



INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: FIREBASE: REGISTRO E AUTENTICAÇÃO

Aula Prática 05

Parte 1: Configuração do Firebase

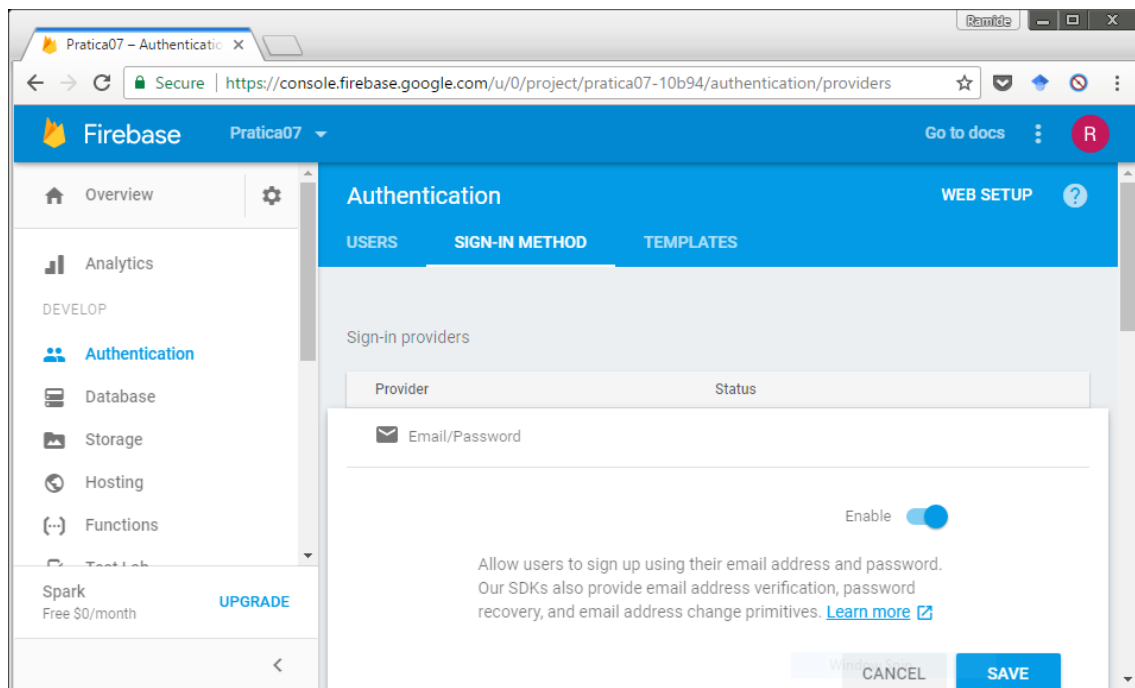
Passo 1: Visite <http://firebase.google.com> e habilite o serviço com sua conta pessoal do Google.

Faça o login com sua conta pessoal e clique em “Get started”. **Importante:** Use uma janela anônima se estiver fazendo a prática em uma máquina do laboratório.

Passo 2: Crie um novo projeto (Add project) chamado “Pratica05”. Escolha “Brazil” (ou “Brasil”) como região.

Passo 3: Habilite o método de autenticação via e-mail e senha.

No menu à esquerda, selecione “Authentication”, depois a aba “Sign-in Method” e habilite a opção “Email/Password”, como exibido abaixo:



Com isso estamos habilitando o serviço de autenticação via e-mail e senha do Firebase. Ele mantém essa base de credenciais, realizando criptografia e backups periódicos.

Parte 2: Preparando a aplicação para usar o Firebase

Passo 1: No Android Studio, crie um novo projeto chamado **Pratica05**, com uma `MainActivity` do tipo “Empty Activity”.

Importante: no campo “Company Domain”, forneça um nome de domínio único, usando, por exemplo, seu nome. Isso é importante porque o domínio é usado para identificar a aplicação no Firebase. Ao longo da prática será usado o domínio `pdm.tads.ifpe.edu.br`, não se esqueça de substituir com o seu.

