

Angular + Firebase

Esse tutorial irá te ensinar como preparar um boilerplate inicial para trabalhar com firebase no Angular 12.

Primeiros passos

Primeiramente crie um novo projeto angular com:

```
ng new diario-app
```

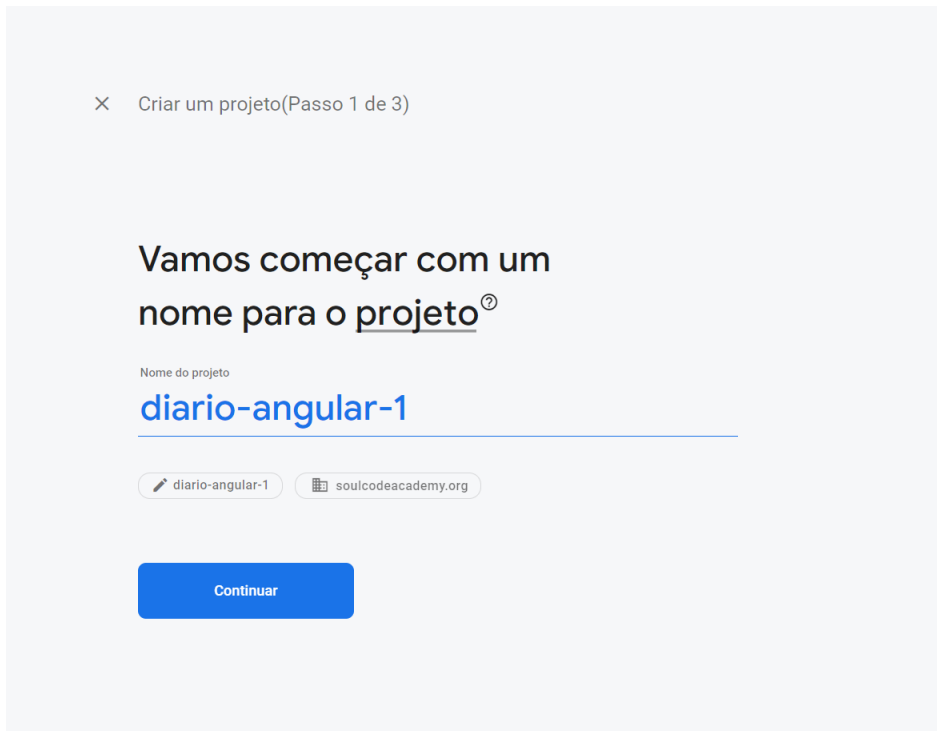
Em seguida configure dessa forma:

```
E:\Documentos\Projects\Angular>ng new test
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SCSS  [ https://sass-lang.com/documentation/syntax#scss ]
  Sass  [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less  [ http://lesscss.org ]
```

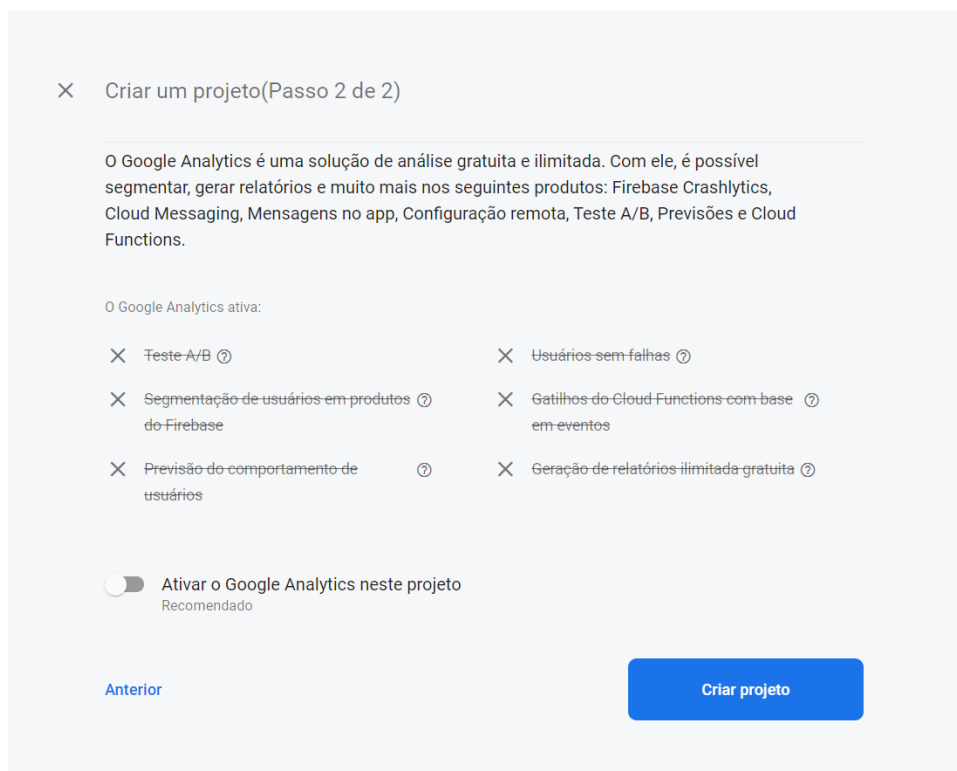
Iremos configurar rotas posteriormente e trabalharemos com Bootstrap para facilitar as coisas. Com o projeto criado, é necessário instalar as dependências necessárias para utilizar o Firebase no Angular.

