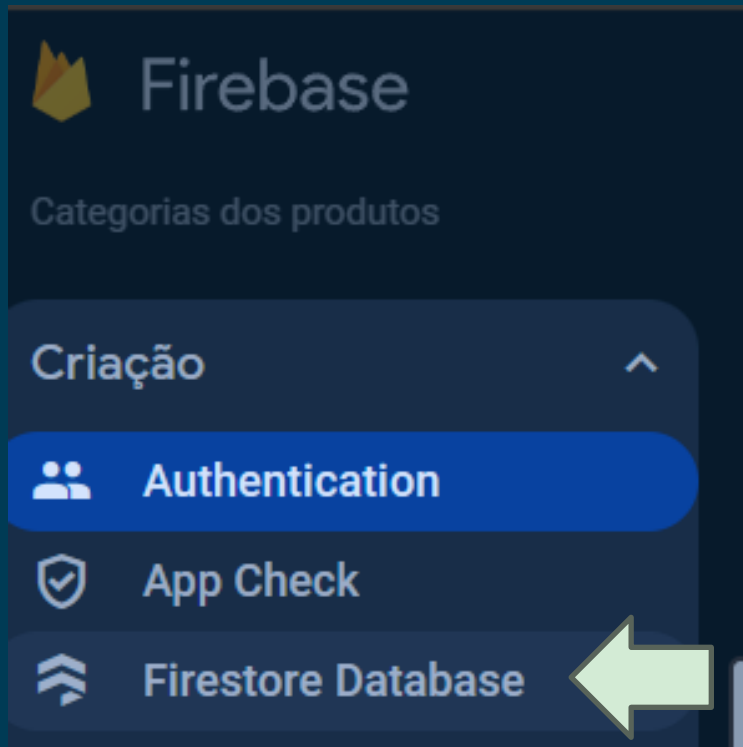


Salvando nosso dados no Firebase

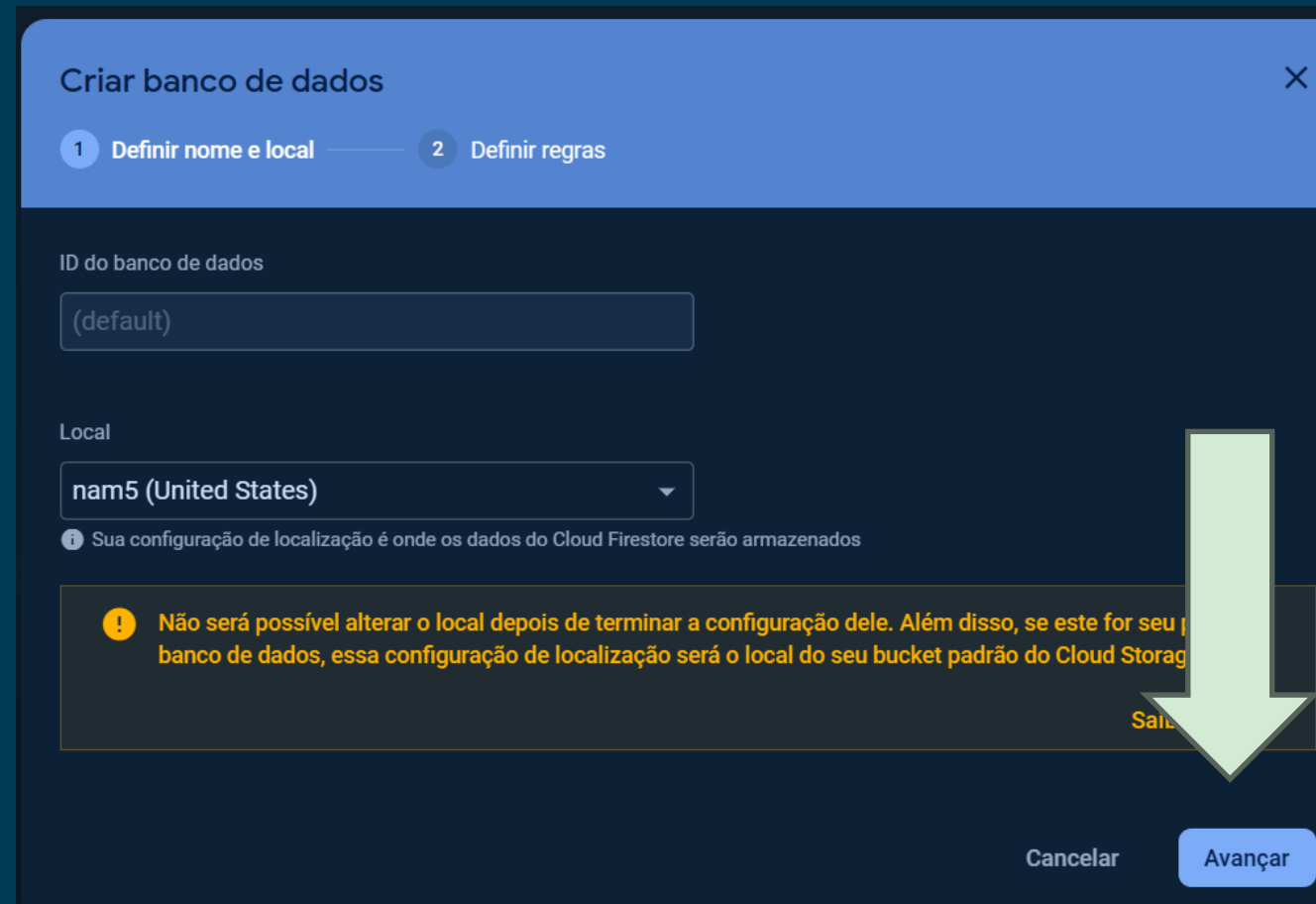
PROFESSOR CRISPIM LUIZ MARTINS

Voltando nosso assunto.

Quando adicionamos o cadastro, deve-se lembrar que o nome não é salvo. Sendo assim, temos que criar um banco onde colocaremos os dados.



Clique em Firestore



Modulo teste

Criar banco de dados

✓ Definir nome e local — 2 De

Após definir a estrutura, é preciso criar regras.
[Saiba mais](#)

☐ Iniciar no modo de produção

Seus dados são particulares por padrão. O acesso de leitura/gravação do cliente vai ser concedido apenas se especificado por suas regras de segurança.

☒ Iniciar no modo de teste

Por padrão, seus dados estão definidos para permitir uma configuração rápida. Porém, você precisa atualizar suas regras de segurança em até 30 dias para permitir o acesso de leitura/gravação do cliente em longo prazo.

Projeto IMEPAC ▼

Cloud Firestore

Dados

Regras

Índices

Uso

Extensions

Mudando nossas regras

Tirei os comentários, e vou alterar nossas regras para que só pode Mudar ou ler quem estiver logado. Veja no próximo slide.

Cloud Firestore

DadosRegrasÍndicesUsoExtensions

★ Hoje • 4:38 PM

○ Hoje • 4:36 PM

```
1 rules_version = '2';
2
3 service cloud.firestore {
4   match /databases/{database}/documents {
5
6
7     match /{document=**} {
8       allow read, write: if request.time < timestamp.date(2024, 3, 27);
9     }
10  }
11 }
```

Usuário deve estar logado

Dados Regras Índices Uso Extensions



Hoje • 4:41 PM



Hoje • 4:38 PM



Hoje • 4:36 PM

```
1 rules_version = '2';
2
3 service cloud.firestore {
4   match /databases/{database}/documents {
5
6
7     match /{document=**} {
8       allow read, write: if request.auth != null;
9     }
10  }
11 }
```

Para ler e escrever deve ser diferente de null

Antes de mais nada!!!!!!

