

# MIT em Desenvolvimento Full Stack

Front-end Jamstack com Gatsby

# Agenda

## Aula 8: Implantação e Publicação de uma Aplicação Gatsby.

- Dados Internos: Firestore.
- Firebase Authentication.
- Criar Conta.
- Considerações de Segurança.



# Dados Internos: Firestore

O **Cloud Firestore** é um banco de dados NoSQL **orientado a documentos**.

Ao contrário de um banco de dados SQL, não há tabelas nem linhas. Em vez disso, os dados são armazenados em documentos, que são organizados em coleções.

Cada documento contém um conjunto de pares chave-valor.

O **Cloud Firestore** é otimizado para armazenar grandes coleções de documentos pequenos.

```
■ alovelace  
first : "Ada"  
last : "Lovelace"  
born : 1815
```

Documento

```
■ usuários  
■ alovelace  
first : "Ada"  
last : "Lovelace"  
born : 1815  
  
■ aturing  
first : "Alan"  
last : "Turing"  
born : 1912
```

Coleção

```
■ salas  
■ salaA  
name : "my chat room"  
  
■ mensagens  
■ mensagem1  
from : "alex"  
msg : "Hello World!"  
  
■ mensagem2  
...  
  
■ salaB  
...
```

Subcoleção

1

Actors	
PK	ActorId
	ActorName

FilmActors	
PK,FK1	ActorId
PK,FK2	FilmId

2

Films	
PK	FilmId
	Title Category

Relacional

2

```
{
  "_id": 97, "Title": "BRIDE INTRIGUE",
  "Category": "Action",
  "Actors": [
    { "actorId": 65, "Name": "ANGELA HUDSON" }
  ]
}

{
  "_id": 115, "Title": "CAMPUS REMEMBER",
  "Category": "Action",
  "Actors": [
    {
      "actorId": 8, "Name": "MATTHEW JOHANSSON" },
      { "actorId": 45, "Name": "REESE KILMER" },
      { "actorId": 168, "Name": "WILL WILSON" }
    ]
}

{
  "_id": 105, "Title": "BULL SHAWSHANK",
  "Category": "Action",
  "Actors": [
    { "actorId": 2, "Name": "NICK WAHLBERG" },
    { "actorId": 23, "Name": "SANDRA KILMER" }
  ]
}
```

1

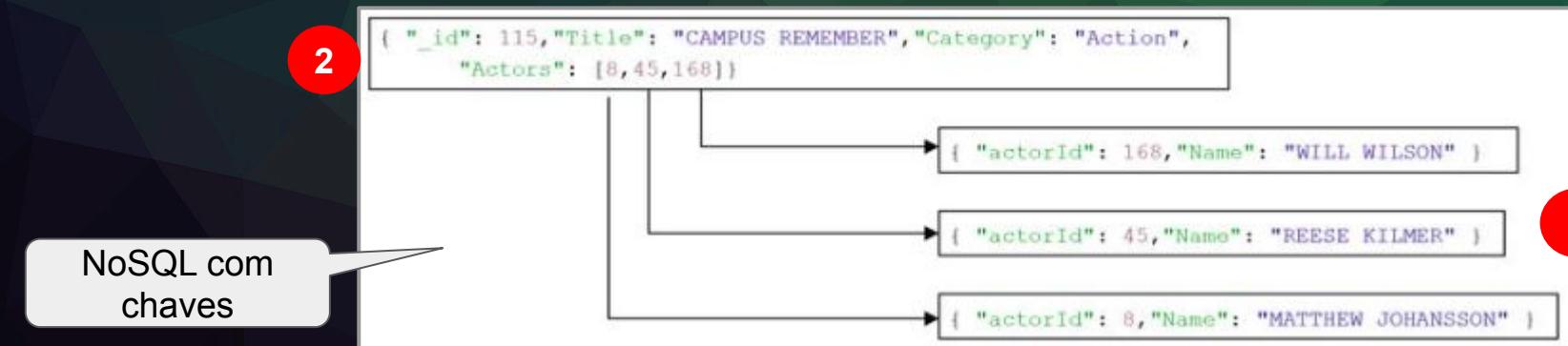
NoSQL com subcoleções

Film Document

Actor document

Collection

Array of actors



# Olá, Armênio

Este é o Firebase!

