

Vamos substituir nosso arquivo “registro_tela.dart”

```
registro_tela.dart lib\telas

1  import 'package:flutter/material.dart';
2
3
4  class RegistroTela extends StatefulWidget {
5    const RegistroTela({super.key});
6
7    @override
8    State<RegistroTela> createState() => _RegistroTelaState();
9  }
10
11  class _RegistroTelaState extends State<RegistroTela> {
12    // Agora os controladores ficam dentro da classe State, onde podem ser gerenciados
13    final TextEditingController _nomeController = TextEditingController();
14    final TextEditingController _emailController = TextEditingController();
15    final TextEditingController _senhaController = TextEditingController();
16    final TextEditingController _confirmaSenhaController = TextEditingController();
17
18    @override
19    void dispose() {
20      _nomeController.dispose();
21      _emailController.dispose();
22      _senhaController.dispose();
23      _confirmaSenhaController.dispose();
24      super.dispose();
25    }
26
27    @override
28    Widget build(BuildContext context) {
29      return Scaffold(
30        body: Container(
31          color: Colors.blue,
32          padding: EdgeInsets.all(16),
33          child: Center(
34            child: Column(
35              mainAxisAlignment: MainAxisAlignment.center,
36              children: [
37                Container(
38                  padding: EdgeInsets.all(16),
39                  decoration: BoxDecoration(
40                    color: Colors.white,
41                    borderRadius: BorderRadius.circular(16),
42                  ), // BoxDecoration
43                child: Column(
44                  children: [
45                    FlutterLogo(size: 76),
46                    SizedBox(height: 16),
47                    TextField(
48                      controller: _nomeController,
49                      decoration: InputDecoration(hintText: 'Nome: '),
50                    ), // TextField
51
```

```

51
52     SizedBox(height: 16),
53     TextField(
54       controller: _emailController,
55       decoration: InputDecoration(hintText: 'E-mail '),
56     ), // TextField
57     SizedBox(height: 16),
58     TextField(
59       obscureText: true,
60       controller: _senhaController,
61       decoration: InputDecoration(
62         hintText: 'Digite sua senha ',
63       ), // InputDecoration
64     ), // TextField
65     SizedBox(height: 16),
66     TextField(
67       obscureText: true,
68       controller: _confirmaSenhaController,
69       decoration: InputDecoration(
70         hintText: 'Confirme sua senha: ',
71       ), // InputDecoration
72     ), // TextField
73     SizedBox(height: 16),
74     ElevatedButton(
75       onPressed: () {},
76       child: const Text('Cadastrar'),
77     ), // ElevatedButton
78   ],
79 ), // Column
80 ), // Container
81 ],
82 ), // Column
83 ), // Center
84 ), // Container
85 ); // Scaffold
86 }
87 }

```

Agora vamos o banco de dados

1 –Baixe o Node.JS no site www.evandrojv.com.br → Downloads

Descompacte e instale.

```
npm install firebase
```

```
dart pub global activate flutterfire_cli
```

Vamos acessar

<https://console.firebase.google.com/>

Faça o login com sua conta Google.

Olá, Professor

É ótimo ter você no Firebase!

Vamos começar



Comece configurando um Projeto do Firebase

Integre produtos do Firebase para turbinar seu app

Testar um app de exemplo



Teste um app de planejamento de viagens com tecnologia de IA

Implante um app de exemplo que use o Firestore, o Firebase Authentication e a entrada multimodal no Firebase AI Logic. Depois disso, estude o código no Firebase Studio.



Criar um app do Flutter com tecnologia de IA

Implante um app de exemplo que mostre como a API Gemini Live, os comandos multimodais e a criação de imagens com o Nano Banana funcionam no Flutter

× Criar um projeto

Vamos começar nomeando o projeto[®]

Nome do projeto

controlehora

controlehora-6d5d3

☒ Participe do [Programa para desenvolvedores do Google](#) para aperfeiçoar sua jornada com assistência de IA, recursos de aprendizado, selos de perfil e muito mais.

