

# PRÁTICAS AVANÇADAS EM DESENVOLVIMENTO WEB

Davi Schneid - [davi.schneid@gmail.com](mailto:davi.schneid@gmail.com)

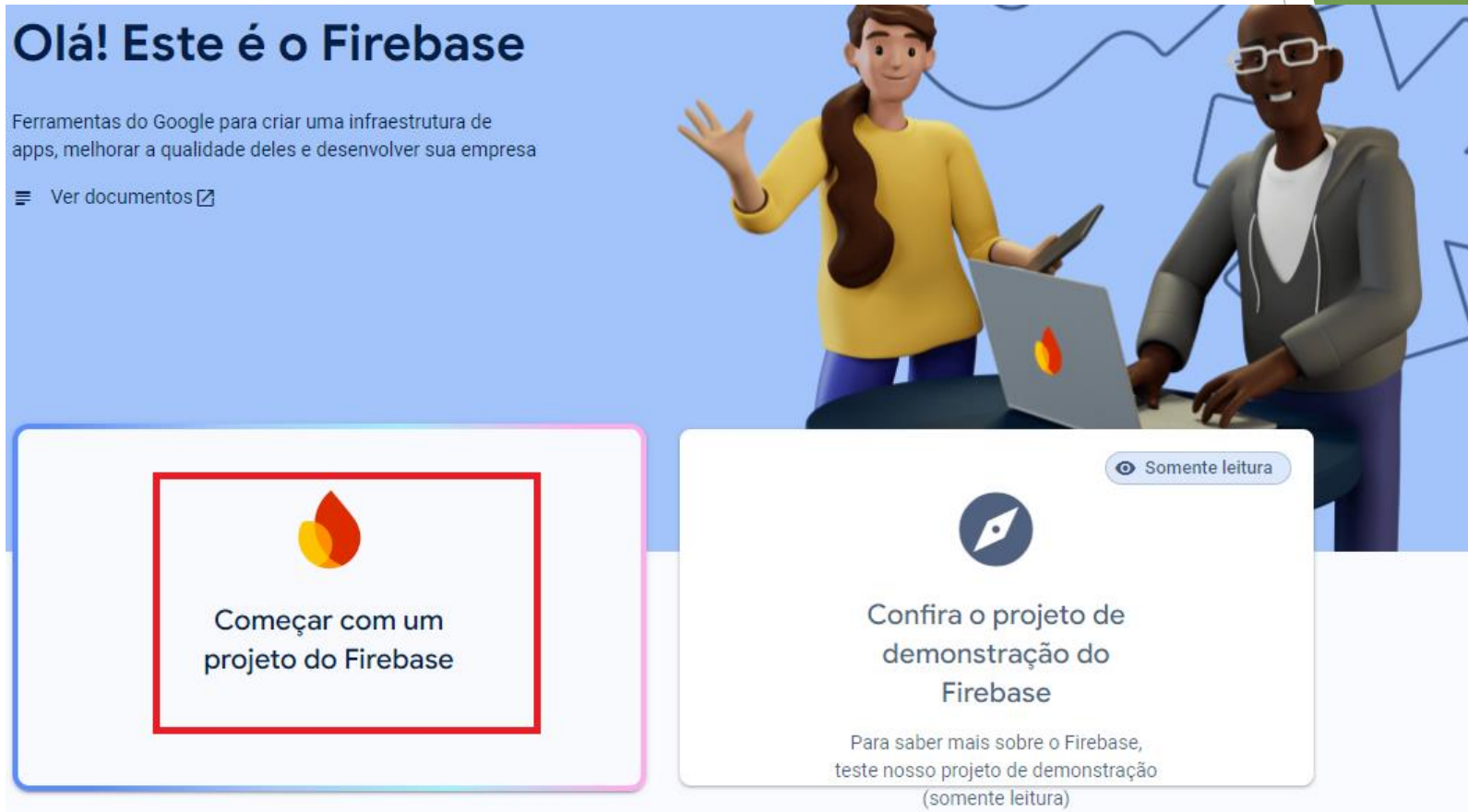
05/08/2024

# Agenda

- ▶ Criando aplicação backend com Firebase
  - ▶ Configuração do Projeto Firebase
  - ▶ Instalação do SDK do Firebase
  - ▶ Configuração do Firebase no Node.js
- ▶ Instalações necessárias
  - ▶ `npm init -y`
  - ▶ `npm install firebase-admin`
  - ▶ `npm install firebase`
  - ▶ `npm install -g firebase-tools`
  - ▶ `npm install dotenv`
  - ▶ `npm install express`
  - ▶ `npm install swagger-jsdoc`
  - ▶ `npm install swagger-ui-express`

# API Firebase

- ▶ Criar o projeto no Firebase
- ▶ Acessar <https://console.firebase.google.com/>



# API Firebase


- ▶ Criar o projeto no Firebase

× Criar um projeto

Vamos começar nomeando o projeto<sup>?</sup>

Nome do projeto

3miltalentos

 miltalentos-d9445

☒

 Aceito os [Termos do Firebase](#).

