



# Firestore

## Computação para Dispositivos Móveis

Prof. Gustavo Custodio  
[gustavo.custodio@ulife.com.br](mailto:gustavo.custodio@ulife.com.br)



firebase

# Firebase

- Firebase é um conjunto de ferramentas para criar **aplicações na nuvem**.
  - Bancos de dados;
  - Autenticação;
  - Hospedagem;
  - etc.

# Firebase

- O Firebase segue o padrão ***Backend as a Service (BaaS)***.
  - Não há necessidade de se preocupar com detalhes do *backend* quando cria sua aplicação.
  - Podemos focar apenas em aspetor do *frontend*.

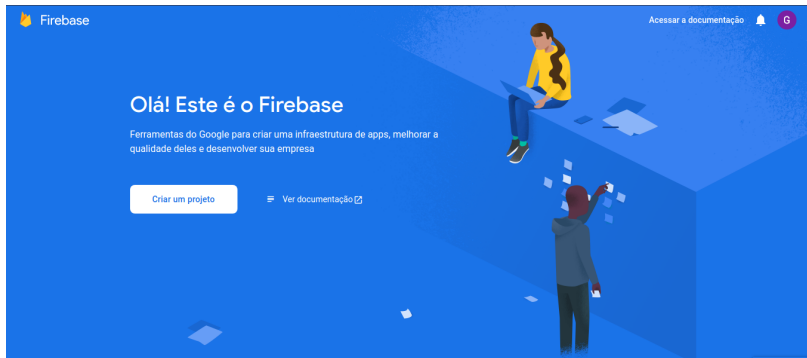


firebase

# Configurando um App com Firebase

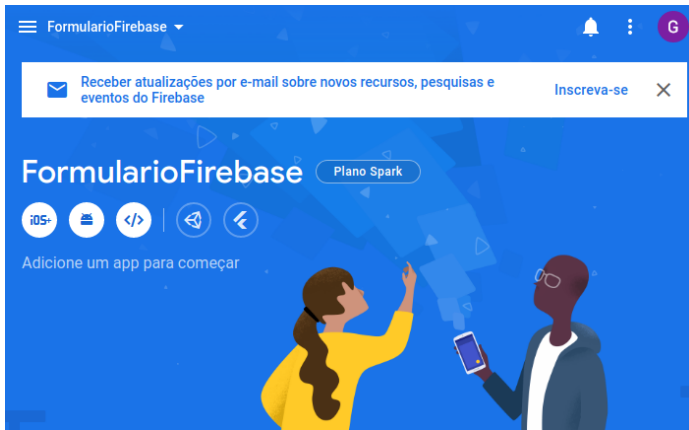
# Configurando

- Entre em <https://firebase.google.com>



# Configurando

- Crie um aplicativo chamado FormularioFirebase.



# Configurando

- Agora vamos adicionar um app para começar.
- O nome do pacote Android **deve ser único**.
  - Não use o mesmo nome do exemplo.
  - Mas siga o formato do nome em três partes separados por ponto:
    - Exemplo: `meu.app.android`



## Configurando

[illegible]

- Clique em registrar app.

# Configurando

- Baixe o arquivo **google\_services.json**.
  - Adicione esse arquivo para o diretório:
    - <meu\_projeto>/android/app.

# Configurando

- Adicione o trecho na pasta <meu\_projeto>/android.

```
1  dependencies {  
2      classpath 'com.google.gms:google-services:4.3.13'  
3      ...  
}
```

# Configurando

- Adicione o trecho na pasta <meu\_projeto>/android/app.

```
1  apply plugin: ...  
2  apply plugin: 'com.google.gms.google-services'  
3  ...
```

# Configurando

- Adicione o trecho na pasta <meu\_projeto>/android/app.

```
1  defaultConfig {  
2      ...  
3      // Coloque o id da sua aplicação aqui  
4      applicationId "com.dispomoveis.exemplo_flutter_firebase"  
5      // You can update the following values to match your ...  
6      ...
```

# Configurando

- Adicione o trecho na pasta <meu\_projeto>/android/app.

```
1 dependencies {  
2     implementation platform('com.google.firebase:firebase-bom:31.0.2')  
3     implementation 'com.google.firebase:firebase-analytics'  
4     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
5 }
```