Já tem um projeto do Google Cloud?
[Adicionar o Firebase ao projeto do Google Cloud](#)

Continuar

× Criar um projeto

O Google Analytics é uma solução de análise ilimitada e gratuita. Com ele, é possível segmentar, gerar relatórios e muito mais nos seguintes produtos: Firebase Crashlytics, Cloud Messaging, Mensagens no app, Configuração remota, Teste A/B e Cloud Functions.

O Google Analytics ativa:

-  Teste A/B 
-  Segmentação de usuários em produtos do Firebase 
-  Registros de navegação estrutural no Crashlytics 
-  Gatilhos do Cloud Functions com base em eventos 
-  Geração de relatórios ilimitada gratuita 

☒ Ativar o Google Analytics neste projeto
Recomendado

[Anterior](#)

[Continuar](#)

× Criar um projeto

Configurar o Google Analytics

Escolha ou crie uma conta do Google Analytics 

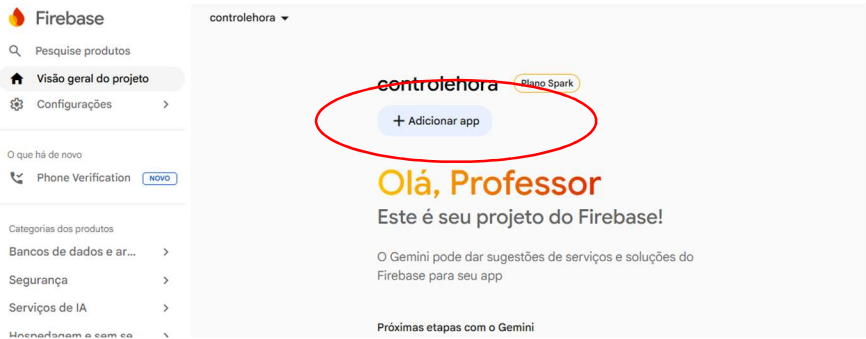
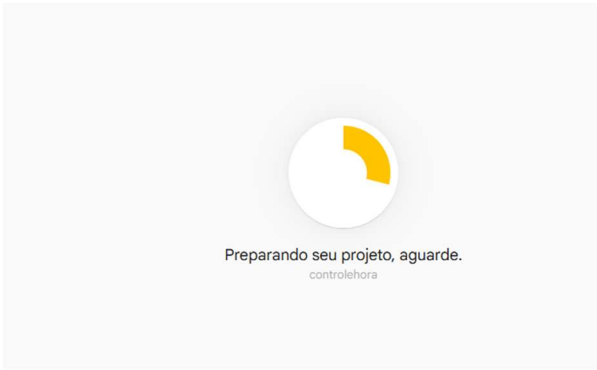
 Default Account for Firebase 

Crie automaticamente uma nova propriedade nesta conta 

Após a criação do projeto, uma nova propriedade do Google Analytics será criada na conta selecionada e vinculada ao seu projeto do Firebase. Esse processo vai permitir o fluxo de dados entre os produtos. Os dados da propriedade do Google Analytics exportados para o Firebase ficam sujeitos aos Termos de Serviço do Firebase, e os dados do Firebase importados para o Google Analytics ficam sujeitos aos Termos de Serviço do Google Analytics. [Saiba mais](#) .

[Anterior](#)

[Criar projeto](#)



× Adicionar o Firebase ao seu app do Flutter

1 Preparar seu espaço de trabalho

2 Instalar e executar a CLI do FlutterFire

Em qualquer diretório, execute o comando:

```
$ dart pub global activate flutterfire_cli
```

Em seguida, na raiz do diretório do seu projeto do Flutter, execute o comando:

```
$ flutterfire configure --project=controlehora-9046f
```

Com isso, seus apps são registrados automaticamente por plataforma com o Firebase, e um arquivo de configuração `lib/firebase_options.dart` é adicionado ao seu projeto do Flutter.

Anterior

Avançar



Pesquise produtos

Visão geral do projeto

Configurações

O que há de novo

Phone Verification **NOVO**

Atalhos do projeto

Authentication

Categorias dos produtos

Bancos de dados e ar...

Segurança

Serviços de IA

Hospedagem e sem se...