Configurando um projeto no Firebase

Basta ter uma conta no Google para configurar um projeto do zero no Firebase.



Crie um projeto com o nome diario-angular, no meu caso utilizarei diario-angular-1. Clique em Continuar.



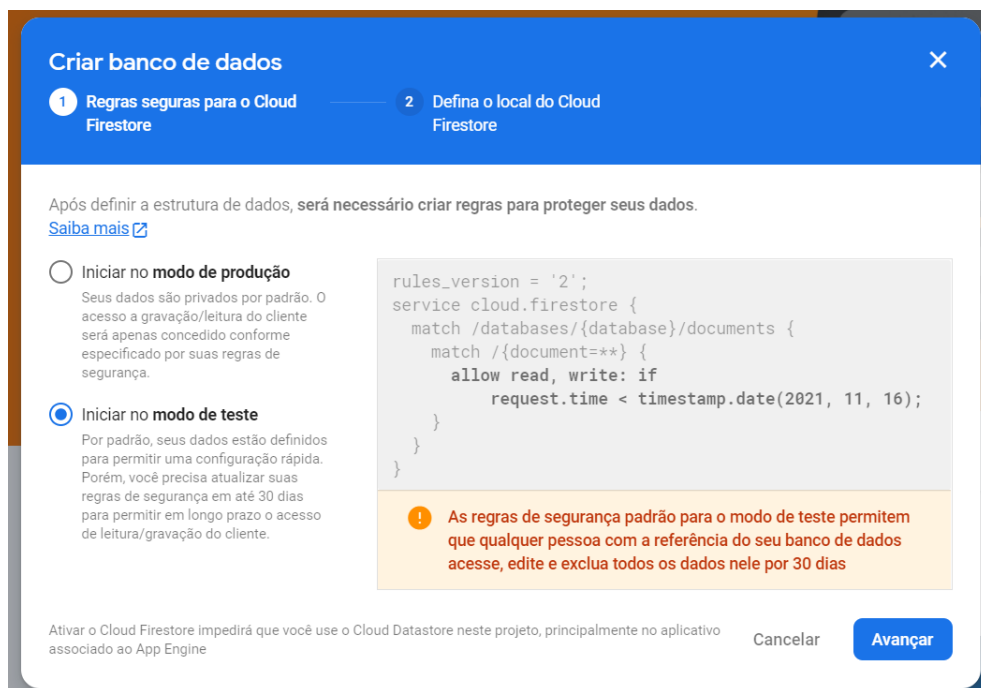
Desative a opção "Ativar o Google Analytics neste projeto", pois nesta lição não iremos trabalhar com os serviços do Analytics. Para saber mais sobre visite: [Google analytics](https://www.google.com/analytics/).

Enfim, clique em criar projeto, esse processo levará alguns segundos para ser concluído. Logo após concluir, abra o projeto e ative os serviços necessários nessa lição:

1. Firestore Database



Clique em Criar banco de dados. E defina as regras de segurança para o banco de dados. Para propósitos de desenvolvimento, iniciaremos no modo teste, porém essas regras podem ser alteradas posteriormente.



