

## Introdução

---

Na aula anterior, você aprendeu como listar dados usando o Firebase, nós vimos como adicionar um listener que recupera os dados diretamente do Firebase. Nesta aula, você verá como cadastrar um usuário e verificar se este usuário está logado.

## Configurações Iniciais

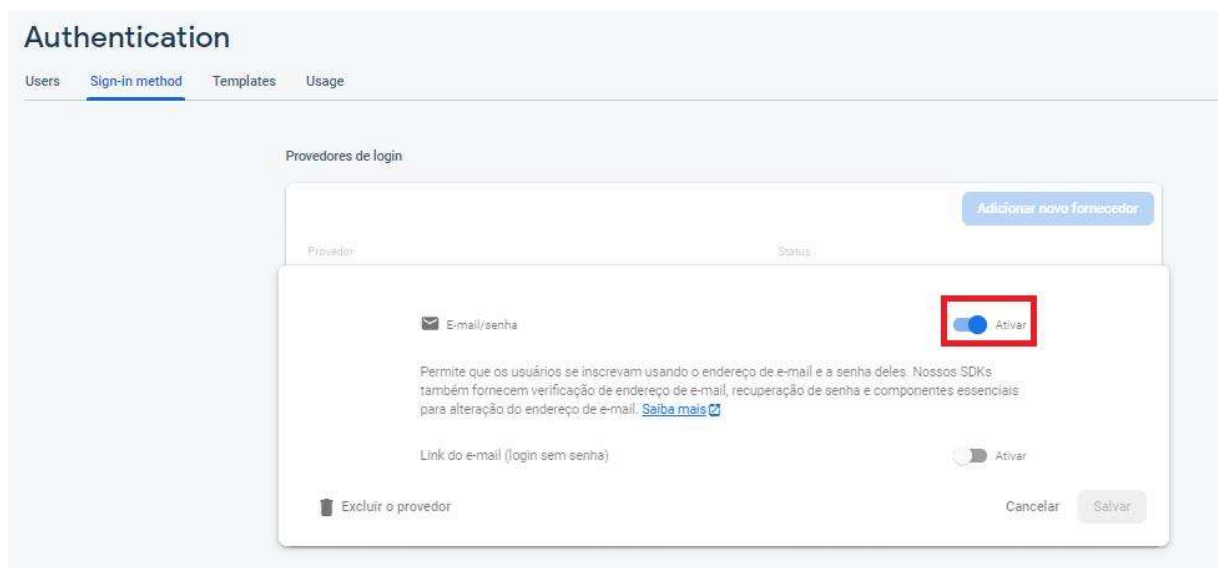
---

Vá até <https://firebase.google.com/>, clique em **ir para o console**, escolha o projeto que você está usando para as aulas, e clique no lado esquerdo da tela em **Authentication**.



Clique em **Sign-in methods (métodos de login)**. Nesta tela, você consegue visualizar quais os provedores de acesso que sua aplicação pode oferecer. Observe que estão disponíveis a autenticação com e-mail e senha, somente e-mail, Yahoo, Facebook, Github e outros provedores.

Neste caso, vou mostrar como programar uma tela de login usando e-mail e senha. Para isso, clique em **E-mail/Senha** e clique em **Ativar**.



Com isso, o usuário do seu aplicativo poderá fazer login usando e-mail e senha. Clique em **Users (Usuários)**.