## Configurando

- No arquivo `pubspec.yaml`, adicione as seguintes dependências:

```
1  dependencies:
2    flutter:
3      sdk: flutter
4    firebase_core: ^2.1.1
5    firebase_auth: ^4.1.1
6    cloud_firestore: ^4.0.4
```

# Configurando

- Altere o código do seu arquivo main.dart:

```
1  import 'package:firebase_core/firebase_core.dart';
2
3  Future main() async {
4    WidgetsFlutterBinding.ensureInitialized();
5    await Firebase.initializeApp();
6    runApp(const MyApp());
7  }
```

- Tente rodar o app.



## Configurando

- Se nenhum erro ocorreu, a conexão com o Firebase foi um sucesso!
- Reverta o método `main` para sua forma anterior.



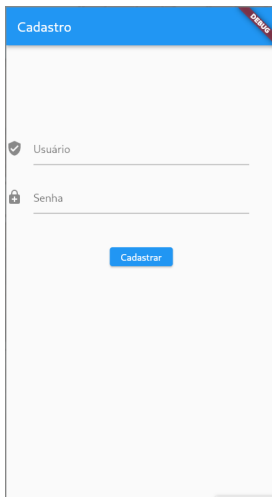
firebase

# Criando um App com Firebase

## Criando o App

- Vamos criar uma tela de cadastro de usuário.
  - Contendo usuário e senha.
  - Armazenaremos essas informações no Firebase.

# Criando o App



A mobile application registration screen titled 'Cadastro' in a blue header bar. A red diagonal banner in the top right corner reads 'DRAFT'. The screen features two input fields: 'Usuário' with a checkmark icon and 'Senha' with a lock icon. A blue 'Cadastrar' button is positioned below the fields. The bottom of the screen shows a standard mobile OS navigation bar.

Cadastro

Usuário

Senha

Cadastrar

## Criando o App

- Crie uma pasta chamada shared e dentro dela um arquivo `firebase_authentication.dart`.
- No topo do arquivo coloque:

```
1  import 'package:firebase_auth/firebase_auth.dart';  
2  import 'dart:async';
```

## Criando o App

- No mesmo arquivo crie a classe:

```
1  class FirebaseAuthentication {  
2      // Instância do firebase  
3      final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;  
4  }
```

## Criando o App

- Crie uma função para cadastrar um usuário.

```
1 Future<String?> createUser(String email, String password) async {  
2     try {  
3         UserCredential credential = await _firebaseAuth  
4             .createUserWithEmailAndPassword(email: email, password: password);  
5         if (credential.user == null) {  
6             return null;  
7         }  
8         return credential.user!.uid;  
9     } on FirebaseAuthException {  
10         return null;  
11     }  
12 }
```

## Criando o App

- Crie um arquivo chamado `cadastro.dart` contendo os seguintes *imports*:

```
1  import 'package:flutter/material.dart';  
2  import 'shared/firebase_authentication.dart';  
3  import 'package:firebase_core/firebase_core.dart';
```



## Criando o App

- Crie um StatefulWidget chamado **FormularioCadastro**.
  - Adicione dois Controllers na classe e um atributo mensagem.

```
1  class _FormularioCadastroState extends State<FormularioCadastro> {  
2  
3      final TextEditingController txtUserName = TextEditingController();  
4      final TextEditingController txtPassword = TextEditingController();  
5      String _mensagem = "";
```

## Criando o App

- Ainda dentro dessa classe, crie quatro métodos diferentes para construir Widgets.
  - `_criarTxtUserName(context);`
  - `_criarTxtSenha(context);`
  - `_criarBotaoCadastro(context);`
  - `_criarCampoMensagem(context).`

## Criando o App

```
1  Widget _criarTxtUserName(BuildContext context) {  
2    return Padding(  
3      padding: EdgeInsets.only(top: 128, right: 40),  
4      child: TextFormField(  
5        controller: txtUserName,  
6        keyboardType: TextInputType.emailAddress,  
7        decoration: InputDecoration(  
8          hintText: 'Usuário',  
9          icon: Icon(Icons.verified_user),  
10       ),  
11     ),  
12   );  
13 }
```