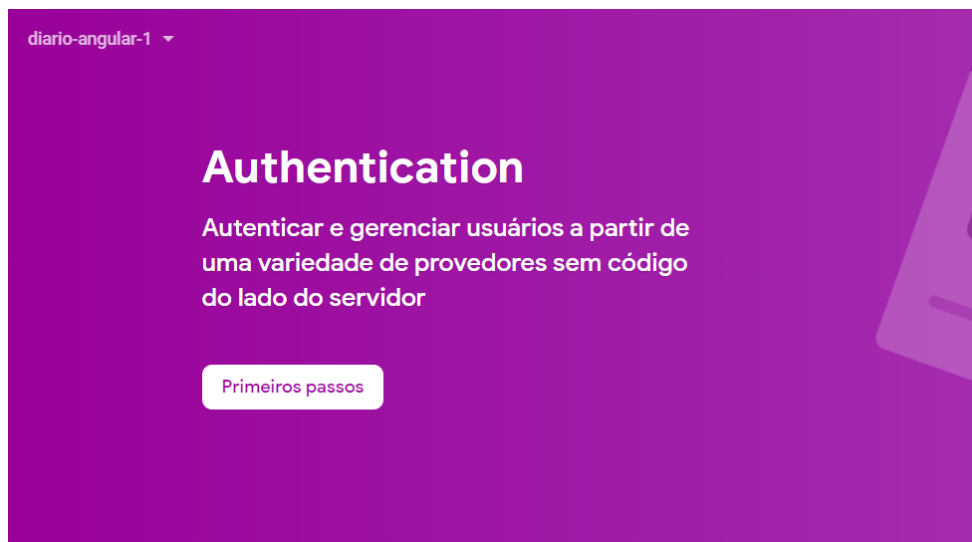
Após clicar em avançar, na seção 2 clique em Ativar e seu banco de dados já está configurado. Caso queira saber mais como definir regras de segurança para o Firestore, visite [Cloud Firestore Security Rules](#).

2. Firebase Storage

O serviço Storage não precisa de nenhuma configuração específica.


3. Firebase Authentication

No Authentication, clique em Primeiros passos:



Ele abrirá uma página para selecionar os provedores de login, como email, google, facebook e outros. Os dois principais são Email, Smartphone e Anônimo. Para esta lição utilizaremos o provedor por Email/Senha, para isso ativamos ele:

Provedores de login

 E-mail/senha

☒ Ativar

Permite que os usuários se inscrevam usando o endereço de e-mail e a senha deles. Nossos SDKs também fornecem verificação de endereço de e-mail, recuperação de senha e componentes essenciais para alteração do endereço de e-mail. [Saiba mais](#)

Link do e-mail (login sem senha)

☐ Ativar



Cancelar

Salvar

Sua página deve ficar assim:

Provedores de login

Adicionar novo provedor

Provedor	Status
 E-mail/senha	 ativado

Por fim iremos configurar um projeto web e obter as credenciais necessárias para que nosso app Angular consiga fazer acesso aos serviços do Firebase.

diario-angular-1

diario-angular-1 Plano Spark

Comece adicionando o Firebase ao seu aplicativo

iOS







Adicione um app para começar



Clique na tag HTML para configurar um projeto da web.

× Adicionar o Firebase ao seu app da Web

1 Registrar app

Apelido do app ?

diario-angular-1

☐ Também configure o **Firebase Hosting** para este app. [Saiba mais](#)

A configuração do Hosting também pode ser feita depois. Comece a usar a qualquer momento sem pagar nada.

Registrar app

2 Adicionar o SDK do Firebase

Clique em Registrar app e aparecerá as seguintes informações:

☒ Usar o npm ? ☐ Usar a tag <script> ?

Se você já estiver usando o [npm](#) e um bundler de módulo, como [webpack](#) ou [Rollup](#), é possível executar o seguinte comando para instalar o SDK mais recente:

```
$ npm install firebase
```

Após isso, inicie o Firebase e comece a usar os SDKs dos produtos.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCaIERAOMLz0QVw7kVx99F-wUXvZa0H-pA",
  authDomain: "diario-angular-1.firebaseio.com",
  projectId: "diario-angular-1",
  storageBucket: "diario-angular-1.appspot.com",
  messagingSenderId: "72497673516",
  appId: "1:72497673516:web:b46f6c2108a0e6b0fc2889"
};
```

```
// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Observação: essa opção usa o [SDK modular do JavaScript](#), que reduz o tamanho do SDK.

