

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

Prática 05

ATENÇÃO: Continuação da Prática 04; use controle de versões (Git).

Parte 1: Configuração do Firebase

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

Faça o login com sua conta do instituto e clique em “*Get started*” (pode ser preciso usar a sua conta pessoal caso tenha problemas). Ou vá direto no console: <https://console.firebase.google.com/>

Importante: Use uma janela anônima/privativa se estiver fazendo a prática no laboratório. Ao usar suas credenciais no Android Studio, faça o *logoff* depois.

Passo 2: No Firebase Console, vá em *Add project + (Criar um projeto +)*.

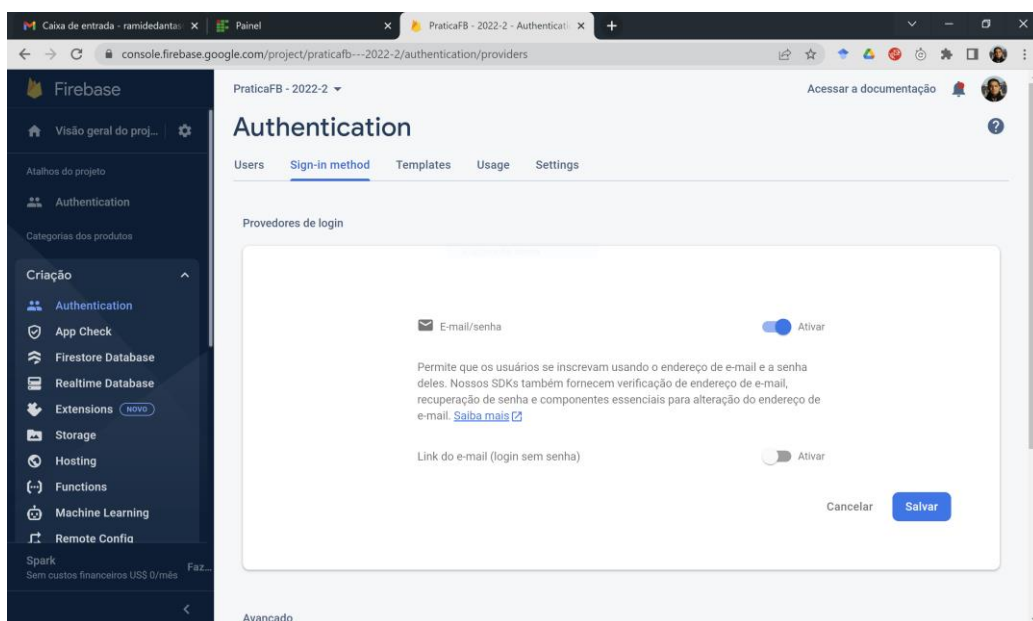
Escolha o projeto “WeatherApp” criado nas Prática 04. (Pode ser preciso clicar em “Adicionar o Firebase ao projeto do Cloud” abaixo).

Desative a opção “*Analytics*” nas próxima páginas e finalize a criação.

Passo 3: Habilite o método de autenticação via e-mail e senha no novo projeto.

No menu à esquerda, selecione Criação > *Authentication*, depois a aba “*Sign-in Method*”/“Método de login” e habilite a opção “*Email/Password*” (ver 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.

OBS.: Clique em “*Primeiros Passos/Vamos Começar/First Steps*” se aparecer um tutorial antes.

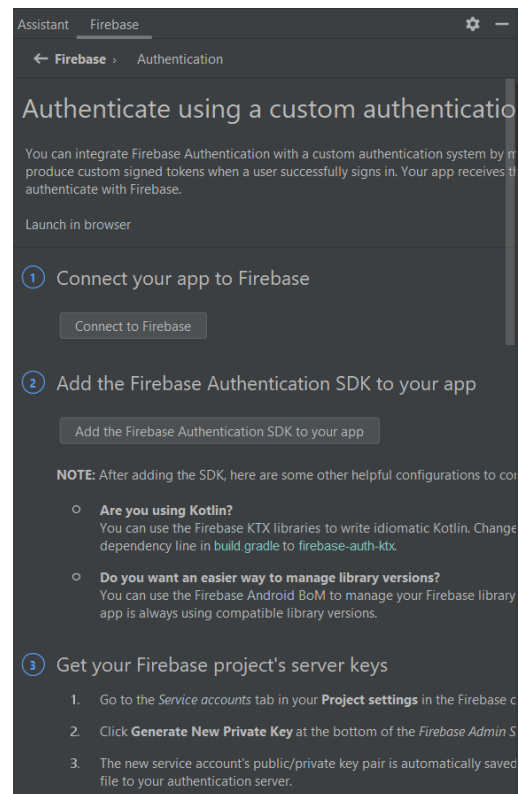


Passo 4: No Android Studio (AS), abra o projeto **WeatherApp** e adicione suporte ao Firebase à sua aplicação (menu *Tools > Firebase*):

Será aberta uma aba lateral com opções de serviços do Firebase. Selecione *Authentication > Authenticate using a custom authentication system*. 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* (Passo 1 na imagem). O Android Studio pode solicitar para logar na sua conta Google.

Será aberto um *browser* para você escolher qual projeto associar ao App. Escolha o projeto **WeatherApp**. Esse passo irá configurar sua aplicação adequadamente.

No Android Studio, selecione *Add Firebase Authentication SDK to your app* (Passo 2 na imagem). Isso adicionará as dependências necessárias no seus *scripts gradle*. O passo 3 não se aplica à prática (usado com servidor próprio). Os passos seguintes são trechos de código para usar o serviço de autenticação do Firebase.