O Firebase Authentication e o Firebase Firestore são tecnologias relacionadas, mas têm propósitos diferentes:

Suporta autenticação usando senhas, números de telefone e provedores de identidade federados como Google, Facebook e Twitter. Com bibliotecas de IU prontas para autenticar usuários no seu aplicativo e segue padrões como OAuth 2.0 e OpenID Connect.

Firebase Firestore:

É um banco de dados em tempo real e totalmente gerenciado.

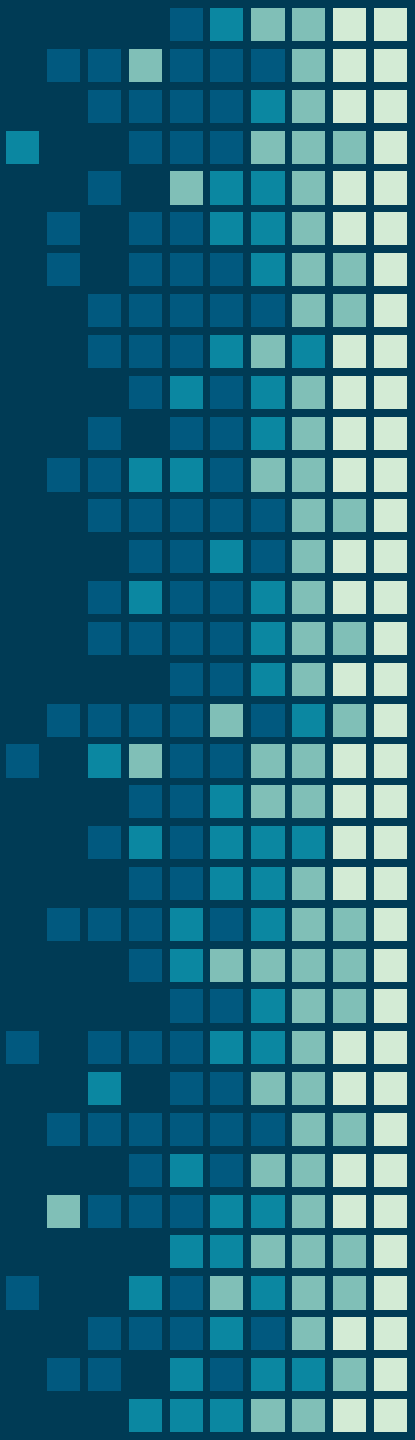
Oferece atualizações em tempo real, sincronização offline, escalabilidade e implantação em várias regiões.

Embora seja parte do Firebase, é um serviço separado e não está diretamente relacionado à autenticação.

Por isso, irei clonar os dados que seriam do Auth para o Firestore. Sendo assim, uso o Auth somente para conexão e não para dados.

Nossa próxima função pegará os dados que já irão para o Auth e salvará os mesmo dados no Firestore.

Após o código de como salvados dados no
Firebase Firestore teremos slides de como
ficou todo formCadastro.
Caso fique confuso a explicação, passe na
hora de fazer o código para toda a tela.



Vamos para o código para salvar nome do usuário

Iremos criar uma variável para pegar o usuário logado em FormCadastro.kt

```
16
17 ▶ </> class FormCadastro : AppCompatActivity() {
18     //Criando as variáveis para receber os id
19     private lateinit var edit_nome: EditText
20     private lateinit var edit_email: EditText
21     💡 private lateinit var edit_senha: EditText
22     private lateinit var btnCadastrar: Button
```