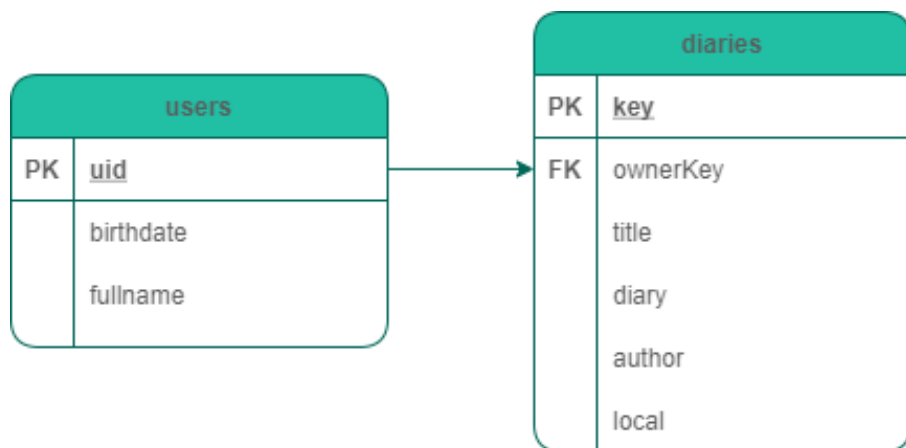
Saiba mais sobre o Firebase para Web: [Primeiros passos](#), [Referência da API Web SDK](#), [Amostras](#)

Continuar no console

Guarde/copie seu objeto pois ele será utilizado depois dentro do Angular. Enfim você configurou a parte do servidor, e ela está pronta para ser utilizada pelo seu app.

O que vamos construir?

Iremos construir uma aplicação completa com CRUD e autenticação utilizando os serviços do Firebase! Nossa aplicação consiste de um diário de viagens onde os usuários podem colocar suas opiniões sobre algum lugar/evento que foram. Ou seja, podemos imaginar um conjunto de dados chamado ****diaries**** que contém os diários de cada usuário, vinculados pela **uid(user identification)** de autenticação. E uma coleção **users** que guarda informações extras do usuário, como data de nascimento, e etc. Em um banco de dados relacional teríamos praticamente esse esquema:



Instalando dependências

Para trabalhar no Angular com Firebase vamos instalar algumas dependências em nosso projeto. Primeiramente instalamos o Bootstrap:

```
ng add @ng-bootstrap/ng-bootstrap
```

Em seguida instalamos o AngularFire:

```
ng add @angular/fire
```

Esse comando irá exigir algumas configurações para definir quais serviços do Firebase serão utilizados no projeto do Angular:

```
? What features would you like to setup? (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
( ) Performance Monitoring
(*) Cloud Storage
( ) Remote Config
> ( ) ng deploy -- hosting
(*) Authentication
(*) Firestore
( ) Realtime Database
(Move up and down to reveal more choices)
```

Utilizando espaço para marcar e as setas direcionais para ver as opções, marque apenas: **Authentication, Firestore e CloudStorage**. Então pressione ENTER para finalizar.

Configurando o AngularFire

A configuração está quase pronta, porém agora é preciso adicionar os módulos e configurações necessárias. Primeiro vamos configurar nossas credenciais:

TS environment.ts X

src > environments > TS environment.ts > ...

```
1 // This file can be replaced during build by using the `fileReplacements` array.
2 // `ng build` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 export const environment = {
6   production: false,
7   firebase: {
8     apiKey: 'AIzaSyCaIERAOMLzOQVw7kVx99F-wUXvZaOH-pA',
9     authDomain: 'diario-angular-1.firebaseio.com',
10    projectId: 'diario-angular-1',
11    storageBucket: 'diario-angular-1.appspot.com',
12    messagingSenderId: '72497673516',
13    appId: '1:72497673516:web:b46f6c2108a0e6b0fc2889',
14  },
15 };
```

No arquivo **enviroment.ts** cole o objeto com as credenciais. Realize o mesmo processo no arquivo **enviroment.prod.ts**. Em seguida iremos configurar o **AppModule**:

TS app.module.ts M X

src > app > TS app.module.ts > ...

```
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppComponent } from './app.component';
5 import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
6 import { AngularFireModule } from '@angular/fire/compat';
7 import { AngularFireStoreModule } from '@angular/fire/compat/firestore';
8 import { AngularFireStorageModule } from '@angular/fire/compat/storage';
9 import { AngularFireAuthModule } from '@angular/fire/compat/auth';
10 import { environment } from 'src/environments/environment';
11 import { FormsModule } from '@angular/forms';
12
13 @NgModule({
14   declarations: [
15     AppComponent
16   ],
17   imports: [
18     BrowserModule,
19     NgbModule,
20     AngularFireModule.initializeApp(environment.firebase),
21     AngularFireStoreModule,
22     AngularFireStorageModule,
23     AngularFireAuthModule,
24     FormsModule
25   ],
26   providers: [],
27   bootstrap: [AppComponent]
28 })
29 export class AppModule { }
30
```

Seu **AppModule** deve estar com estes imports para que as dependências sejam injetadas para os componentes. Seu projeto está finalmente pronto para criar os componentes e services necessários! A versão final está disponível no [Github](#) e foi hospedada também utilizando o [Firebase](#).