## Criando o App

```
1  Widget _criarTxtSenha(BuildContext context) {
2    return Padding(
3      padding: EdgeInsets.only(top: 24, right: 40),
4      child: TextFormField(
5        controller: txtPassword,
6        keyboardType: TextInputType.emailAddress,
7        obscureText: true,
8        decoration: InputDecoration(
9          hintText: 'Senha',
10         icon: Icon(Icons.enhanced_encryption)),
11      ),
12    );
13 }
```

## Criando o App

```
1  Widget _criarBotaoCadastro(BuildContext context) {  
2    return Padding(  
3      padding: EdgeInsets.only(top: 50),  
4      child: ElevatedButton(  
5        onPressed: (){},  
6        child: Text('Cadastrar'),  
7      ),  
8    );  
9  }
```

## Criando o App

```
1  Widget _criarCampoMensagem(BuildContext context) {  
2    return Text(  
3      _mensagem,  
4      style: TextStyle(  
5        fontSize: 16,  
6        color: Theme.of(context).primaryColorDark,  
7        fontWeight: FontWeight.bold),  
8    );  
9  }
```

## Criando o App

- Adicione esses widgets em uma coluna no método build

```
1  Widget build(BuildContext context) {  
2    return Scaffold(  
3      appBar: AppBar(  
4        title: Text("Cadastro"),  
5      ),  
6      body: Column(  
7        children: [  
8          _criarTxtUserName(context),  
9          _criarTxtSenha(context),  
10         _criarBotaoCadastro(context),  
11         _criarCampoMensagem(context),  
12       ],  
13     ),  
14   );  
15 }
```

## Criando o App

- No arquivo `main.dart` mude o método `build` para invocar o formulário:

```
1      @override
2      Widget build(BuildContext context) {
3          return MaterialApp(
4              title: 'Flutter Demo',
5              home: FormularioCadastro(),
6          );
7      }
```



## Criando o App

- No início do `FormularioCadastro` adicione mais um atributo:
  - `FirebaseAuthentication? auth;`
- Quando o app é iniciado, a inicialização do *firebase* deve ser realizada.
  - Adicionamos essa inicialização no método `initState`.

## Criando o App

```
1  @override
2  void initState() {
3      Firebase.initializeApp().whenComplete(() {
4          auth = FirebaseAuth();
5          setState(() {});
6      });
7      super.initState();
8  }
```

## Criando o App

- Agora adicione o código dentro do onPressed do botão criado.

```
1
2  onPressed: (){
3    String userId = "";
4    auth!.createUser(txtUserName.text, txtPassword.text).then((value) {
5      if (value == null) {
6        setState(() {
7          _mensagem = 'Erro de cadastro.';
8        });
9      }else {
10        userId = value;
11        setState(() {
12          _mensagem = 'Usuário $userId cadastrado.';
13        });
14      }
15    });
16  }
```

## Criando o App

- Essa função tenta cadastrar um novo usuário utilizando o serviço de autenticação do *firebase*.
  - Se o usuário é cadastrado, o campo de texto mostra que a operação foi um sucesso.
  - Se o cadastro falha, o usuário também recebe um *feedback*.

## Criando o App

- Adicione a função de *login* no arquivo `firebase_authentication.dart`.

```
1  Future<String?> login(String email, String password) async {  
2    try {  
3      UserCredential credential = await _firebaseAuth  
4        .signInWithEmailAndPassword(email: email, password: password);  
5      if (credential.user == null) {  
6        return null;  
7      }  
8      return credential.user!.uid;  
9    } on FirebaseAuthException {  
10     return null;  
11   }  
12 }
```

## Criando o App

- Adicione um botão de login ao formulário, utilizando a função de login criada.

## Resumo

- Configuramos o *firebase*.
- Utilizamos o *firebase* para criar um usuário.
- Autenticamos o usuário criado com o *firebase*.

Obrigado

[gustavo.custodio@ulife.com.br](mailto:gustavo.custodio@ulife.com.br)