Passo 2: Adicione o Firebase à sua aplicação, clicando em *Tools > Firebase*:

Será aberto uma aba lateral com opções de serviços do Firebase. Selecione “Authentication” e “Email and Password authentication”. Será fornecido um passo a passo de como adicionar o Firebase à aplicação (**imagem ao lado**). O primeiro passo é conectar a aplicação ao Firebase, clicando “Connect to Firebase”. Será mostrado um dialogo que lhe dará a opção de criar um novo projeto no Firebase ou usar um que já existe. Selecione a segunda opção e use “Pratica05”. Entre outras configurações, será baixado um arquivo chamada “google-services.json” para o diretório do seu projeto, que será usado pelo Firebase.

Depois selecione “Add Firebase Authentication to your app”. Isso adicionará as dependências necessárias no seus scripts gradle. Os passos seguintes são trechos de código para usar o serviço de autenticação do Firebase.

Os passos descritos acima podem ser feitos manualmente através do browser e editando os scripts gradle no Android Studio.

Email and password authentication

You can use Firebase Authentication to let your users sign in with their email addresses and passwords, and to manage your app's password-based accounts. This tutorial helps you set up an email and password system and then access information about the user.

[Launch in browser](#)

1 Connect your app to Firebase

Connect to Firebase

2 Add Firebase Authentication to your app

Add Firebase Authentication to your app

To use an authentication provider, you need to enable it in the [Firebase console](#). Go to the Sign-in Method page in the Firebase Authentication section to enable Email/Password sign-in and any other identity providers you want for your app.

3 Listen for auth state

Declare the `FirebaseAuth` and `AuthStateListener` objects.

```
private FirebaseAuth mAuth;
```

```
private FirebaseAuth.AuthStateListener mAuth
```

Parte 3: Criando um Novo Usuário

Passo 1: Crie uma nova atividade chamada `SignUpActivity` do tipo “Empty Activity”.

Essa atividade será usada para o registro de novos usuários.

Prepare o layout da atividade de forma a conter três campos de texto (`EditText`), com os seguintes ids: `edit_name`, `edit_email`, `edit_password`. Adicione um botão à atividade com o texto “Sign Up” (registrar). Para o campo senha use `android:inputType="textPassword"` para ocultar o conteúdo digitado. Disponha os elementos visualmente de forma adequada.

Passo 2: Para facilitar os testes, faça com que `SignUpActivity` seja a atividade “principal” da sua aplicação temporariamente.

Para isso, modifique o `AndroidManifest.xml` movendo o trecho delimitado por `<intent-filter>...</intent-filter>` que se encontra associado à `MainActivity` para a `SignUpActivity`.

Passo 3: Em `SignUpActivity`, associe um tratador de clicks ao botão com o nome `buttonSignUpClick`, contendo o código abaixo:

```
public void buttonSignUpClick(View view) {
    String email = edEmail.getText().toString();
    String password = edPassword.getText().toString();

    FirebaseAuth mAuth = FirebaseAuth.getInstance();
    mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                String msg = task.isSuccessful() ? "SIGN UP OK!":
                    "SIGN UP ERROR!";
                Toast.makeText(SignUpActivity.this, msg,
                    Toast.LENGTH_SHORT).show();
            }
        });
}
```

Esse método usa um objeto singleton do tipo `FirebaseAuth` para criar a credencial do nosso usuário no banco de dados remoto do Firebase. Se o usuário for registrado com sucesso, ele já é automaticamente logado no Firebase.

É registrado um tratador de eventos (`onCompleteListener<...>`) que é chamado quando o registro é completo. Nesse caso, ele apenas exibe um toast caso informando se a operação foi bem sucedida ou não. (Nesse trecho, `edEmail` e `edPassword` são referências do tipo `EditText` inicializadas no método `onCreate()` referenciando os respectivos campos.)

Passo 4: Teste a aplicação, fornecendo valores para os campos e-mail e senha.

Dica: Para facilitar os testes, dê valores default para esses campos através da propriedade `text` no arquivo de layout.

Passo 5: Verifique no console de administrador do Firebase se o usuário foi criado realmente.

Parte 4: Realizando o Login do Usuário

Passo 1: Modifique a `MainActivity` para que ela seja uma tela de Login:

Se baseie na atividade `SignUpActivity` e crie campos de login e senha (`edit_login` e `edit_passwd`) e um botão “Sign In”. Adicione também um botão para lançar a atividade `SignUpActivity` através de um `Intent`.

