



React-Native - Integração com Firebase

Mobile App Development



Agenda

- O que é o Firebase?
- Recursos disponíveis
- Integração com o React-Native
- Autenticação
- Leitura complementar
- Exercícios



Agenda

- **O que é o Firebase?**
- Recursos disponíveis
- Integração com o React-Native
- Autenticação
- Leitura complementar
- Exercícios

O que é o Firebase?

O **Firebase** é um serviço originado a partir de uma Startup em **2011** e que foi adquirida pela **Google** em **2014**.

Como produto, fornecia inicialmente um produto chamado **Firebase Real-time Database**, no qual **sincronizava dados de aplicativos mobile e web** de forma simples na cloud.

O que é o Firebase?

Isso permitia a **abstração da lógica** de sincronizar os dados, incluindo os momentos em que um **APP estivesse offline e entre plataformas diferentes**.

Outro recurso adicionado posteriormente ao Firebase foi a abstração de logins sociais / login e-mail + senha / login via número de telefone conhecido como **Firebase Authentication**.



Firebase

O que é o Firebase?

Após a aquisição realizada pela **Google**, o **Firebase** se tornou um dos principais **Backend as a Service (bass)** para aplicativos móveis, também chamado de “mbass”.

Outros recursos foram adicionados após isto, como o **Firebase Analytics**.

A página oficial do Firebase é:

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

Vantagens

- Implementação muito simples;
- Documentação bem completa;
- Suporte para iOS, Android, WEB e Unity;
- Agiliza a produção de aplicativos;
- O desenvolvedor não precisa se preocupar com todo o Back-end;
- Desenvolvedor consegue focar mais tempo apenas na regra de negócio;
- Possui um bom sistema de segurança baseado em regras de acesso;



Agenda

- O que é o Firebase?
- **Recursos disponíveis**
- Integração com o React-Native
- Autenticação
- Leitura complementar
- Exercícios

Recursos disponíveis

Os principais recursos que temos disponíveis no Firebase são:

Cloud Firestore:

- O mais recente banco de dados em cloud com sincronização em tempo real e mais recursos que o seu antecessor: Realtime Database



Firebase

Recursos disponíveis

Realtime Database:

- O primeiro banco de dados em cloud com sincronização em tempo real disponibilizado pelo Firebase.

Autenticação:

- Gerenciamento de usuários e com diversos métodos de autenticação, incluindo autenticação pelas principais redes sociais e via número de telefone.



Firebase

Recursos disponíveis

Cloud Storage

- Armazenamento de arquivos gerados pelo usuário integrado com todo o sistema de segurança fornecido pelo Firebase;

Cloud Functions

- Códigos personalizados e escaláveis para serem executados pelo Back-end quando algum evento acontecer, como por exemplo, após o cadastro de um novo usuário no Firebase.



Firebase

Recursos disponíveis

Crashlytics

- Relatório de erros com alertas em tempo real, facilitando assim encontrar o que causa algum determinado bug dentro de um APP.

TestLab

- Ferramenta que permite testes automatizados de navegação e uso do aplicativo em dispositivos reais e virtuais com as mais variadas configurações.



Firebase

Recursos disponíveis

Google Analytics

- Análise do comportamento do usuário, da mesma forma que temos para a Web.

Cloud Messaging

- Envio de push notifications de forma gratuita e segmentada por usuários.

Remote Config

- Personalização da exibição do aplicativo para o usuário sem a necessidade de disponibilizar uma nova versão do APP.



Firebase



Agenda

- O que é o Firebase?
- Recursos disponíveis
- **Integração com o React-Native**
- Autenticação
- Leitura complementar
- Exercícios

Integração com React-Native

Vamos começar a integração do **Firebase** com o **React-Native** criando um novo projeto com o comando:

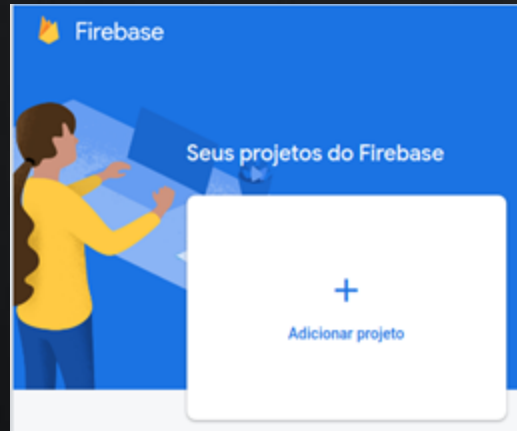
```
npx react-native init nome_do_seu_projeto_aqui
```

Integração com React-Native

Após a criação do projeto, abra o console do **Firebase** acessando a URL:

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

Vamos criar um novo projeto no **Firebase**, clicando no seguinte link:



Integração com React-Native

Na página a seguir, insira um nome para o projeto sem acentuação.

✕ Criar um projeto(Passo 1 de 3)

Vamos começar com um
nome para o projeto®

Insira o nome do projeto

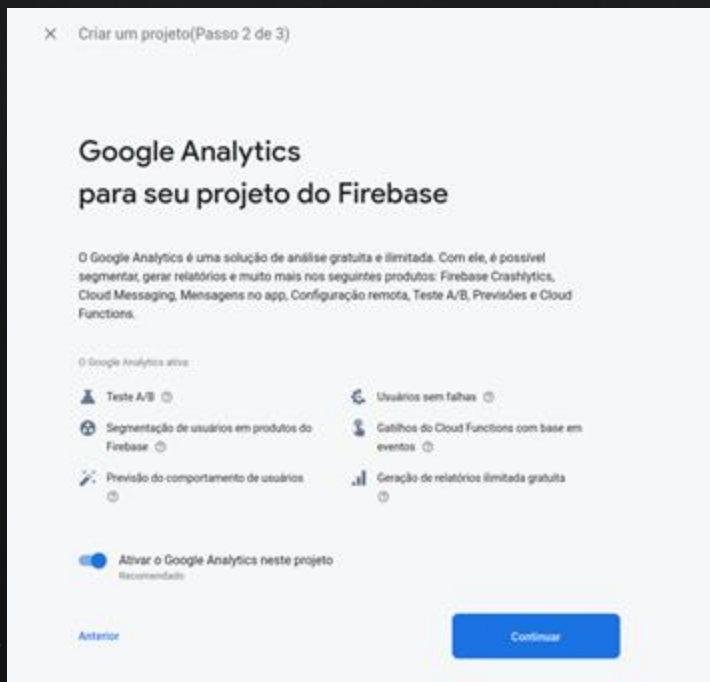
O nome do projeto é obrigatório

my-awesome-project-123

Continuar

Integração com React-Native

Ative o Google Analytics para seu aplicativo caso queira. Alguns dos recursos do Firebase dependem dessa funcionalidade habilitada:



Integração com React-Native

Crie uma conta do Google Analytics para este Aplicativo e clique em “**Criar projeto**”:

× Criar um projeto(Passo 3 de 3)

Configurar o Google Analytics

Novo nome da conta do Google Analytics

O nome da conta é obrigatório.

Cancelar

Salvar

Após a criação do projeto, uma nova propriedade do Google Analytics será criada na conta selecionada 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 Analytics. [Saiba mais](#)

Anterior

Criar projeto

Integração com React-Native

Após a criação do projeto você será redirecionado para a tela onde são exibidos todos os serviços disponíveis pelo **Firebase** no menu lateral.



Integração com React-Native

Clique em adicionar aplicativo para Android:

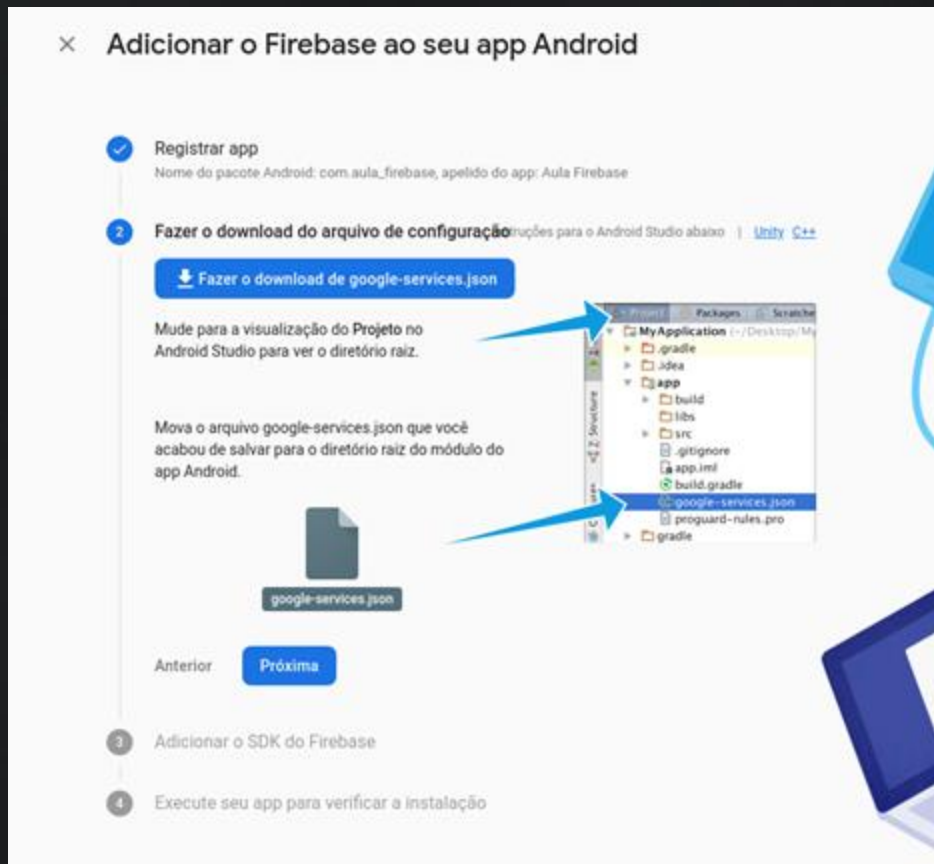


do seu aplicativo no

•

<https://developers.google.com/android/guides/client-auth>

Faça o download do arquivo **google-services.json** para dentro do diretório **android/app** do seu projeto React-Native.



Integração com React-Native

Siga os procedimentos descritos na etapa 03 do processo, conforme a imagem ao lado dentro dos arquivos do gradle da pasta android do projeto em React-Native:

3

Adicionar o SDK do Firebase

Instruções para Gradle | [Unity](#) [C++](#)

O plug-in dos serviços do Google para [Gradle](#) carrega o arquivo google-services.json cujo download você acabou de fazer. Modifique seus arquivos build.gradle para usar o plug-in.

build.gradle no nível do projeto (<project>/build.gradle):

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.3'
    }
}

allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    ...
}
```

build.gradle no nível do app (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // add the Firebase SDK for Google Analytics
    implementation 'com.google.firebase:firebase-analytics:17.5.0'
    // add SDKs for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```


Integração com React-Native

Para a integração com o React-Native vamos utilizar um plugin conhecido:

- **React Native Firebase:**

<https://rnfirebase.io/>

OBS: Como a instalação e configuração pode sofrer atualizações, vamos sempre seguir o passo a passo descrito na página do projeto.

Realize a instalação no projeto conforme a documentação.

Integração com React-Native

Habilite na aplicação o suporte ao Multidex conforme a documentação:

<https://rnfirebase.io/enabling-multidex>

Integração com React-Native

Execute a aplicação no emulador através do comando:

```
npx react-native run-android
```

Se todos os passos anteriores foram executados corretamente, então a aplicação irá compilar com sucesso!



Agenda

- O que é o Firebase?
- Recursos disponíveis
- Integração com o React-Native
- **Autenticação**
- Leitura complementar
- Exercícios

Autenticação

O plugin **React-Native-Firebase** é modular, ou seja, permite adicionarmos apenas as funcionalidades do Firebase que queremos utilizar.

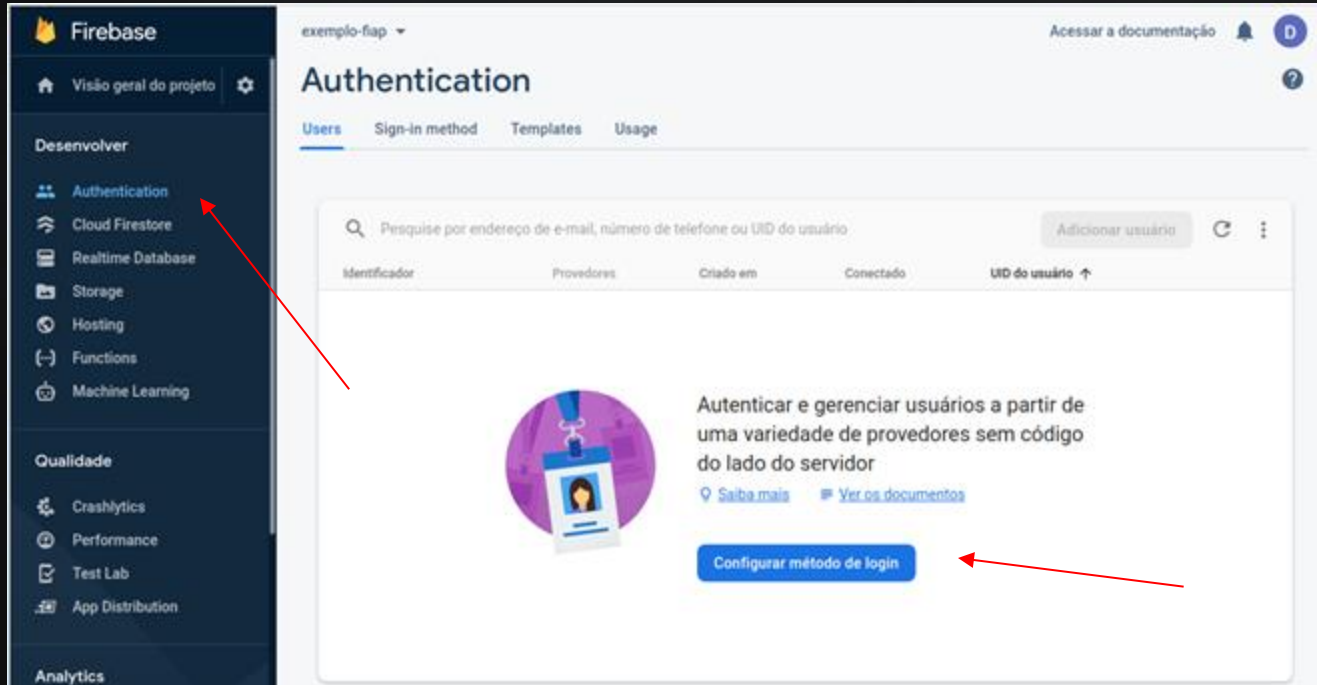
A primeira funcionalidade do **Firebase** que vamos explorar é a **autenticação**.

Siga as orientações de instalação em:

<https://rnfirebase.io/auth/usage>

Autenticação

No console do Firebase clique em Authentication e vamos configurar as formas de login que queremos em nosso App.



Autenticação

Ative as opções:

- E-mail/senha;
- Anônimo.

exemplo-fiap ▾ Authentication Acessar a documentação

Users Sign-in method Templates Usage

Provedores de login

Provedor	Status
E-mail/senha	<div><div></div>Ativar</div> <p>Permite que os usuários se inscrevam usando o endereço de e-mail e a senha deles. Nossos SDKs também oferecem verificação de endereço de e-mail, recuperação de senha e primitivos de alteração de endereço de e-mail. Saiba mais</p> <div><div></div>Ativar</div> <p>Link do e-mail (login sem senha)</p> <div>Cancelar Salvar</div>
Smartphone	Desativado
Google	Desativado
Play Games	Desativado
Game Center Beta	Desativado
Facebook	Desativado
Twitter	Desativado
GitHub	Desativado
Yahoo	Desativado
Microsoft	Desativado

Firebase

Visão geral do projeto

Desenvolver

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualidade

- Crashlytics
- Performance
- Test Lab
- App Distribution

Analytics

- Dashboard
- Events
- Conversions
- Audiences

Extensions

Spark
Plano gratuito US\$ 0/mês Fazer upgrade

Autenticação

JS App.js > ...