☒

 Confirmo que vou usar o Firebase exclusivamente para fins relacionados ao meu comércio, empresa, ofício ou profissão.

Já tem um projeto do Google Cloud?  
[Adicionar o Firebase ao projeto do Google Cloud](#)

Continuar

# API Firebase

- Clicar em continuar

× Criar um projeto

## Google Analytics para seu projeto do Firebase

O Google Analytics é uma solução de análise ilimitada e sem custos financeiros. Com ele, é possível segmentar, gerar relatórios e muito mais nos seguintes produtos: Firebase Crashlytics, Cloud Messaging, Mensagens no app, Configuração remota, Teste A/B e Cloud Functions.

O Google Analytics ativa:

 Teste A/B 

 Segmentação de usuários em produtos   
do Firebase

 Registros de navegação estrutural no   
Crashlytics

 Ativar o Google Analytics neste projeto  
Recomendado

 Gatilhos do Cloud Functions com base   
em eventos

 Geração de relatórios ilimitada gratuita 

AnteriorContinuar

# API Firebase

× Criar um projeto

## Configurar o Google Analytics

Localização do Analytics ⓘ

Brasil

O Google Analytics é uma ferramenta de negócios. Use-a exclusivamente para fins relacionados ao seu comércio, negócio, ofício ou profissão.

Configurações de compartilhamento de dados e termos do Google Analytics



Usar as configurações padrão para o compartilhamento de dados do Google Analytics. [Saiba mais](#)

- × Compartilhe seus dados do Analytics com o Google para melhorar os produtos e serviços da empresa
- ✓ Compartilhe seus dados do Analytics com o Google para ativar o Comparativo de mercado
- ✓ Compartilhe seus dados do Analytics com o Google para ativar o suporte técnico
- ✓ Compartilhe seus dados do Analytics com os especialistas em contas do Google



Eu aceito os [Termos do Google Analytics](#)

Após a criação do projeto, uma nova propriedade do Google Analytics será criada e vinculada ao seu projeto do Firebase. Esse processo permitirá o fluxo de dados entre os produtos. Os dados da propriedade do Google Analytics exportados para o Firebase ficam sujeitos aos Termos de Serviço do Firebase, e os dados do Firebase importados para o Google Analytics ficam sujeitos aos Termos de Serviço do Google Analytics. [Saiba mais](#)

[Anterior](#)

**Criar projeto**

# API Firebase

 **Firebase**

[🏠 Visão geral do pr...](#) [⚙️](#)

IA generativa

✦ [Build with Gemini](#) **NOVO**

Categorias dos produtos

[Criação](#) ▾

[Executar](#) ▾

[Analytics](#) ▾

🗪 Todos os produtos

Ferramentas de desenvolvimento relacionadas

[IDX](#) ⓘ ⓘ

[Checks](#) ⓘ ⓘ

3miltalentos ▾



3miltalentos

✓ Seu projeto do Firebase está pronto

[Continuar](#)

3miltalentos [Plano Spark](#)

## Comece adicionando o Firebase ao seu aplicativo



Adicione um app para começar



as e eventos do Firebase [Inscreva-se](#)


# API Firebase

× Adicionar o Firebase ao seu app da Web

1 Registrar app

Apelido do app ⓘ

☒ Configure também o **Firebase Hosting** para este app. [Saiba mais](#) ⓘ  
O Hosting pode ser configurado a qualquer momento sem custos financeiros.  

 miltalentos-d9445 (Nenhuma implantação ainda) ▼

Registrar app

2 Adicionar o SDK do Firebase

3 Instalar a CLI do Firebase


4 Implantar no Firebase Hosting




# API Firebase

- ▶ Baixar o arquivo `serviceAccountKey.json` do Firebase Console e colocá-lo na pasta `firebase` do backend.
  - ▶ Acesse o Firebase Console:
  - ▶ Firebase Console
  - ▶ Selecione seu projeto:
  - ▶ Na página inicial do Firebase Console, selecione o projeto para o qual você deseja baixar a chave.
  - ▶ Vá para as configurações do projeto:
  - ▶ Clique no ícone de engrenagem no canto superior esquerdo, ao lado do nome do projeto, e selecione "Configurações do projeto" no menu suspenso.
  - ▶ Selecione a aba "Contas de serviço":
  - ▶ No menu lateral esquerdo, clique em "Contas de serviço".
  - ▶ Gerar nova chave privada:
  - ▶ Na seção "Firebase Admin SDK", clique em "Gerar nova chave privada". Um arquivo `serviceAccountKey.json` será gerado e baixado automaticamente para o seu computador.