## Authentication

Users Sign-in method Templates Usage

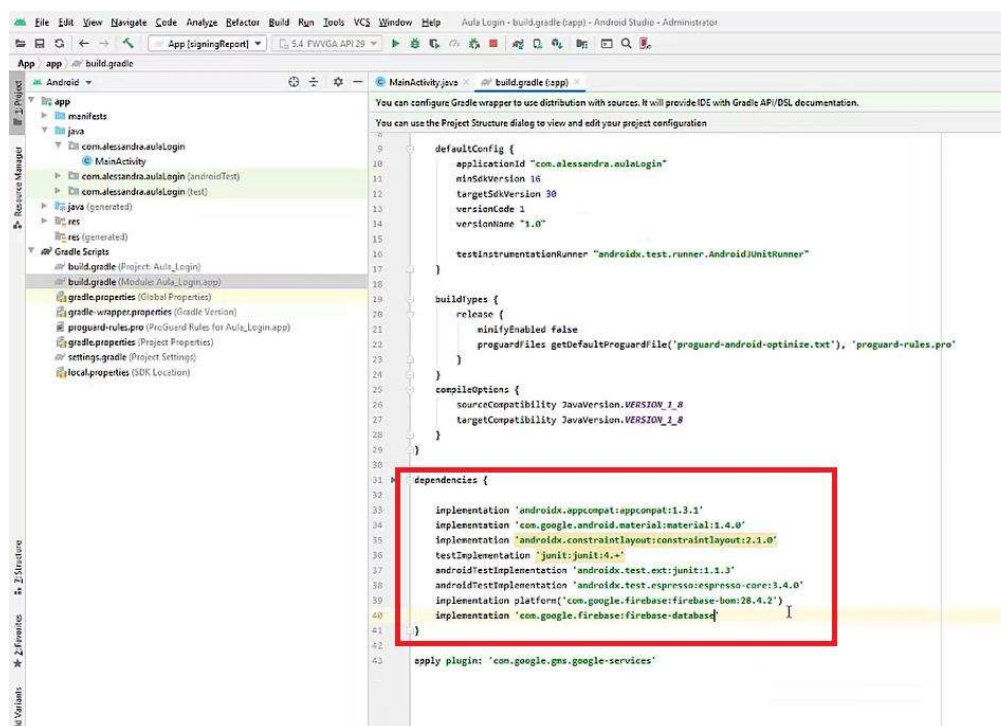
Desenvolva protótipos e realize testes de ponta a ponta com o Pacote do emulador local, agora com o Firebase Authentication. [Primeiros passos](#)

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			
alessandrapensadora2@g...		7 de out. de ...	7 de out. de ...	yWQq7EQ5YGb9ikH00BPWPi3qm...			
alessandrapensadora1@g...		6 de out. de ...	6 de out. de ...	NwVailTgHeerna5Qjdt1DN9Ef1			
alessandrapensadora@gm...		6 de out. de ...	6 de out. de ...	FXJzYJUJplexOMg070fK7fDIJvb2			
Linhas por página: 50					1 - 3 of 3		

Observe que eu tenho alguns usuários cadastrados. Você ainda não terá nenhum. É nesta tela que aparecerão os usuários que criarem uma conta no seu aplicativo.

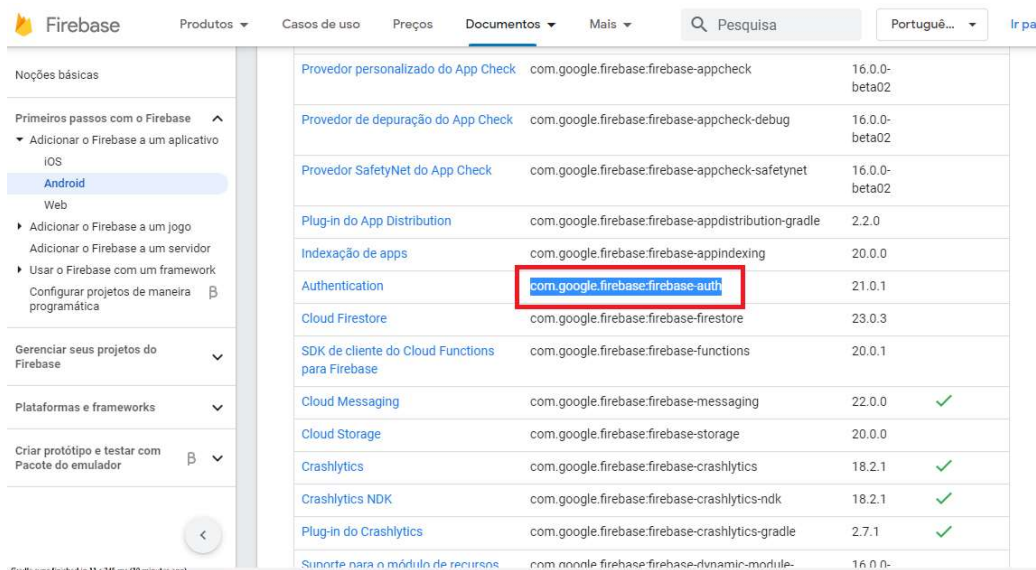
Para isso, nós iremos usar uma classe específica, utilizada para o gerenciamento de usuários (criar usuários, fazer login, fazer logoff etc.). Teremos que adicionar esta classe às dependências do Gradle.

