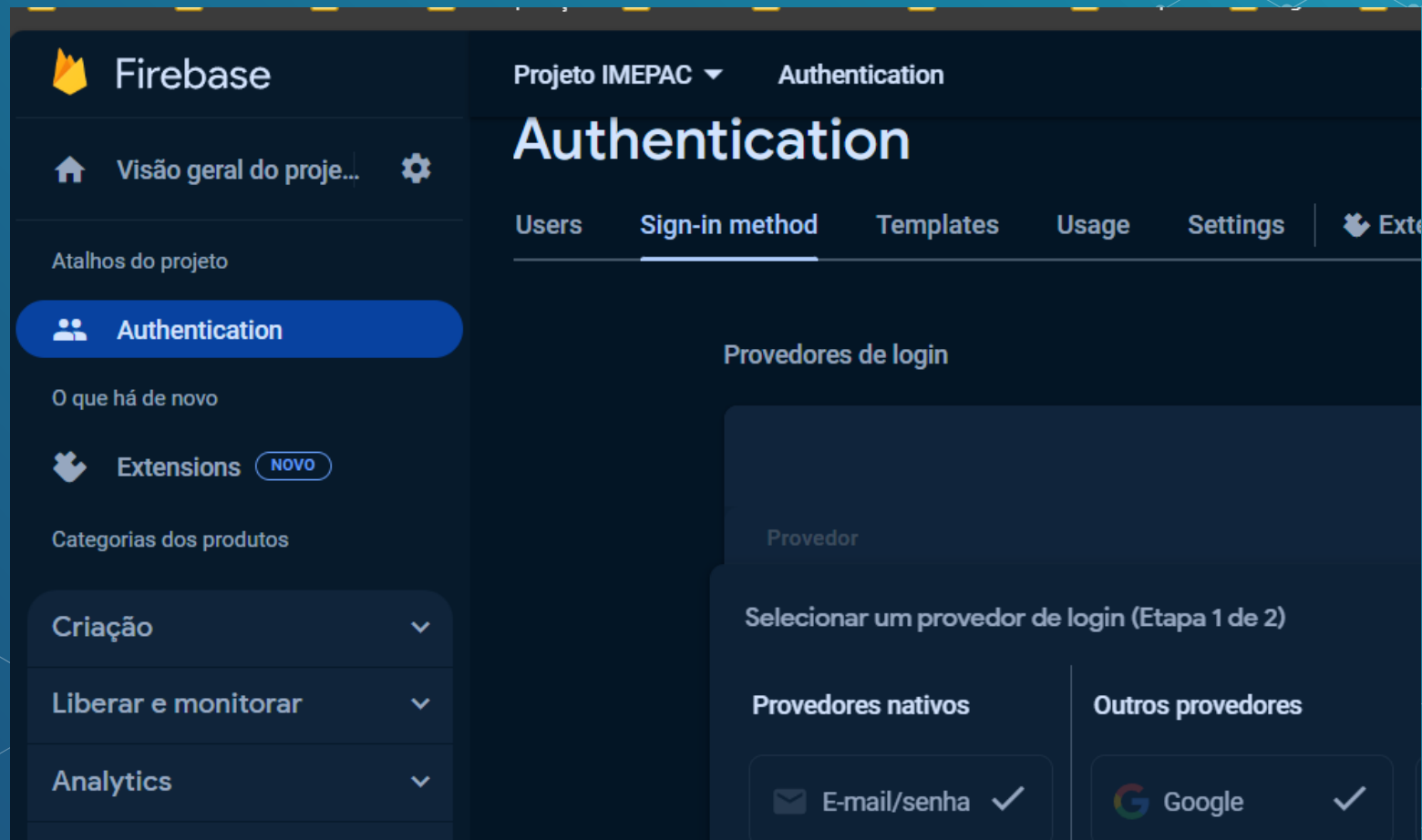


# Autenticação e Cadastro de Usuários

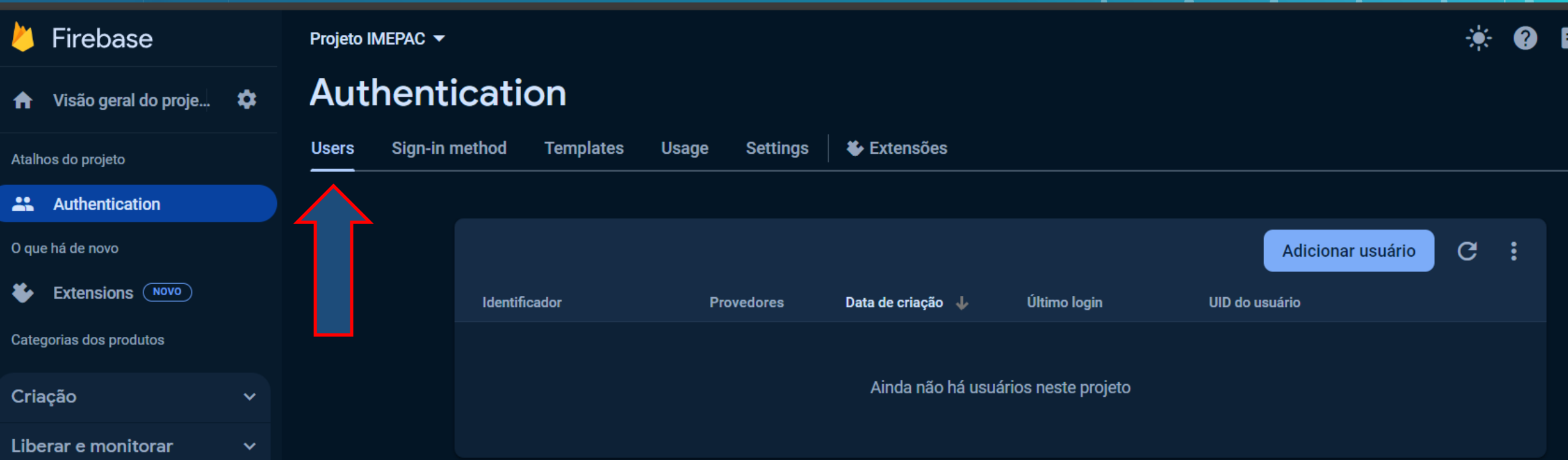
Professor: Crispim Luiz Martins

# Cadastrar usuários - FormCadastro

- ◆ No <https://console.firebase.google.com/> vá e authentication vamos habilitar o cadastro por e-mail e senha.



User será habilitado para ver os usuários até cadastrá-los diretamente



Projeto IMEPAC

## Authentication

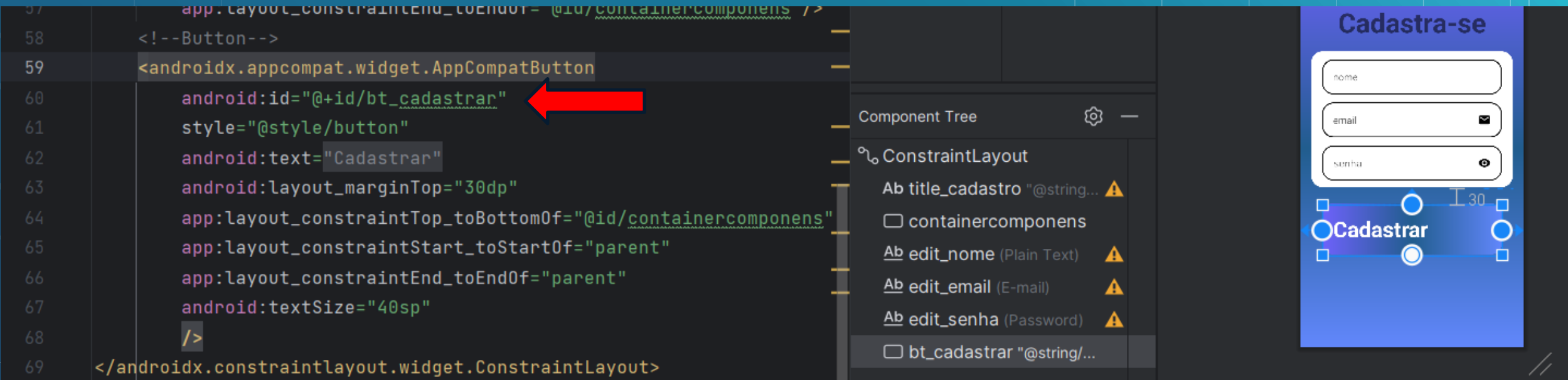
Users Sign-in method Templates Usage Settings Extensões

Adicionar usuário

Identificador	Provedores	Data de criação ↓	Último login	UID do usuário
Ainda não há usuários neste projeto				

# Começaremos a pensar sobre os dados da aplicação seja enviado para nosso banco

- ◆ Entenda que agora nossos id de cada elemento são importantes. Identificam para input de texto e ação feita pelo button.



The screenshot displays the Android Studio interface. On the left, the XML code for a button is shown, with a red arrow pointing to the `android:id="@+id/bt_cadastrar"` attribute. The code includes various styling and layout attributes. On the right, the 'Component Tree' panel lists the UI components, including `bt_cadastrar`. To the far right, a preview of the app shows a registration form titled 'Cadastra-se' with fields for 'nome', 'email', and 'senha', and a 'Cadastrar' button.

```
57 app:layout_constraintEnd_toEndOf="@id/containercomponens" />
58 <!--Button-->
59 <androidx.appcompat.widget.AppCompatButton
60     android:id="@+id/bt_cadastrar"
61     style="@style/button"
62     android:text="Cadastrar"
63     android:layout_marginTop="30dp"
64     app:layout_constraintTop_toBottomOf="@id/containercomponens"
65     app:layout_constraintStart_toStartOf="parent"
66     app:layout_constraintEnd_toEndOf="parent"
67     android:textSize="40sp"
68 />
69 </androidx.constraintlayout.widget.ConstraintLayout>
```

- ◆ Veja que button tem um id, e ao clicar no elemento pelo modelo nos mostra cada um. Veja que seu Button tem id, se não tiver coloque!!!