# API Firebase

 **Firebase**


3miltalentos ▾

Visão geral do pr... 


**Configurações do projeto**

Gerar Cloud Messaging Integrações **Contas de serviço** Privacidade dos dados Usuários e permissões


[Gerenciar permissões da conta de serviço](#)

 **SDK Admin do Firebase**

Credenciais legadas

 **Secrets do Database**

Todas as contas de serviço

 [3 contas de serviço](#)

**SDK Admin do Firebase**

Sua conta de serviço do Firebase pode ser usada para autenticar vários recursos dessa plataforma (Database, Storage e Auth) programaticamente usando o SDK Admin unificado. [Saiba mais](#)

Conta de serviço do Firebase  
firebase-adminsdk-lon6f@miltalentos-d9445.iam.gserviceaccount.com


Snippet de configuração do SDK Admin

☒ Node.js ☐ Java ☐ Python ☐ Go

```
var admin = require("firebase-admin");

var serviceAccount = require("path/to/serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});
```



**Gerar nova chave privada**

Spark  
Sem custos financeiros US\$ 0/mês

Fazer upgrade

# API Firebase

## ► Arquivo serviceAccountKey.json


```
config > {} serviceAccountKey.json > ...
```



```
1  {
2    "type": "service_account",
3    "project_id": "miltalentos-d9445",
4    "private_key_id": "794d5f04ac6891e3ac9de8e719a6e480c5eaa665",
5    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAKggggSkAgEAAoIBAC
6    "client_email": "firebase-adminsdk-lon6f@miltalentos-d9445.iam.gserviceaccount.com",
7    "client_id": "106882397528506975777",
8    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9    "token_uri": "https://oauth2.googleapis.com/token",
10   "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11   "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-lon6f@miltalentos-d9445.iam.gserviceaccount.com",
12   "universe_domain": "googleapis.com"
13 }
14
```

# Firebase


## ► Ativar o serviço:

<https://console.cloud.google.com/apis/api/firestore.googleapis.com/metrics?project=3miltalentos-d9445>

 Comece seu teste gratuito com US\$ 300 em crédito. Não se preocupe, você não será cobrado se seus créditos acabarem. [Saiba mais](#)

[←](#) Detalhes do produto



## Google Cloud Firestore API

[Google Enterprise API](#)

Easily store and sync app data at global scale

[ATIVAR](#) [TESTAR ESTA API](#)

Clique para testar esta API no APIs Explorer

[VISÃO GERAL](#) [DOCUMENTAÇÃO](#) [PRODUTOS RELACIONADOS](#)

### Visão geral

Cloud Firestore is a NoSQL document database that simplifies storing, syncing, and querying data for your mobile and web apps at global scale. Its client libraries provide live synchronization and offline support, while its security features and integrations with the Firebase and Google Cloud platforms accelerate building truly serverless apps.

### Tutoriais e documentação

[Learn more](#)

### Mais detalhes

Tipo: [SaaS & APIs](#)