Montando estrutura

Você pode agora opcionalmente criar os arquivos e componentes vazios que serão construídos em aula rodando os seguintes comandos:

```
ng g c navbar
ng g c home
ng g c login
ng g c signup
ng g c diarios
ng g c diarios/add-diario
ng g c diarios/edit-diario

ng g s services/auth
ng g s services/diarios
ng g s services/error
ng g s services/upload-image
```

Configurando as rotas

Considerando o passo anterior, iremos configurar as rotas do nosso app:

```
ng g m app-routing --flat --module=app
```

Esse comando gerará um novo módulo em src/app, enfim configure-o da seguinte forma:

```
@NgModule({
  declarations: [],
  imports: [
    CommonModule,
    RouterModule.forRoot(routes)
  ],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

Enfim seu AppRoutingModule precisa estar assim inicialmente:


```

import { DiariosComponent } from './diarios/diarios.component';
import { SignupComponent } from './signup/signup.component';
import { LoginComponent } from './login/login.component';
import { HomeComponent } from './home/home.component';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { Route, RouterModule } from '@angular/router';

const routes: Route[] = [
  { path: '', component: HomeComponent },
  { path: 'login', component: LoginComponent },
  { path: 'signup', component: SignupComponent },
  { path: 'diarios', component: DiariosComponent },
];

@NgModule({
  declarations: [],
  imports: [
    CommonModule,
    RouterModule.forRoot(routes)
  ],
  exports: [RouterModule],
})
export class AppRoutingModule {}

```

Posteriormente iremos adicionar **Guards** nas rotas necessárias.

Hospedar no Firebase Hosting

Essa etapa é interessante realizar após construir toda aplicação. Para isso é necessário instalar as ferramentas do firebase com:

```
npm install -g firebase-tools
```

Antes de tudo realize login para que a ferramenta tenha as permissões necessárias:

```
firebase login
```

Esse comando irá abrir uma janela do navegador para que você realize login com uma conta google. Após realizar o procedimento, pode fechar a janela. Agora iremos configurar o firebase hosting, rode o comando `firebase init`:

```
E:\Documentos\Projects\Angular\test>firebase init

##### 
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##### 
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##

You're about to initialize a Firebase project in this directory:

  E:\Documentos\Projects\Angular\test

? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices.
e
  ( ) Firestore: Configure security rules and indexes files for Firestore
  ( ) Functions: Configure a Cloud Functions directory and its files
> (*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
  ( ) Hosting: Set up GitHub Action deploys
  ( ) Storage: Configure a security rules file for Cloud Storage
  ( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)
```

Com espaço selecione **Hosting: Configure files for Firebase Hosting** e pressione ENTER. Em seguida selecione **Use an existing project.:**

```
? Please select an option: Use an existing project
? Select a default Firebase project for this directory:
  auth-tests-f637f (google-auth-tests)
> diario-angular-1 (diario-angular-1)
  diario-de-viagens-6c221 (diario-viagens)
  diario-viagens-3d8c6 (diario-viagens)
  experimento-soulcode (experimento-soulcode)
  flutter-teams (flutter-teams)
```

Selecione o projeto que está sendo utilizado pelo angular. Agora você irá configurar algumas opções de hospedagem:

```
? What do you want to use as your public directory? dist/test
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
```

No caso foi escolhido **dist/test** pois *dist* é o diretório que o Angular coloca os builds e **test** é o nome do projeto, que no caso de vocês pode ter sido *diario-app*. Gere o build do Angular utilizando **ng build**. Após o processo terminar, basta realizar deploy com o firebase hosting toda vez que realizar um novo build do seu app: **firebase deploy**.

```
E:\Documentos\Projects\Angular\test>firebase deploy

=== Deploying to 'diario-angular-1'...

i  deploying hosting
i  hosting[diario-angular-1]: beginning deploy...
i  hosting[diario-angular-1]: found 7 files in dist/test
+  hosting[diario-angular-1]: file upload complete
i  hosting[diario-angular-1]: finalizing version...
+  hosting[diario-angular-1]: version finalized
i  hosting[diario-angular-1]: releasing new version...
+  hosting[diario-angular-1]: release complete

+  Deploy complete!

Project Console: https://console.firebase.google.com/project/diario-angular-1/overview
Hosting URL: https://diario-angular-1.web.app
```

Visite a URL para verificar se está tudo funcionando com as rotas e a aplicação no geral.