Quando for para cadastrar o usuário queremos salvar o nome e demais no Firestore

Pense comigo, já temos uma função para cadastrarUsuário. Temos no mesmo formulário a variável que desejamos salvar. Quando o usuário clica em criar usuário já podemos salvar essas variáveis. Vamos chamar nossa função dentro de outra.

```
51 fun cadastrarUsuario(it: View){
52     val email = edit_email.text.toString().trim()
53     val senha = edit_senha.text.toString().trim()
54     FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, senha)
55         .addOnCompleteListener{task-> //task é o objeto do resultado vindo o firebase
56         if (task.isSuccessful){
57             salvarDadosUsuario()//metodo para salvar o cadastro
58             val mensagemOk = "Cadastro realizado com sucesso"
59             val snackbar = Snackbar.make(it, mensagemOk, Snackbar.LENGTH_LONG)
60             snackbar.show()
61         }else{
62             val mensagemErro = "Erro ao cadastrar usuário"
63             val snackbar = Snackbar.make(it, mensagemErro, Snackbar.LENGTH_LONG)
64             snackbar.show()
65         }
66     }
67 }
68 }
```



salvarDadosUsu

71 – Variável do banco

72 – Variável nome.

73 e 74 – Busca do Auth

75 – Verifica se veio do

banco as variáveis se

diferente de null e

hasMapOf para envio

para o banco

81 – se der certo adicione o

Usuário ou errado dará uma

mensagem na tela.

Em usuários serão criados

coleções com dados de cada

usuário.

TelaPerfil.kt

FormCadastro.kt ×

```
70 fun salvarDadosUsuario() {
71     val db = FirebaseFirestore.getInstance()
72     val nome = edit_nome.text.toString().trim()
73     val usuarioID = FirebaseAuth.getInstance().currentUser?.uid
74     val email = FirebaseAuth.getInstance().currentUser?.email
75     if (usuarioID != null && email != null) {
76         val usuarios = hashMapOf(
77             "nome" to nome,
78             "email" to email,
79             "uid" to usuarioID
80         )
81         db.collection(collectionPath: "Usuarios") CollectionReference
82             .add(usuarios) Task<DocumentReference!>
83             .addOnSuccessListener { documentReference -> // Add com sucesso
84                 println("Documento adicionado com ID: ${documentReference.id}")
85             }
86             .addOnFailureListener { e -> // Ocorreu um erro ao adicionar
87                 println("Erro ao adicionar documento: $e")
88             }
89     } else { // O usuário não está autenticado
90         println("Erro na autenticação")
91     }
92 }
93 }
```

Cloud Firestore

Adicionar banco de dados

Dados Regras Índices Uso Extensions

Proteja os recursos do Cloud Firestore de abusos, como fraude de faturamento ou phishing Configurar o App Check

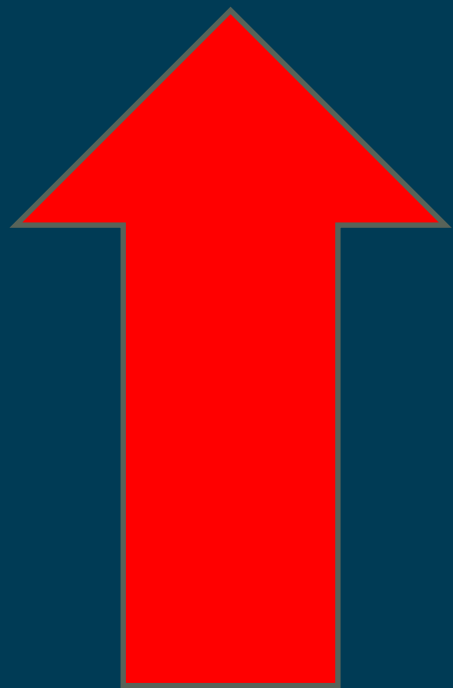
Visualização do painel

Home > Usuarios > AITbUHUGMIby..