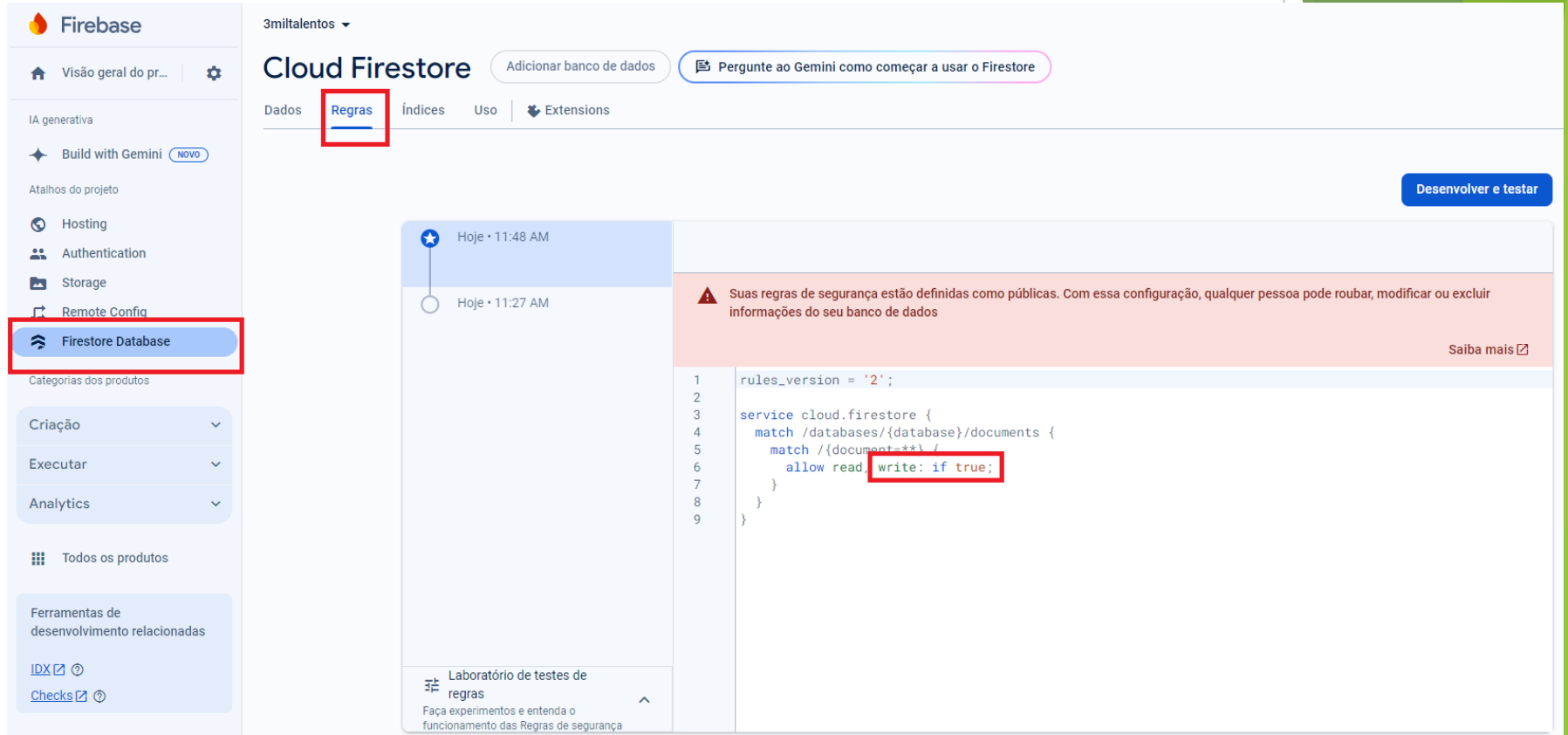
Última atualização do produto: 30/04/2022

Categoria: [Databases](#), [Google Enterprise APIs](#)

Nome do serviço: firestore.googleapis.com

# Firebase

## ► Permitir escrita no firestore database



The screenshot shows the Firebase Cloud Firestore console. In the left sidebar, the 'Firestore Database' option is highlighted with a red box. The main panel shows the 'Regras' (Rules) tab, also highlighted with a red box. A warning message at the top states: 'Suas regras de segurança estão definidas como públicas. Com essa configuração, qualquer pessoa pode roubar, modificar ou excluir informações do seu banco de dados.' Below this, the security rules are displayed in a code editor. The rule for 'write' is highlighted with a red box:

```
1 rules_version = '2';
2
3 service cloud.firestore {
4   match /databases/{database}/documents {
5     match /{document=**} {
6       allow read, write: if true;
7     }
8   }
9 }
```

At the bottom of the console, there is a section for 'Laboratório de testes de regras' (Rules Testing Lab) with a link to 'Faça experimentos e entenda o funcionamento das Regras de segurança'.

# APiFirebase - Criar API backend

- ▶ Criar a aplicação backend criando a pasta ApiFirebase na pasta de aplicações
- ▶ mkdir Apifirebase
- ▶ cd ApiFirebase

```
PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb> mkdir ApiFirebase

Directory: C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb

Mode                LastWriteTime         Length Name
----                -
d-----          05/08/2024    08:45             ApiFirebase

PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb> cd .\ApiFirebase\
PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase>
```

- ▶ Iniciar a aplicação npm init -y
- ▶ Instalar o firebase: npm install firebase-admin e npm install firebase

```
PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> npm install firebase-admin

added 156 packages, and audited 157 packages in 36s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

# APIFirebase

- ▶ Criar uma pasta config na raiz do projeto
- ▶ Dentro da pasta config criar o arquivo firebaseConfig.js
- ▶ Instalar npm `install -g firebase-tools`