Vamos começar



## Criar um novo projeto do Firebase

Integre produtos do Firebase para turbinar seu app

1



## Começar a programar um app

Criar um app com um dos modelos do Firebase Studio

Testar um app de exemplo



## Criar um app do Flutter com tecnologia de IA

Implante um app de exemplo que mostre como a API Gemini Live, os comandos multimodais e a criação de imagens com o Nano Banana funcionam no Flutter



## Testar um app agêntico de barista

Implante um app de exemplo que use o Firestore, o Firebase Authentication e a chamada de função no Firebase AI Logic. Depois disso, estude o código no Firebase Studio.



Pesquisar todos os projetos e espaços de trabalho

Projetos e espaços de trabalho ▾



## INFNET-JAVA-AT

infnet-java-at



## React 2024

react-2024-b96bd



## Softwares Escalaveis

softwares-escalaveis



## Estudos de Jamstack

estudos-de-jamstack-2b3a1



## React

react-139c2



1 a 5 de 5



- × Criar um projeto

Vamos começar nomeando o projeto 

2

Nome do projeto  
**Gatsby**

 gatsby-ee4fa

 prof.infnet.edu.br

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

Continuar

- × Criar um projeto

3

## Google Analytics para seu projeto do Firebase

O Google Analytics é uma solução de análise ilimitada e gratuita. 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
- × Gatilhos do Cloud Functions com base  em eventos
- × Geração de relatórios ilimitada gratuita 

Ativar o Google Analytics neste projeto  
Recomendado

Anterior

Criar projeto

4

Visão geral do projeto

O que há de novo

AI Logic **NOVO**

Categorias dos produtos

Criação

Executar

Analytics

AI

Gatsby **Plano Spark**

Teste um app agêntico de barista com a tecnologia do Firebase AI Logic! [Testar app de exemplo](#) [X](#)

+ Adicionar app

## Olá, Armênio

Este é seu projeto do Firebase!