# Criando nossas variáveis



```
FormCadastro.kt x  </> activity_form_cadastro.xml
1  package br.com.faculdade.imepac
2
3  > import ...
9
10 class FormCadastro : AppCompatActivity() {
11     //Criando as variáveis para receber os id
12     private lateinit var edit_nome: EditText
13     private lateinit var edit_email: EditText
14     private lateinit var edit_senha: EditText
15     private lateinit var btnCadastrar: Button
16
```

- ◆ O modificador `lateinit` em Kotlin é usado para declarar propriedades que serão inicializadas posteriormente, fora do construtor. Essa propriedade não pode ser nula e não precisa ser inicializada imediatamente. No entanto, se você acessá-la antes de inicializá-la, ocorrerá uma exceção.

A função `onCreate` ficou na linha abaixo





# Em FormCadastro já tínhamos um função onCreate, vamos usa-la para criamos nossa regra de cadastro

FormCadastro.kt x activity\_form\_cadastro.xml

```
22  override fun onCreate(savedInstanceState: Bundle?) {  
23      super.onCreate(savedInstanceState)  
24      setContentView(R.layout.activity_form_cadastro)  
25      //No comando abaixo mando esconder o Toolbar  
26      getSupportActionBar()?.hide();  
27  
28      // o que estiver no elemento será enviado para a variável  
29      edit_nome = findViewById(R.id.edit_nome)  
30      edit_email = findViewById(R.id.edit_email)  
31      edit_senha = findViewById(R.id.edit_senha)  
32      btnCadastrar = findViewById(R.id.bt_cadastrar)  
33  }
```

- ◆ Até a linha 26 está tudo normal, como antes.
- ◆ Linha 28 começamos e pegar o elementos por seus ids e passamos para as variáveis!

R dentro do (R.id.idpedido) é uma classe de recursos produzido pelo Android durante o processo de compilação é feita para pegar o id e separar um campo de id dentro do Android, então no exemplo da linha 29 temos uma buscar do classe R no elemento que tem o id edit\_nome



# Função setOnClickListener que é acionada pelo button btnCadastrar.

```
34 btnCadastrar.setOnClickListener { it: View!           toString converte para String
35     val nome = edit_nome.text.toString().trim()
36     val email = edit_email.text.toString().trim()      Trim tira espaços em branco
37     val senha = edit_senha.text.toString().trim()
38
39     if (nome.isEmpty() || email.isEmpty() || senha.isEmpty()) {
40         val mensagemErro = "Campos não preenchidos, tente novamente"
41         val snackbar = Snackbar.make(it, mensagemErro, Snackbar.LENGTH_LONG);
42         snackbar.show();
43     } else {
44         cadastrarUsuario(it);
45     }
46 }
47 }
```

◆ Nosso button ao ser clicado irá pegar os recursos que estão nos id e serão lançados nas variáveis.

◆ Dentro da mesma função temos um if que pergunta se tem algo preenchido nas variáveis, se não, temos mensagem para preencher.

Na linha 43 temos um else que se está tudo preenchido, passando pelo if irá acionar a função `cadastrarUsuario(it)`.



```

48 fun cadastrarUsuario(it: View){
49     val email = edit_email.text.toString().trim()
50     val senha = edit_senha.text.toString().trim()
51     FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, senha)
52         .addOnCompleteListener{task-> //task é o objeto do resultado vindo o firebase
53         if (task.isSuccessful){
54             val mensagemOk = "Cadastro realizado com sucesso"
55             val snackbar = Snackbar.make(it, mensagemOk, Snackbar.LENGTH_LONG);
56             snackbar.show();
57         }else{
58             val mensagemErro = "Erro ao cadastrar usuário"
59             val snackbar = Snackbar.make(it, mensagemErro, Snackbar.LENGTH_LONG);
60             snackbar.show();
61         }
62     }
63 }
64 }
65 }

```

Task → pega a informação de e-mail e senha e enviar para o banco, se sucesso temos a mensagem e se erro também temos a mensagem.

- ◆ Linha 48 → nossa função tem que enviar uma mensagem de sucesso ou erro, por isso, a variável no parâmetro. Veja que para o cadastro não precisamos do nome, por isso, somente Email e senha.
- Linha 51 → Veja que usamos os recursos do banco Firebase. Pode ter acesso como se faz na documentação, mas já está feito para compreensão.







## Bora testar

- ◆ Preenchido, se a mensagem der ok, pode ir ver no seu banco de dados