```
PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> npm install -g firebase-tools  
  
added 630 packages in 2m  
  
69 packages are looking for funding  
run `npm fund` for details
```

# APIFirebase

- ▶ Instalar o dotenv: `npm install dotenv`

```
PS C:\Users\SenacRS\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> npm install dotenv  
  
added 1 package, and audited 197 packages in 2s  
  
14 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities
```

- ▶ Instalar o Express: `npm install express`

```
PS C:\Users\SenacRS\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> npm install express  
  
added 66 packages, and audited 263 packages in 5s  
  
25 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
PS C:\Users\SenacRS\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase>
```



# APIFirebase

- ▶ Instalar o Swagger:
  - ▶ npm install swagger-ui-express
  - ▶ npm install swagger-jsdoc

```
● PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> npm install swagger-jsdoc
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it.
  Consider using the https://github.com/isaacs/inflection module instead.
npm warn deprecated glob@7.1.6: Glob versions prior to v9 are no longer supported

added 30 packages, and audited 293 packages in 10s

26 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
● PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> npm install swagger-ui-express

added 2 packages, and audited 295 packages in 4s

26 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
○ PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\ApiFirebase> 
```

# APIFirebase

- Criar o arquivo index.js na raiz do projeto, escrever o código:

```
JS index.js > ...
1  const express = require('express');
2
3  const rotas = require('./rotas/fireBaseRotas');
4
5  //importa as configuracoes do arquivo de configuracao
6  require("dotenv").config();
7
8  const app = express();
9  const PORT = process.env.PORT;
10
11 //Importar o modulo Swagger
12 const setupSwagger = require('./swagger');
13
14 // Configurar Swagger
15 setupSwagger(app);
16
17 app.use(express.json());
18 app.use('/api', rotas);
19
20 app.listen(PORT, () => {
21   console.log(`Servidor rodando na porta ${PORT}`);
22 });
```

# APIFirebase

- Criar a pasta rotas e dentro desta pasta criar o arquivo fireBaseRotas.js escrevendo o código.

```
rotas > JS fireBaseRotas.js > ...
1 //Importa o modulo Express
2 const express = require('express');
3 const fireBaseController = require('../controller/fireBaseController');
4
5
6
7 //Cria o objeto rotas
8 const router = express.Router();
9
10 /**
11  * @swagger
12  * components:
13  *   schemas:
14  *     User:
15  *       type: object
16  *       required:
17  *         - id
18  *         - name
19  *         - email
20  *       properties:
21  *         id:
22  *           type: string
23  *           description: ID do usuário
24  *         name:
25  *           type: string
26  *           description: Nome do usuário
27  *         email:
28  *           type: string
29  *           description: Email do usuário
30  */
31
```

# APIFirebase

## ► Continuação código fireBaseRotas.js

```
31
32 /**
33  * @swagger
34  * tags:
35  *   name: Users
36  *   description: API para gerenciar usuários
37  */
38
39 /**
40  * @swagger
41  * /user:
42  *   post:
43  *     summary: Adiciona um novo usuário
44  *     tags: [Users]
45  *     requestBody:
46  *       required: true
47  *       content:
48  *         application/json:
49  *           schema:
50  *             $ref: '#/components/schemas/User'
51  *     responses:
52  *       200:
53  *         description: Usuário adicionado com sucesso
54  *       500:
55  *         description: Erro ao adicionar o usuário
56  */
57 router.post('/user', firebaseController.addUser);
58
```

```
61 /**
62  * @swagger
63  * /user/{id}:
64  *   get:
65  *     summary: Retorna um usuário pelo ID
66  *     tags: [Users]
67  *     parameters:
68  *       - in: path
69  *         name: id
70  *         required: true
71  *         schema:
72  *           type: integer
73  *         description: ID do usuário
74  *     responses:
75  *       200:
76  *         description: Dados do usuário
77  *         content:
78  *           application/json:
79  *             schema:
80  *               $ref: '#/components/schemas/User'
81  *       404:
82  *         description: Usuário não encontrado
83  *       500:
84  *         description: Erro ao buscar o usuário
85  */
86 router.get('/user/:id', firebaseController.getUser);
87
88 //exporta as rotas criadas
89 module.exports = router;
```

# APIFirebase

- Criar a pasta controller e dentro da pasta criar o arquivo fireBaseController.js, escrever o código.