O Gemini pode dar sugestões de serviços e soluções do Firebase para seu app

5

## Configurações do projeto

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

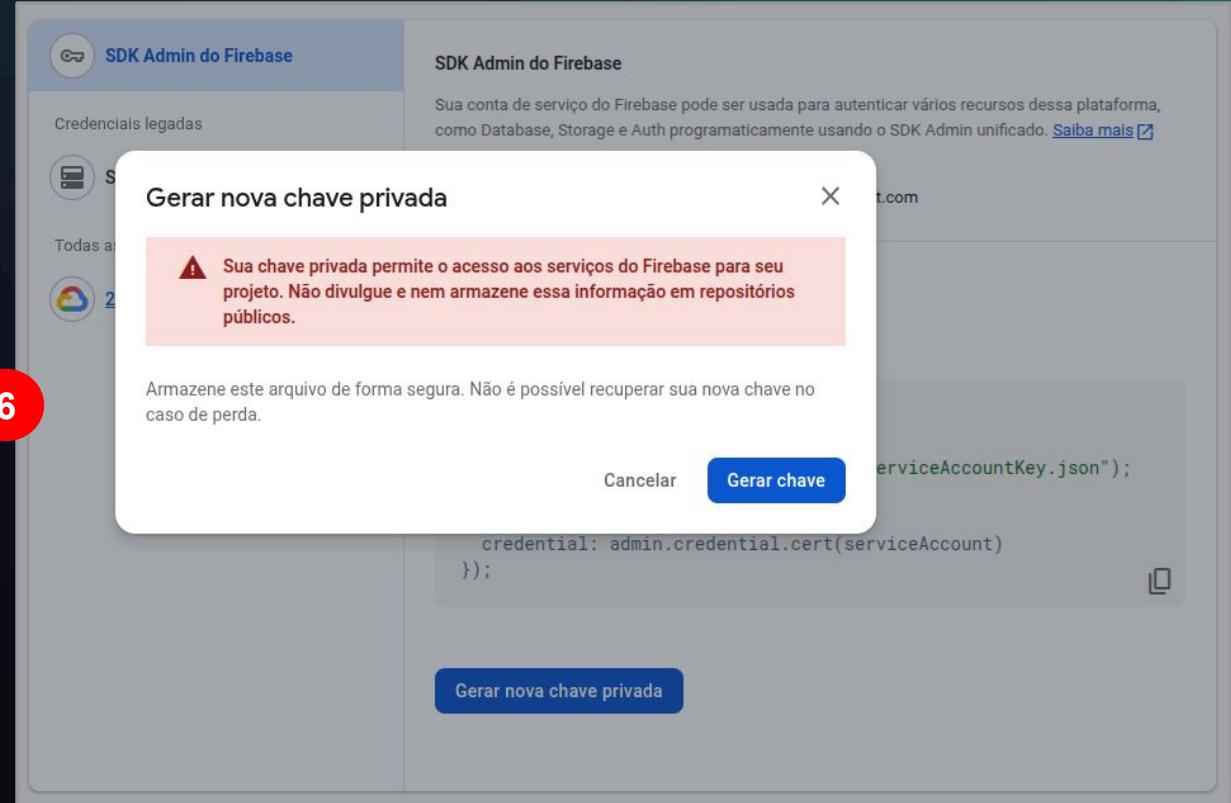
### Seu projeto

Nome do projeto	Gatsby
ID do projeto	gatsby-ee4fa
Número do projeto	885521705987
Empresa-mãe/pasta no GCP	prof.infnet.edu.br
Chave de API da Web	Nenhuma chave de API da Web para este projeto

### Ambiente

Esta configuração personaliza o projeto para diferentes fases do ciclo de vida do app

Tipo de ambiente	Não especificado
------------------	------------------



```
project/
  └── firebase-key.json
  └── gatsby-config.js
  └── gatsby-node.js
  └── package.json
  └── src/
      └── pages/
          └── users.js
```

**7**

**Cloud Firestore**

Atualizações em tempo real, consultas eficientes, escalonamento automático e compatibilidade com o MongoDB

**Criar banco de dados** Pedir ao Gemini

**Criar um banco de dados**

- Selecionar a edição**
  - Edição Standard  
Mecanismo de consulta simples com indexação automática. Para documentos de até 1 MB.
  - Edição Enterprise  
Advanced query engine with MongoDB compatibility. For documents up to 4 MB. Supports MongoDB Drivers and Tools only.

Não sabe qual é a melhor edição para você? Compare as edições [Compare as edições](#)

**Avançar**
- ID e local do banco de dados**
- Configurar**

**Criar um banco de dados**

- Selecionar a edição**
- ID e local do banco de dados**

ID do banco de dados  
(default)

Local  
southamerica-east1 (São Paulo)

O sua configuração de localização é onde os dados do Cloud Firestore serão armazenados

**1** Não será possível mudar o local depois de terminar a configuração dele. [Saiba mais](#)

**Avançar**
- Configurar**

**Criar um banco de dados**

- Selecionar a edição**
- ID e local do banco de dados**
- Configurar**

Após definir a estrutura de dados, será necessário criar regras para proteger suas informações. [Saiba mais](#)

**Iniciar no modo de produção**

Seus dados são particulares por padrão. O acesso de leitura/gravação do cliente será concedido apenas se especificado por suas regras de segurança.