Passo 2: Desfaça o procedimento realizado no passo 2 da parte 1, de forma que a `MainActivity` volte a ser a atividade de entrada da aplicação.

Passo 3: Crie um tratador para o botão “Sign In” chamado `buttonSignInClick()`:

```
public void buttonSignInClick(View view) {
    String login = edEmail.getText().toString();
    String passwd = edPassword.getText().toString();

    FirebaseAuth mAuth = FirebaseAuth.getInstance();
    mAuth.signInWithEmailAndPassword(login, passwd)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                String msg = task.isSuccessful() ? "SIGN IN OK!":
                    "SIGN IN ERROR!";
                Toast.makeText(MainActivity.this, msg,
                    Toast.LENGTH_SHORT).show();
            }
        });
}
```

Esse método é semelhante ao de registro, porém usa a API do Firebase para autenticação com e-mail e senha (`mAuth.signInWithEmailAndPassword()`).

Passo 4: Teste a aplicação. Use tanto credenciais corretas como incorretas.

Passo 5: Crie uma atividade chamada `HomeActivity`. Nessa prática ela deve conter apenas uma mensagem de boas vindas e um botão “Sign Out”.

Passo 6: Modifique temporariamente o método `buttonSignInClick()` em `MainActivity` de forma a lançar um `Intent` para a atividade `HomeActivity`.

Passo 7: Crie um tratador para o botão “Sign Out” chamado `buttonSignOutClick()`, em `HomeActivity` com o código abaixo:

```
public void buttonSignOutClick(View view) {
    FirebaseAuth mAuth = FirebaseAuth.getInstance();
    FirebaseUser user = mAuth.getCurrentUser();
    if (user != null) {
        mAuth.signOut();
        this.finish();
    } else {
        Toast.makeText(HomeActivity.this, "Error!", Toast.LENGTH_SHORT).show();
    }
}
```

Esse método faz o log out do usuário. Primeiro testa se existe um usuário logado (`mAuth.getCurrentUser()`) e, em caso positivo, chama `signOut()`.

Passo 8: Rode e teste a aplicação.

Parte 5: Monitorando o Status da Sessão do Usuário

Passo 1: Crie a classe `FirebaseAuthListener` com o código abaixo:

```
public class FirebaseAuthListener implements FirebaseAuth.AuthStateListener {

    private final Activity activity;

    public FirebaseAuthListener(Activity activity) {
        this.activity = activity;
    }

    @Override
    public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
        FirebaseUser user = firebaseAuth.getCurrentUser();

        Intent intent = null;
        if ((user != null) && !(activity instanceof HomeActivity)) {
            intent = new Intent(activity, HomeActivity.class);
        }

        if ((user == null) && (activity instanceof MainActivity)) {
            intent = new Intent(activity, MainActivity.class);
        }

        if (intent != null) {
            activity.startActivity(intent);
            activity.finish();
        }
    }
}
```

Essa classe monitora a sessão do usuário, lançando a atividade adequada: `HomeActivity`, caso o usuário esteja logado; `MainActivity`, caso contrário.

Passo 2: Em **todas** as atividades, **adicione** o trecho abaixo ao `onCreate()`:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    this.mAuth = FirebaseAuth.getInstance();
    this.authListener = new FirebaseAuthListener(this);
}
```

Adicione as propriedades `mAuth` e `authListener` com os tipos adequados.

Passo 3: Adicione os métodos `onStart()` e `onStop()` em **todas** as atividades:

```
@Override
public void onStart() {
    super.onStart();
    mAuth.addAuthStateListener(authListener);
}

@Override
public void onStop() {
    super.onStop();
    mAuth.removeAuthStateListener(authListener);
}
```

Dessa forma habilitamos e desabilitamos o listener sempre que uma atividade for iniciada ou parada, respectivamente.

Passo 4: Desfaça o passo 6 da parte 4 e remova `this.finish()` de `buttonSignInClick()` na `HomeActivity`.

Passo 5: Rode e teste a aplicação.