Então, vá até a sua aplicação Android, clique em **Project**, até **App** e em seguida dê dois cliques em **Build Gradle (Module: app)**, vá até as **dependencies**:



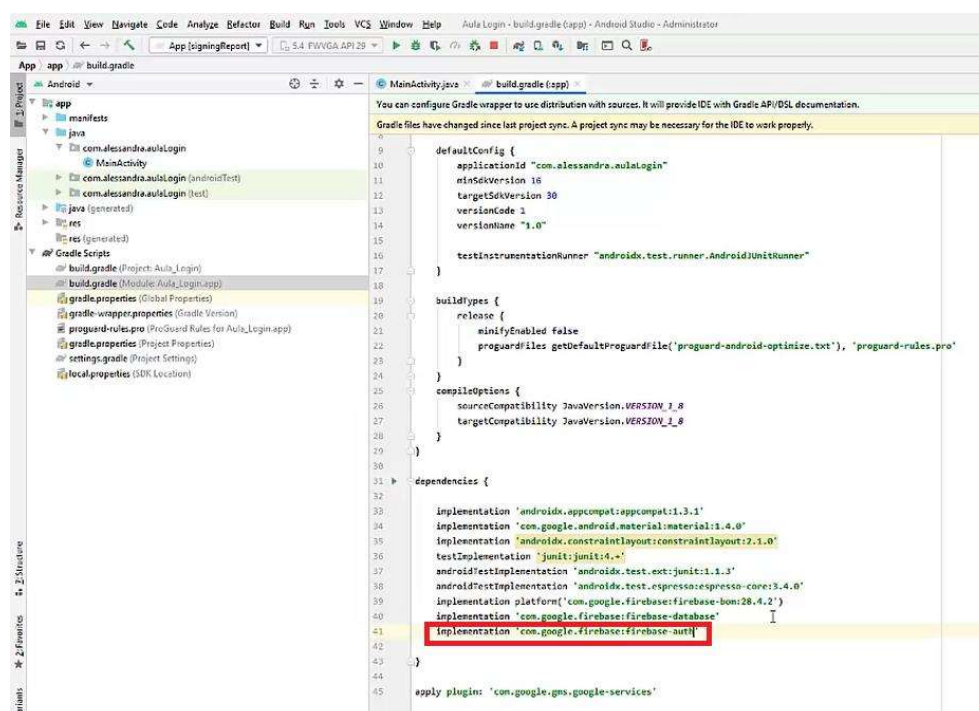
Note que foi neste local que adicionamos a biblioteca do Firebase Realtime Database, que foi necessária para fazer as implementações anteriores. Da mesma maneira, teremos que adicionar aí uma nova biblioteca.

Para isso, volte no Firebase, acesse a **documentação**, clique em **primeiros passos para android**, navegue até o final da página e copie a linha relacionada com a autenticação, conforme a ilustração:



Provedor personalizado do App Check	com.google.firebase:firebase-appcheck	16.0.0-beta02
Provedor de depuração do App Check	com.google.firebase:firebase-appcheck-debug	16.0.0-beta02
Provedor SafetyNet do App Check	com.google.firebase:firebase-appcheck-safetynet	16.0.0-beta02
Plug-in do App Distribution	com.google.firebase:firebase-appdistribution-gradle	2.2.0
Indexação de apps	com.google.firebase:firebase-appindexing	20.0.0
Authentication	<b>com.google.firebase:firebase-auth</b>	21.0.1
Cloud Firestore	com.google.firebase:firebase-firestore	23.0.3
SDK de cliente do Cloud Functions para Firebase	com.google.firebase:firebase-functions	20.0.1
Cloud Messaging	com.google.firebase:firebase-messaging	22.0.0
Cloud Storage	com.google.firebase:firebase-storage	20.0.0
Crashlytics	com.google.firebase:firebase-crashlytics	18.2.1
Crashlytics NDK	com.google.firebase:firebase-crashlytics-ndk	18.2.1
Plug-in do Crashlytics	com.google.firebase:firebase-crashlytics-gradle	2.7.1
Support para o módulo de recursos	com.google.firebase:firebase-dynamic-module-	16.0.0