**Iniciar no modo de teste**

Por padrão, seus dados estão definidos para permitir uma configuração rápida. No entanto, você precisa atualizar suas regras de segurança em até 30 dias para permitir o acesso de leitura/gravação do cliente em longo prazo.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document}** {
      allow read, write: if false;
    }
  }
}
```

**10**

**Todas as leituras e gravações de terceiros serão negadas**

**Criar**

**8**

**9**

Gatsby ▾

11

Cloud Firestore >

## Banco de dados

Adicionar banco de dados

◆ Pergunte ao Gemini como começar a usar o Firestore

Dados

Regras

Índices

Recuperação de desastres

Uso

Extensions



Proteja os recursos do Cloud Firestore de abusos, como fraude de faturamento ou phishing

Configurar o App Check



Visualização do painel

Criador de consultas



usuarios > hhTFpwdz4QBB...

Mais no Google Cloud



(default)

usuarios



hhTFpwdz4QBB45W1eLio



+ Iniciar coleção

+ Adicionar documento

+ Iniciar coleção

usuarios



hhTFpwdz4QBB45W1eLio



+ Adicionar campo

email: "machado@abl.org.br"

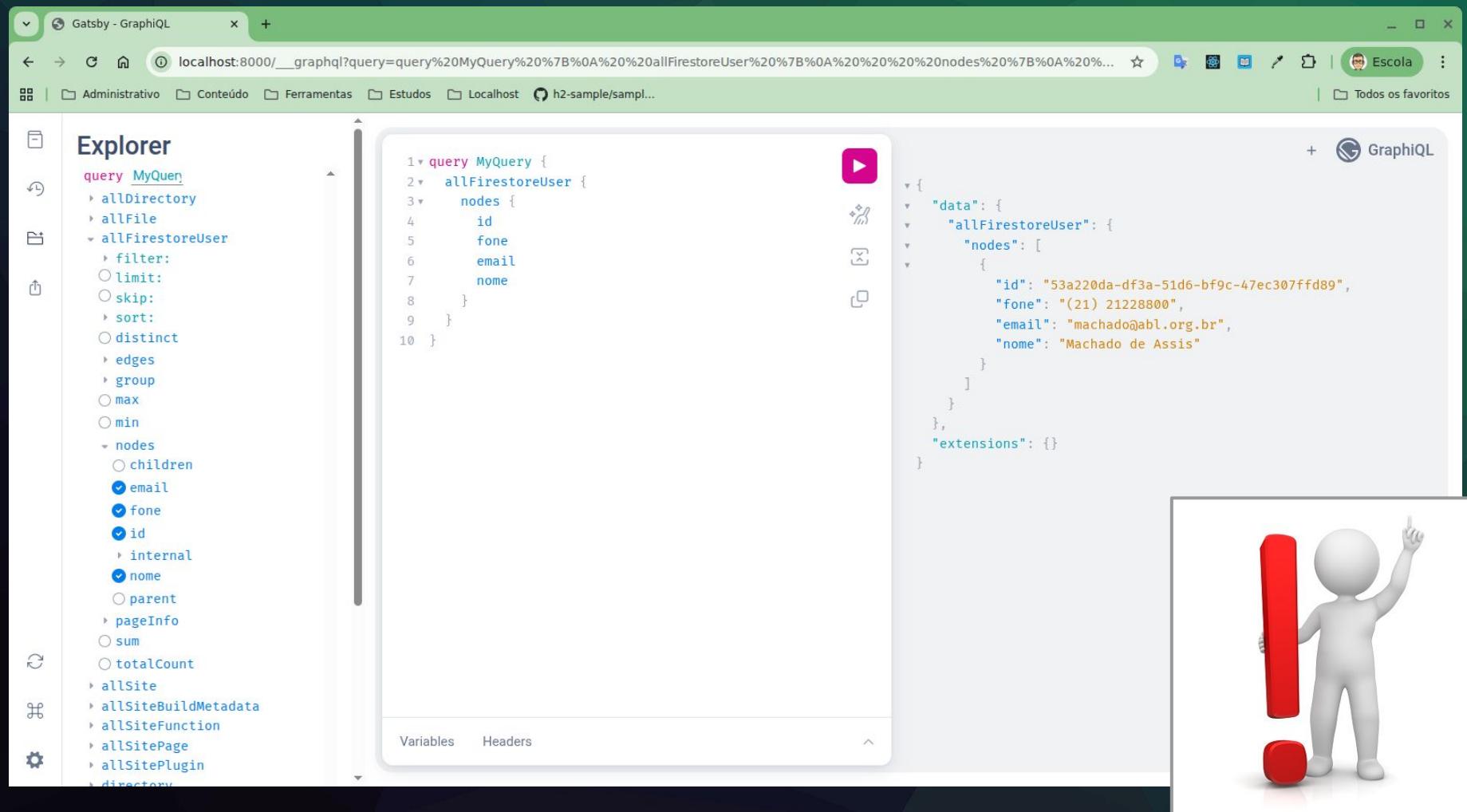
fone: "(21) 21228800"

nome: "Machado de Assis"

12