```
controller > JS fireBaseController.js > getUser > getUser
1
2  const { db } = require('../config/firebaseConfig');
3
4  exports.addUser = async (req, res) => {
5    console.log('Adicionando user');
6    try {
7      const userRef = db.collection('users').doc(req.body.id);
8      await userRef.set({
9        name: req.body.name,
10       email: req.body.email
11     });
12
13     res.status(200).send('User added successfully');
14   } catch (error) {
15     res.status(500).send('Error adding user: ' + error.message);
16   }
17 };
18
19 exports.getUser = async (req, res) => {
20   try {
21     const userRef = db.collection('users').doc(req.params.id);
22     const doc = await userRef.get();
23     if (!doc.exists) {
24       res.status(404).send('No such user!');
25     } else {
26       res.status(200).json(doc.data());
27     }
28   } catch (error) {
29     res.status(500).send('Error getting user: ' + error.message);
30   }
31 };
32
```

# API Firebase

- ▶ Na pasta config criar o arquivo firebaseconfig.js, escrever o código.

```
config > JS firebaseConfig.js > 🔗 databaseURL
1
2  const path = require('path');
3  const admin = require('firebase-admin');
4  const serviceAccount = require(path.join(__dirname, 'serviceAccountKey.json'));
5
6  admin.initializeApp({
7    credential: admin.credential.cert(serviceAccount),
8    databaseURL: "https://miltalentos-d9445.firebaseio.com"
9  });
10
11  const db = admin.firestore(); // Para Firestore
12
13  module.exports = { admin, db };
```

- ▶ Dentro desta pasta salvar o arquivo gerado no [slide](#).
- ▶ Lembre-se de deixar o arquivo com o nome serviceAccountKey.json

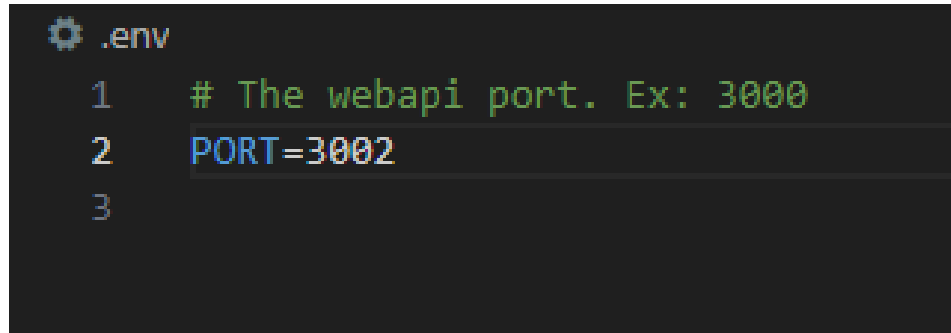
# APiFirebase

- Criar swagger.js, escrever o código abaixo

```
JS swagger.js > ...
1
2 //importa o modulo do swagger jsdoc
3 const swaggerJsdoc = require('swagger-jsdoc');
4
5 //importa o modulo com a interface grafica do swagger
6 const swaggerUi = require('swagger-ui-express');
7
8 const options = {
9   definition: {
10     openapi: '3.0.0',
11     info: {
12       title: 'APiFirebase',
13       version: '1.0.0',
14       description: 'Uma aplicacao conectando no firebase',
15     },
16     servers: [
17       {
18         url: 'http://localhost:3002/api',
19       },
20     ],
21   },
22   // Caminho para os arquivos de rotas
23   apis: ['./rotas/*.js'],
24 };
25
26 const swaggerSpec = swaggerJsdoc(options);
27
28 const setupSwagger = (app) => {
29   app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpec));
30 };
31
32 module.exports = setupSwagger;
```

# APIFirebase

- ▶ Criar o arquivo .env na raiz do projeto escrever o código abaixo

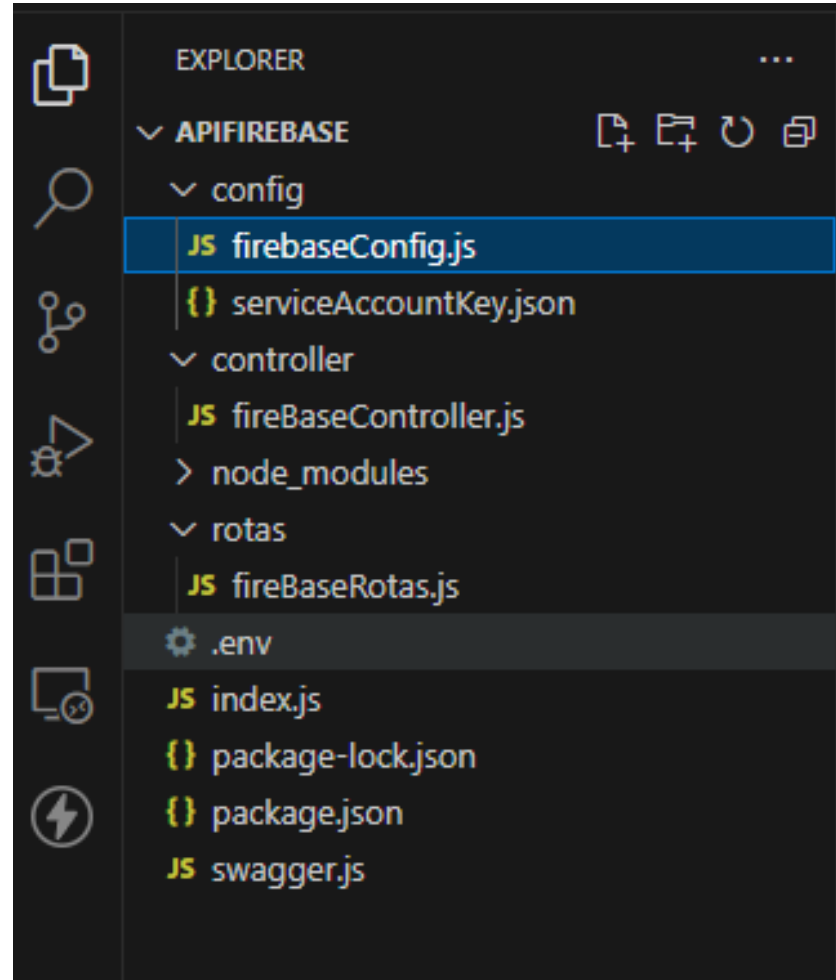
A screenshot of a code editor showing the contents of a .env file. The file is named ".env" and is located in the root of the project. It contains three lines of text: a comment, a port assignment, and an empty line.