Cole esta linha nas dependências do gradle nível módulo:



```
defaultConfig {
    applicationId "com.alexandra.aulaLogin"
    minSdkVersion 16
    targetSdkVersion 30
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
}

buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    }
}

dependencies {
    implementation 'androidx.appcompat:appcompat:1.3.1'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.0'
    testImplementation 'junit:junit:4.13.1'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation platform('com.google.firebase:firebase-bom:28.4.2')
    implementation 'com.google.firebase:firebase-database'
    implementation 'com.google.firebase:firebase-auth'
}

apply plugin: 'com.google.gms.google-services'
```

Não se esqueça de clicar em **sync now** para sincronizar a atualização que você fez no Gradle. Pronto, as configurações que precisávamos fazer estão feitas! Agora basta programar!

## Cadastrando Novos Usuários

---

Vá até a sua **MainActivity.Java** e crie o seguinte atributo:

```
private FirebaseAuth usuario = FirebaseAuth.getInstance();
```

O FirebaseAuth recupera o objeto que permitirá que você manipule os usuários. Com ele, eu consigo fazer o gerenciamento dos usuários.

Vamos começar cadastrando um novo usuário, da seguinte maneira:

```
usuario.createUserWithEmailAndPassword("alessandrapensadora@gmail.com", "a12344567").addOnCompleteListener(
    MainActivity.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            if(task.isSuccessful()){

                Toast.makeText(getApplicationContext(), "login efetuado com sucesso!", Toast.LENGTH_LONG).show();

            }else{

                Toast.makeText(getApplicationContext(), "ERRO AO FAZER LOGIN!", Toast.LENGTH_LONG).show();

            }

        }
    }
);
```

Note que eu usei o objeto **usuario** juntamente com o método **createUserWithEmailAndPassword**. Este é apenas um dos métodos que pode ser

utilizado para este fim. Neste caso, ele serve para criarmos um usuário usando e-mail e senha.

Esse método exige dois parâmetros, que são o e-mail e a senha do usuário. Neste código acima eu os escrevi de maneira estática, porém, você poderá recuperá-los de uma caixa de texto, desenhada na interface.

Somente com isso já seria o suficiente para que nosso código cadastrasse um usuário. Porém, para ficar mais completo, eu utilizo o **addOnCompleteListener** para tratar possíveis erros.

Dentro deste método, eu preciso passar o contexto da minha aplicação, que neste caso é o **MainActivity.this**, pois eu estou nesta classe, e também preciso implementar o **new OnCompleteListener**.

Ao fazer isso, automaticamente, é criado o método **onComplete**. Note que ele tem um objeto de retorno (**task**) que utilizamos para verificar se realmente deu certo o processo de cadastro.

Para isso, usamos **task.isSuccessful()** dentro de uma estrutura condicional, que no caso foi utilizado **if**, mas poderia ser um switch.

Por fim, eu coloquei um **toast**, que é uma mensagem de rodapé na tela do app, para exibir o status: se deu certo ou não o cadastro. Mas você poderia usar um log ou qualquer outra estrutura para exibir a informação.

## Verificando se um Usuário está Logado

---

Para verificar se um usuário está logado no seu aplicativo, usamos o método **getCurrentUser()**, que irá retornar um objeto usuário caso haja algum usuário logado. Em outras palavras, ele irá retornar o usuário atual da sua aplicação.

Então, para verificar se um usuário está logado, basta verificar se ele é diferente de null, através de uma estrutura condicional. Da seguinte forma:

```
if(usuario.getCurrentUser() != null){  
  
    Toast.makeText(getApplicationContext(),"usuário  
logado",Toast.LENGTH_LONG).show();  
}
```

```
        }else{  
            Toast.makeText(getApplicationContext(),"NÃO HÁ UM USUÁRIO  
LOGADO!",Toast.LENGTH_LONG).show();  
        }
```

Ao executar este trecho de código, ele deverá informar que há um usuário logado, uma vez que você acabou de cadastrar um novo usuário.

## Mais informações

---

Para maiores informações e também para responder as questões da semana, faça a leitura da documentação em:  
<https://firebase.google.com/docs/auth/android/firebaseui?hl=pt-Br>.