```
JS gatsby-node.js ✘ X
JS gatsby-node.js > ...
1  const admin = require("firebase-admin");
2  const path = require("path");
3
4  exports.sourceNodes = async ({ actions, createNodeId, createContentDigest }) => {
5    const { createNode } = actions;
6
7    if (!admin.apps.length) {
8      admin.initializeApp({
9        credential: admin.credential.cert(require("./firebase-key.json")),
10      });
11    }
12
13    const db = admin.firestore();
14
15    const snapshot = await db.collection("usuarios").get();
16
17    snapshot.forEach(doc => {
18      const data = doc.data();
19
20      createNode({
21        ...data,
22        id: createNodeId(`firebase-user-${doc.id}`),
23        parent: null,
24        children: [],
25        internal: {
26          type: "FirestoreUser",
27          contentDigest: createContentDigest(data),
28        },
29      });
30    });
31  };

```



13

```
js index.js M X
src > pages > js index.js > ...
1 import React from "react";
2 import { graphql } from "gatsby";
3
4 export const query = graphql`  

5   query {
6     allFirestoreUser {
7       nodes {
8         id
9         fone
10        email
11        nome
12      }
13    }
14  `;
15
```

14

14

```
js index.js M X
src > pages > js index.js > ...
17  const IndexPage = ({ data }) => {
18    const users = data.allFirestoreUser.nodes;
19
20    return (
21      <main style={{ padding: "2rem" }}>
22        <h1>Usuários do Firestore</h1>
23        <ul>
24          {users.map((user) => (
25            <li key={user.email}>
26              <strong>{user.nome}</strong> - {user.fone}
27            </li>
28          )));
29        </ul>
30      </main>
31    );
32  };
33
34  export default IndexPage;
35
36  export const Head = () => <title>Home Page</title>;
```

# Firebase Authentication

1

Firebase

Visão geral do projeto

Atalhos de projetos

Firestore Database

Authentication

O que há de novo

AI Logic NOVO

Categorias dos produtos

Criação

Executar

Analytics

AI

Gatsby ▾

# Authentication

Autenticar e gerenciar usuários a partir de uma variedade de provedores sem código do lado do servidor

Vamos começar

◆ Pedir ao Gemini

The screenshot shows the Firebase console's Authentication interface. A red circle with the number '1' highlights the 'Authentication' tab in the sidebar. The main area displays the title 'Authentication' and a brief description about managing users. Below the description are two buttons: 'Vamos começar' (Let's start) and 'Pedir ao Gemini' (Ask Gemini). To the right is a stylized illustration of a yellow badge with a person's photo, a padlock, and a gear.

2

Gatsby ▾

# Authentication

Usuários

Método de login

Modelos

Uso

Configurações

Extensions

### Provedores de login

Comece a usar o Firebase Auth ao adicionar seu primeiro método de login

Provedores nativos	Outros provedores	Provedores personalizados
E-mail/senha	Google, Facebook, Play Games, Smartphones, Game Center, Apple, GitHub, Microsoft, Twitter, Yahoo, Anônimo	OpenID Connect, SAML

The screenshot shows the 'Authentication' settings page. A red circle with the number '2' highlights the 'Método de login' tab. The page lists various authentication providers under three categories: 'Provedores nativos' (Email/password, Smartphone, Anônimo), 'Outros provedores' (Google, Facebook, Play Games, Game Center, Apple, GitHub, Microsoft, Twitter, Yahoo), and 'Provedores personalizados' (OpenID Connect, SAML).

# Authentication

Usuários   Método de login   Modelos   Uso   Configurações   🌐 Extensions

ⓘ Os seguintes recursos de autenticação vão parar de funcionar quando os Firebase Dynamic Links forem desativados em breve:  
autenticação por link de e-mail em apps para dispositivos móveis e suporte do Cordova ao OAuth em apps da Web.

Adicionar usuário



Identificador

Provedores

Criação ↓

Último login

UID do usuário

3

Adicionar um usuário com e-mail/senha

E-mail

machado@abl.org.br

Senha

capitu

Cancelar

Adicionar usuário

Nenhum usuário para este projeto ainda

4

Gatsby

Visão geral do projeto

Atalhos de projetos

Firestore Database

Authentication

O que há de novo

AI Logic (NOVO)

Categorias dos produtos

Criação

Gatsby (Plano Spark)

iOS+ Selecionar uma plataforma

Olá, Armênio

Este é seu projeto do Firebase!

5

### Adicionar o Firebase ao seu app da Web

1 Registrar app

Apelido do app (Gatsby)

Configure também o **Firebase Hosting** para este app. [Saiba mais](#)

O Hosting pode ser configurado a qualquer momento. Faça isso quando quiser sem custos financeiros.

**Registrar app**

2 Adicionar o SDK do Firebase

6

### Adicionar o Firebase ao seu app da Web

1 Registrar app

2 Adicionar o SDK do Firebase

Usar o npm  Usar a tag <script>

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