```
1 # The webapi port. Ex: 3000
2 PORT=3002
3
```



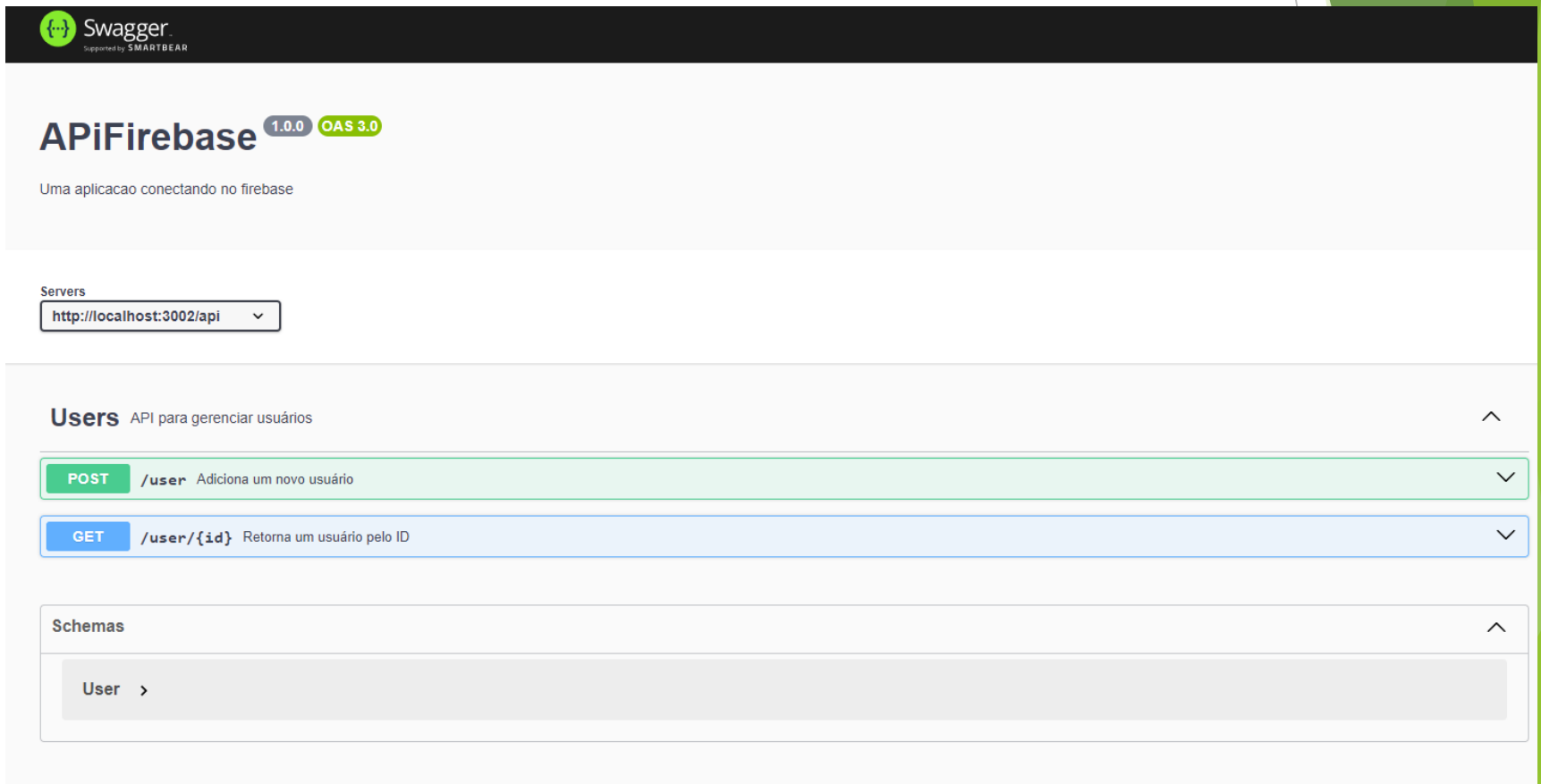
# APIFirebase

- Nossa aplicação ficará assim:



# Swagger

- ▶ Iniciar a aplicação: `npm start`
- ▶ Acessar o a URL: <http://localhost:3002/api-docs>
- ▶ Testar os métodos POST e GET usando o Swagger



The screenshot displays the Swagger UI for an API named 'APiFirebase'. At the top, the Swagger logo is visible, along with the text 'Supported by SMARTBEAR'. Below this, the API title 'APiFirebase' is shown with version tags '1.0.0' and 'OAS 3.0'. A description reads 'Uma aplicacao conectando no firebase'. A 'Servers' section contains a dropdown menu set to 'http://localhost:3002/api'. The main content area is titled 'Users' with the subtitle 'API para gerenciar usuários'. It lists two endpoints: a POST method for '/user' (Adiciona um novo usuário) and a GET method for '/user/{id}' (Retorna um usuário pelo ID). At the bottom, a 'Schemas' section shows a 'User' schema with a right-pointing arrow.

Swagger  
Supported by SMARTBEAR

**APiFirebase** 1.0.0 OAS 3.0

Uma aplicacao conectando no firebase

Servers

http://localhost:3002/api

**Users** API para gerenciar usuários

**POST** /user Adiciona um novo usuário

**GET** /user/{id} Retorna um usuário pelo ID

Schemas

User >

- ▶ Metodo POST, salvar user.

POST

/user

Adiciona um novo usuário

⌵

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{
  "id": "5",
  "name": "TESTE",
  "email": "TESTE"
}
```