```
1  import React, {
2    useState,
3    useEffect
4  } from 'react';
5
6  import {
7    Button,
8    Text,
9    View
10 } from 'react-native';
11
12 import auth from '@react-native-firebase/auth';
13
14 const App = (props) => {
15
16
17 }
18
19 export default App;
20
```

Crie o seguinte código no arquivo App.js do seu projeto.

Autenticação

Continue a alteração adicionando dentro do componente App as seguintes lógicas:

```
14 const App = (props) => {
15
16   const [initializing, setInitializing] = useState(true);
17   const [user, setUser] = useState();
18
19   if (initializing) {
20     return null;
21   }
22
23   if (!user) {
24     return (
25       <View>
26         <Button
27           onPress={ () => auth().signInAnonymously().then((user) => setUser(user)) }
28           title="Acessar" />
29       </View>
30     );
31   }
32
33   return (
34     <View>
35       <Text>Olá usuário!</Text>
36
37       <Button
38         onPress={ () => auth().signOut() }
39         title="Sair" />
40     </View>
41   );
42 }
43
44 export default App;
45
```











Autenticação

Adicione após a declaração dos estados o trecho selecionado:

```
14  const App = (props) => {  
15  
16    ..const [initializing, setInitializing] = useState(true);  
17    ..const [user, setUser] = useState();  
18    ..  
19    ..const onAuthStateChanged = (user) => {  
20      ...setUser(user);  
21      ...if (initializing) {  
22        ...setInitializing(false);  
23      ...}  
24    ..}  
25  
26    ..useEffect(() => {  
27      ...const subscriber = auth().onAuthStateChanged(onAuthStateChanged);  
28      ...return subscriber; // unsubscribe on unmount  
29    ..}, []);
```

que a cada login

a login


-  Visão geral do projeto
- 
- Desenvolver**
 -  Authentication
 -  Cloud Firestore
 -  Realtime Database
 -  Storage
 -  Hosting
 -  Functions
 -  Machine Learning
- Qualidade**
 -  Crashlytics

Autenticação

Para cadastro usando E-mail/Senha, basta criar um formulário/estado para receber os dados de login e informá-los para o cadastro com seguinte código:

```
<Button
  onPress={ () => {
    auth()
    .createUserWithEmailAndPassword(email, senha)
    .then(() => {
      console.log('Usuário criado e logado após isso!');
    })
    .catch(error => {
      if (error.code === 'auth/email-already-in-use') {
        console.log('O e-mail já está em uso!');
      } else if (error.code === 'auth/invalid-email') {
        console.log('O e-mail é inválido');
      }
      console.error(error);
    });
  }}
  title="Cadastrar" />
```

Variáveis do seu estado aqui



Autenticação

Para autenticação usando E-mail/Senha já existentes:

```
<Button
  onPress={ () => {
    auth()
      .signInWithEmailAndPassword(email, senha)
      .then(() => {
        console.log('Usuário logado com sucesso!');
      })
      .catch(error => {
        console.error(error);
      });
  }}
  title="Acessar" />
```

Variáveis do seu estado aqui





Agenda

- O que é o Firebase?
- Recursos disponíveis
- Integração com o React-Native
- Autenticação
- **Leitura complementar**
- Exercícios

Leitura complementar

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

<https://firebase.google.com/use-cases>

<https://rnfirebase.io/>

<https://rnfirebase.io/auth/usage>

<https://developers.google.com/android/guides/client-auth>

<https://developer.android.com/studio/build/multidex>

<https://firebase.google.com/docs/android/setup#available-libraries>

<https://github.com/invertase/react-native-firebase>



Agenda

- O que é o Firebase?
- Recursos disponíveis
- Integração com o React-Native
- Autenticação
- Leitura complementar
- **Exercícios**

Exercícios

Desenvolva um aplicativo de login que permita o usuário se cadastrar e realizar o login usando o recurso de autenticação E-mail/senha do Firebase.

Regras:

- Se o usuário já existir, não deixar cadastrar novamente
- Ao logar, exibir uma tela informando que o usuário está em uma área restrita.
- Permitir o usuário deslogar