```
S npm install firebase
```

Depois inicialize 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: "AIzaSyC0aQajEy0MAH1xx05Inks795tUwI4eto",
  authDomain: "gatsby-ee4fa.firebaseioapp.com",
  projectId: "gatsby-ee4fa",
  storageBucket: "gatsby-ee4fa.firebaseiostorage.app",
  messagingSenderId: "885521705987",
  appId: "1:885521705987:web:8c5528db259199c7423236"
};

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

7

```
js firebase.js X
```

```
src > infra > js firebase.js > ...
```

```
1 import { initializeApp } from "firebase/app";
2
3 const firebaseConfig = {
4   apiKey: "AIzaSyBn-Ec",
5   authDomain: "react-1",
6   projectId: "react-13",
7   storageBucket: "rea",
8   messagingSenderId: "103699173716",
9   appId: "1:3699173716"
10 };
11
12 const app = initializeApp(firebaseConfig);
13 export const auth = getAuth(app);
14 export default app;
```

8

```
❸ Tel1.jsx ✘ x
src > pages > ⚡ Tel1.jsx > ...
1  export default function Tel1() {
2
3    return (
4      <div className="container">
5        <h3>Login</h3>
6        <form>
7          <label htmlFor="usuario">Email:</label>
8          <br />
9          <input type="text" id="email"  />
10         <br />
11         <label htmlFor="senha">Senha:</label>
12         <br />
13         <input type="password" id="senha"  />
14         <br /><br />
15         <input type="button" value="Login"/>
16       </form>
17     </div>
18   )
19 }
```

9

```
JS usuarios.js U X

src > infra > JS usuarios.js > logarUsuario
1 import { signInWithEmailAndPassword } from 'firebase/auth';
2 import { auth } from '../infra.firebaseio';
3
4
5 export async function logarUsuario(email, senha) {
6     let retorno = new Object();
7     await signInWithEmailAndPassword(auth, email, senha)
8         .then((credenciais) => {
9             console.log(credenciais);
10            retorno.id = credenciais.user.uid;
11            retorno.email = email;
12            retorno.senha = senha;
13        })
14        .catch((error) => {
15            console.log(` ${error.code} = ${error.message}`);
16            retorno.erro = "Login Inválido";
17        });
18    return retorno;
19 }
```

10

```
❷ Tela1.jsx ✘ ×  
src > pages > ❸ Tela1.jsx > ...  
1 import { logarUsuario } from "../infra/usuarios"  
2  
3 export default function Tela1() {  
4  
5     async function handleLogin(event) {  
6         const email = document.getElementById("email").value;  
7         const senha = document.getElementById("senha").value;  
8         let usuario = await logarUsuario(email, senha);  
9         if(usuario.id) {  
10             alert(`Login Efetuado com Sucesso - id: ${usuario.id}`);  
11         } else {  
12             alert(usuario.erro);  
13         }  
14     }  
15 }
```

React App

localhost:3000

Administrativo Conteúdo Ferramentas

Todos os favoritos

Reiniciar para atualizar

localhost:3000 diz

Login Efetuado com Sucesso - id:  
mNjlr3lChHTkYX61e4zhVyl34ml2

OK

Console

top | Filtro

odos os níveis | 2 problemas: 1 1

usuarios.js:9  
UserCredentialImpl {user: UserImpl, pro  
viderId: null, \_tokenResponse: {}, ope  
rationType: 'signIn'}



# Criar Conta

JS usuarios.js U X