⌵

⌵

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:3002/api/user' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "id": "5",
  "name": "TESTE",
  "email": "TESTE"
}'
```

Request URL

```
http://localhost:3002/api/user
```

Server response

Code

Details

200

Response body

```
User added successfully
```

⌵

Download

# Firebase

## Visualizar o registro salvo no Firebase

The screenshot displays the Firebase Cloud Firestore console interface. The left sidebar contains navigation options, with 'Firestore Database' highlighted. The main area shows the 'users' collection with a list of documents. Document '5' is selected, and its details are shown on the right, including fields for 'email' and 'name'.

**Left Sidebar:**

- Visão geral do pr... (highlighted)
- Cloud Firestore (highlighted)
- Build with Gemini (NOVO)
- Atalhos do projeto
- Hosting
- Authentication
- Storage
- Remote Config
- Firestore Database (highlighted)
- Categorias dos produtos
- Criação
- Executar
- Analytics
- Todos os produtos
- Ferramentas de desenvolvimento relacionadas
- IDX
- Checks

**Main Content:**

Proteja os recursos do Cloud Firestore de abusos, como fraude de faturamento ou phishing. Configurar o App Check

Visualização do painel Criador de consultas

users > 5

(default) users 5

+ Iniciar coleção

users

+ Adicionar documento

1

2

3

4

5

+ Iniciar coleção

+ Adicionar campo

email: "TESTE"

name: "TESTE"

# Swagger

► Metodo GET, buscar user.

**GET** /user/{id} Retorna um usuário pelo ID

Parameters

Name	Description
id * required integer (path)	ID do usuário

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:3002/api/user/5' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:3002/api/user/5
```

Server response

Code	Details
200	<div><div>Response body</div><pre>{   "name": "TESTE",   "email": "TESTE" }</pre><div><div>Download</div></div></div>