Ma

(default)	Usuarios	AITbUHUGMIby9eZp1LYr
+ Iniciar coleção	+ Adicionar documento	+ Iniciar coleção
Usuarios >	AITbUHUGMIby9eZp1LYr >	+ Adicionar campo
	b5LbTniPSVuovLREF95Q sV2lnYeVz06152597pb6	email: "teste1@live.com" nome: "teste1" uid: "0CCdCLCxqgapHoWhL2jhgbUCUlo2"

Como está toda a nossa tela FormCadastro



```
TelaPerfil.kt  FormCadastro.kt ×
1      package br.com.faculdade.imepac
2
3      > import ...
16
17  ▶ </> class FormCadastro : AppCompatActivity() {
18          //Criando as variáveis para receber os id
19          private lateinit var edit_nome: EditText
20          private lateinit var edit_email: EditText
21          private lateinit var edit_senha: EditText
22          private lateinit var btnCadastrar: Button
```

```
24  @↑ override fun onCreate(savedInstanceState: Bundle?) {
25      super.onCreate(savedInstanceState)
26      setContentView(R.layout.activity_form_cadastro)
27      //No comando abaixo mando esconder o Toolbar
28      supportActionBar?.hide()
29
30      // o que estiver no elemento será enviado para a variável
31      edit_nome = findViewById(R.id.edit_nome)
32      edit_email = findViewById(R.id.edit_email)
33      edit_senha = findViewById(R.id.edit_senha)
34      btnCadastrar = findViewById(R.id.bt_cadastrar)
35
36      btnCadastrar.setOnClickListener { it: View!
37          val nome = edit_nome.text.toString().trim()
38          val email = edit_email.text.toString().trim()
39          val senha = edit_senha.text.toString().trim()
40
```

TelaPerfil.kt

FormCadastro.kt ×

```
40
41     if (nome.isEmpty() || email.isEmpty() || senha.isEmpty()) {
42         val mensagemErro = "Campos não preenchidos, tente novamente"
43         val snackbar = Snackbar.make(it, mensagemErro, Snackbar.LENGTH_LONG)
44         snackbar.show()
45     } else {
46         cadastrarUsuario(it)
47     }
48 }
49 }
```

```
50
51 fun cadastrarUsuario(it: View) {
52     val email = edit_email.text.toString().trim()
53     val senha = edit_senha.text.toString().trim()
54     FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, senha)
55         .addOnCompleteListener { task -> //task é o objeto do resultado vindo o firebase
56             if (task.isSuccessful) {
57                 salvarDadosUsuario()//metodo para salvar o cadastro
58                 val mensagemOk = "Cadastro realizado com sucesso"
59                 val snackbar = Snackbar.make(it, mensagemOk, Snackbar.LENGTH_LONG)
60                 snackbar.show()
61             } else {
62                 val mensagemErro = "Erro ao cadastrar usuário"
63                 val snackbar = Snackbar.make(it, mensagemErro, Snackbar.LENGTH_LONG)
64                 snackbar.show()
65             }
66         }
67     }
68 }
```



```
70 fun salvarDadosUsuario() {
71     val db = FirebaseFirestore.getInstance()
72     val nome = edit_nome.text.toString().trim()
73     val usuarioID = FirebaseAuth.getInstance().currentUser?.uid
74     val email = FirebaseAuth.getInstance().currentUser?.email
75     if (usuarioID != null && email != null) {
76         val usuarios = hashMapOf(
77             "nome" to nome,
78             "email" to email,
79             "uid" to usuarioID
80         )
81         db.collection(collectionPath: "Usuarios") CollectionReference
82             .add(usuarios) Task<DocumentReference!>
83             .addOnSuccessListener { documentReference -> // Add com sucesso
84                 println("Documento adicionado com ID: ${documentReference.id}")
85             }
86             .addOnFailureListener { e -> // Ocorreu um erro ao adicionar
87                 println("Erro ao adicionar documento: $e")
88             }
89     } else { // O usuário não está autenticado
90         println("Erro na autenticação")
91     }
92 }
93 }
```

Veja que estamos buscando os valores no banco e colocando na variável

Esses dados que foram buscamos do Auth irão para o Firestore.

Se já salvou no banco o nome do cadastro

Iremos voltar para nosso form_login.xml
E FormLogin.kt

Até a próxima!!!!