```
src > infra > JS usuarios.js > ...
1  import { createUserWithEmailAndPassword, signInWithEmailAndPassword } from 'firebase/auth';
2  import { auth } from '../infra.firebaseio';
3
4
5 > export async function logarUsuario(email, senha) {
19    }
20
21  export async function criarConta(email, senha) {
22      let retorno = new Object();
23      await createUserWithEmailAndPassword(auth, email, senha)
24          .then((credenciais) => {
25              console.log(credenciais);
26              retorno.id = credenciais.user.uid;
27              retorno.email = email;
28              retorno.senha = senha;
29          })
30          .catch((error) => {
31              console.log(` ${error.code} = ${error.message}`);
32              retorno.erro = "Login Inválido";
33          });
34 }
```

1

2 Tela2.jsx ✘

```
src > pages > Tela2.jsx > ...
1  export default function Tela2() {
2
3      return (
4          <div className="container">
5              <h3>Nova Conta</h3>
6              <form>
7                  <label htmlFor="email">Email:</label><br><br>
8                  <input type="text" id="novoEmail" />
9                  <br />
10                 <label htmlFor="senha">Senha:</label><br><br>
11                 <input type="password" id="novaSenha" />
12                 <br />
13                 <label htmlFor="confirma">Confirma:</label><br><br>
14                 <input type="password" id="confirma" />
15                 <br /><br />
16                 <input type="button" value="Criar Conta" />
17             </form>
18         </div>
19     )
20 }
```

3

```
❶ Tela2.jsx ✘ ❷ usuarios.js ✘
src > pages > ❸ Tela2.jsx > ...
1  import { criarConta } from "../infra/usuarios";
2
3  export default function Tela2() {
4
5      async function handleCriarConta(event) {
6          const email = document.getElementById("novoEmail").value;
7          const senha = document.getElementById("novaSenha").value;
8          const confirma = document.getElementById("confirma").value;
9          if (senha === confirma) {
10              let usuario = await criarConta(email, senha);
11              if (usuario.id) {
12                  alert(`Conta Criada com Sucesso - id: ${usuario.id}`);
13              } else {
14                  alert(usuario.erro);
15              }
16          }
17      }
}
```

React - Authentication - Cor ✎ +

console.firebaseio.google.com/project/react-139c2/authentication/users?hl=pt-br&fb\_uti... ☆

Administrativo Conteúdo Ferramentas Livros | Todos os favoritos

Reiniciar para atualizar :

## Firebase

Visão geral do p... 🚪

Atalhos do projeto

Authentication

Firestore Database

Categorias dos produtos

Criação

Liberar e monitorar

Analytics

Engajamento

Spark  
Sem custos financeiros US\$ 0/mês

React ▾

# Authentication

Usuários Método de login Modelos Uso Configurações Extensions

Pesquise por endereço de e-mail, número de telefone ou UID do usuário

Adicionar usuário

Identificador	Provedores	Data de criação	Último login	UID do usuário
clarice@yahoo.com.br	✉️	9 de mai. de 2024	9 de mai. de 2024	aKS0CghO8LTquh6nEEgcCDY...
rachel@gmail.com	✉️	9 de mai. de 2024	9 de mai. de 2024	o5md7npuhROZe2zStqFYaCR...
machado@abl.org.br	✉️	6 de fev. de 2024	9 de mai. de 2024	mNjlr3lChHTkYX61e4zhVyl34...

Fazer upgrade

Linhas por página: 50 1 – 3 of 3

React App

localhost:3000

Administrativo Conteúdo Ferramentas

Todos os favoritos

Reiniciar para atualizar

localhost:3000 diz

Conta Criada com Sucesso - id:  
aKS0CghO8LTquh6nEEgcCDYxEMG2

OK

Console

top Filtro

odos os níveis 2 problemas: 1 1

usuarios.js:25

UserCredentialImpl {user: UserImpl, pro  
viderId: null, \_tokenResponse: {}, ope  
rationType: 'signIn'}



# Considerações de Segurança

Camada	Acesso	Implementação	Observações
Build-time SSG	Seguro	firebase-admin no gatsby-node.js	Usa Service Account → acesso total, mas roda apenas no servidor durante build
Runtime SPA	Restrito	firebase/app + firebase/auth + firebase/firestore	Usa config pública do app, mas regras do Firestore protegem os dados
Segurança	Garantida	Firestore Security Rules + Auth	Define quem pode ler/escrever