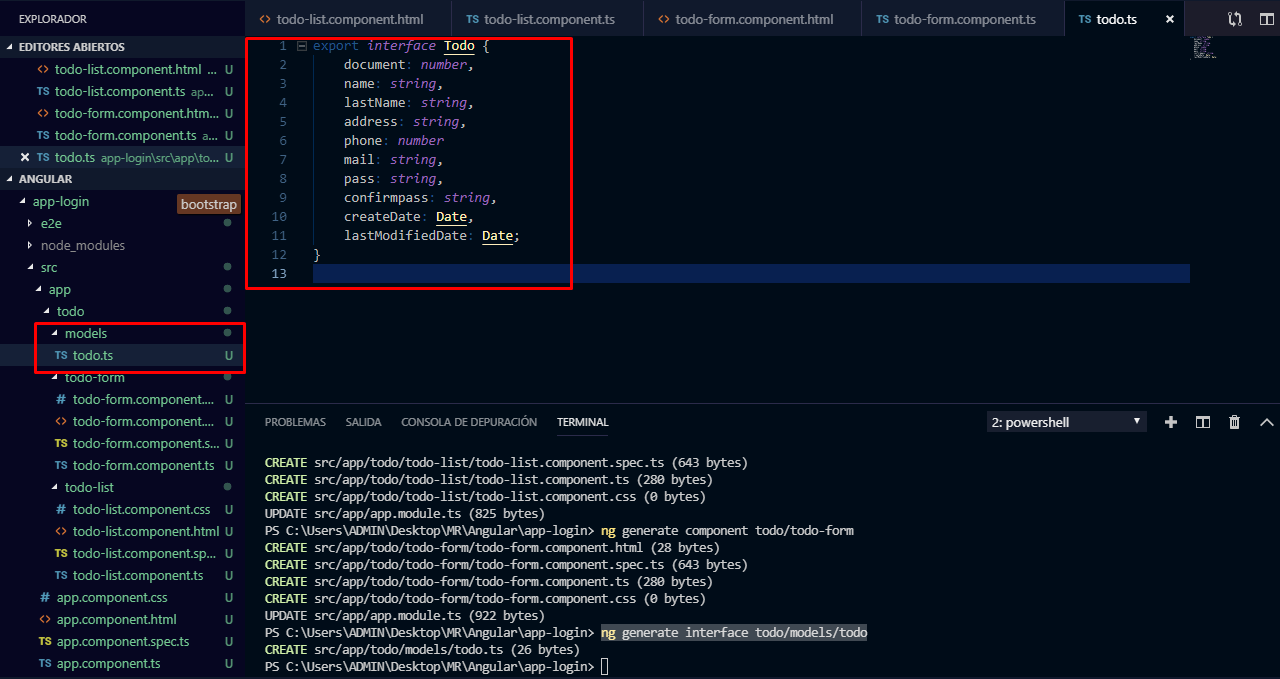
Desde nuestro component debemos agregar todos los campos que componen el formGroup indicando los que son requeridos



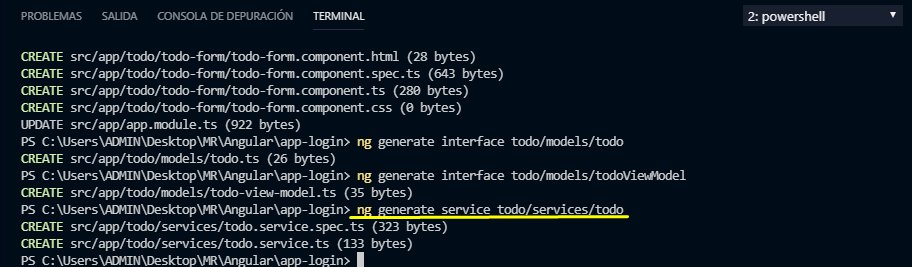
Creamos un modelo (representa la clase que tendrá los datos del formulario)

Ejecutamos en consola el comando **ng generate interface todo/models/todo**

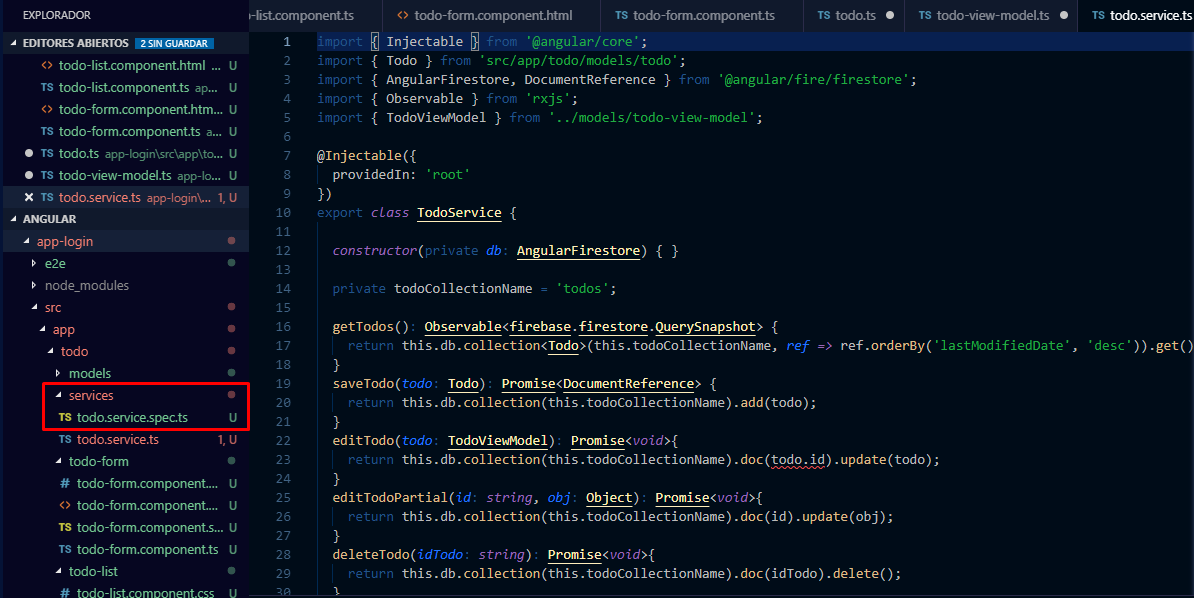




Ahora creamos un servicio



En este modulo realizaremos la conexión con la BD además del CRUD



<https://www.youtube.com/watch?v=6TRv1xT3Y-E>