Passo 5: Rode e teste o aplicativo para verificar se tudo está OK. Faça um commit se for o caso.

Por questões de segurança, não inclua o arquivo **google-services.json** no commit pois ele possui sua chave de acesso ao Firebase.

Parte 2: Realizando o Registro, Login e Logout de Usuários

Passo 1: Em *RegisterActivity*, faça o tratador de *click* do botão “Registrar” igual abaixo:

```
firebase.auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(activity!!) { task ->
        if (task.isSuccessful) {
            Toast.makeText(activity,
                "Registro OK!", Toast.LENGTH_LONG).show()
            activity.finish()
        } else {
            Toast.makeText(activity,
                "Registro FALHOU!", Toast.LENGTH_LONG).show()
        }
    }
}
```

Esse método usa o objeto *singleton* *Firebase* para criar a credencial do usuário no banco de dados remoto do Firebase. Se o usuário for registrado com sucesso, ele é automaticamente logado, isto é, **não é preciso solicitar login após o registro**. Não estamos salvando o nome do usuário nesse momento; será feito na Prática 06. Também não é feita verificação quanto a existência do e-mail ou pertence ao usuário; basta ter o formato válido. Use uma senha fácil de lembrar com 6 caracteres ou mais.

Nesse código, um tratador de eventos é chamado quando o registro é terminado. É exibido um *toast* informando se a operação foi bem-sucedida ou não. Em caso de sucesso, a atividade é finalizada.

Passo 2: Teste a aplicação, fornecendo valores para os campos e-mail e senha.

ATENÇÃO: Em caso de erro de registro, desinstale o App do emulador/celular.

Verifique no console do Firebase se o usuário foi criado realmente.

Passo 3: Na LoginActivity, altere o tratados do botão “Login” para o código abaixo:

```


Firebase.auth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(activity!!) { task ->
        if (task.isSuccessful) {
            activity.startActivity(
                Intent(activity, MainActivity::class.java).setFlags(
                    FLAG_ACTIVITY_SINGLE_TOP
                )
            )
            Toast.makeText(activity, "Login OK!", Toast.LENGTH_LONG).show()
        } else {
            Toast.makeText(activity, "Login FALHOU!", Toast.LENGTH_LONG).show()
        }
    }
}

```

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

Passo 4: Rode e teste a aplicação. Use tanto credenciais corretas como incorretas.

ATENÇÃO: O código acima não trata o caso de credenciais vazias.

Passo 5: Modifique tratador do botão de sair  em MainActivity com o código abaixo:

```

Firebase.auth.signOut()
finish()

```

Esse método faz o *logout* do usuário.

Passo 6: Rode e teste a aplicação.

Caso esteja tudo certo, faça um novo *commit*.

Parte 3: Monitorando o Status da Sessão do Usuário

Passo 1: Altere o código do MainViewModel, adicionando o trecho abaixo:

```
class MainViewModel : ViewModel() {  
    ...  
    private var _loggedIn = mutableStateOf(false)  
    val loggedIn: Boolean  
        get() = _loggedIn.value  
  
    private val listener = FirebaseAuth.AuthStateListener {  
        firebaseAuth ->  
            _loggedIn.value = firebaseAuth.currentUser != null  
    }  
  
    init {  
        listener.onAuthStateChanged(Firebase.auth)  
        Firebase.auth.addAuthStateListener(listener)  
    }  
  
    override fun onCleared() {  
        Firebase.auth.removeAuthStateListener(listener)  
    }  
}
```

O trecho adicionado monitora o status da sessão do Firebase através do objeto `listener`, que é ativado sempre que o status muda e seta o atributo booleano `loggedIn` de acordo. Esse atributo é um `State` que é monitorado pela interface, nesse caso, a `MainActivity` (próximo passo). Obs.: quando o `ViewModel` é criado, configuramos o `listener` no `Firebase.auth` (bloco `init`). Quando o `ViewModel` está para ser destruído (`onCleared()`), retiramos o `listener` para evitar que o `ViewModel` fique “pendurado” na memória desnecessariamente.

Passo 2: Altere `MainActivity`, adicionando o trecho abaixo dentro de `setContent()`, depois de pegar o `viewModel`:

```
if (!viewModel.loggedIn) {  
    this.finish()  
}
```

Esse trecho finaliza a atividade se o usuário for deslogado (por conta própria ou automaticamente), impedindo que fique na tela principal se não estiver logado.

Passo 3: Ainda em `MainActivity`, retire o trecho que chama `finish()` explicitamente ao realizar o *sign-out* (ou *log off*, clique do botão de sair).

Isso já não mais necessário. Ao realizar o *sign-out* no Firebase, o código adicionado no passo 1 é ativado, mudando `viewModel.loggedIn` para `false` e forçando a recomposição de `MainActivity`, que entra no `if` do passo 2 e termina.

Passo 4: Rode e teste a aplicação. Faça um *commit* se estiver OK.

Parte 4: (Desafio) Controle de sessão e LoginActivity e RegisterActivity

Passo 1: Refatore `LoginActivity` e `RegisterActivity` como fizemos em `MainActivity`.

Crie um `ViewModel` base contendo as informações adicionadas no Passo 1. que passará a ser a classe base de `MainViewModel` (remova o código nela; a classe base precisa ser declarada como `open`), devendo ser instanciado em `LoginActivity` e `RegisterActivity`. Modifique essas atividades como no Passo 2, mas fazendo com que naveguem automaticamente para `MainActivity` caso o usuário já esteja logado. Remova a navegação quando realizamos o *sign-in* e *sign-up* (isto é, as chamadas à `startActivity()` saem desse pontos).