DevOps e engajamento

Analytics

controlehora

Analytics

Usuários ativos por dia

Retenção no

Sem dados
for the last 14 days

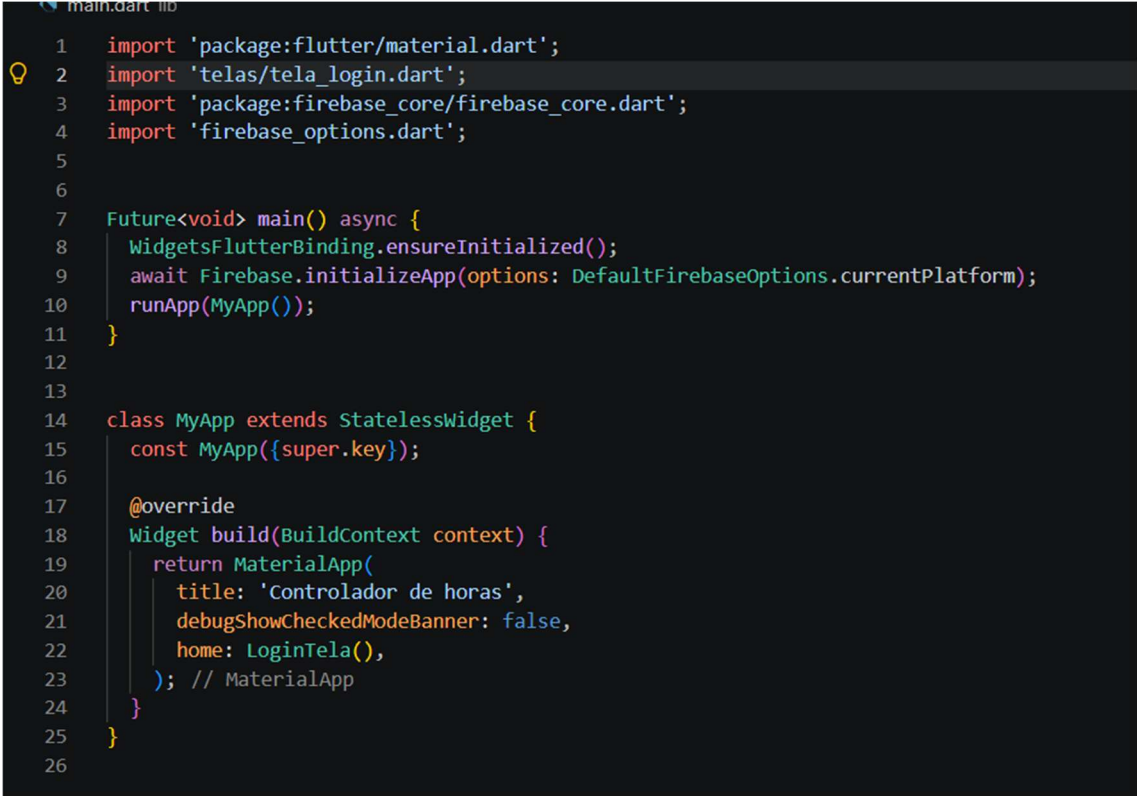
controla_horas (android)	DAU	0
controla_horas (ios)	DAU	0
controla_horas (web)	DAU	0

Vamos abrir o projeto e vamos trabalhar no “TERMINAL”

```
firebase login
```

```
flutter pub add firebase_core
```

Vamos abri o “main.dart” e fazer as seguintes configurações



```
main.dart lib
1  import 'package:flutter/material.dart';
2  import 'telas/tela_login.dart';
3  import 'package:firebase_core/firebase_core.dart';
4  import 'firebase_options.dart';
5
6
7  Future<void> main() async {
8    WidgetsFlutterBinding.ensureInitialized();
9    await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
10   runApp(MyApp());
11 }
12
13
14 class MyApp extends StatelessWidget {
15   const MyApp({super.key});
16
17   @override
18   Widget build(BuildContext context) {
19     return MaterialApp(
20       title: 'Controlador de horas',
21       debugShowCheckedModeBanner: false,
22       home: LoginTela(),
23     ); // MaterialApp
24   }
25 }
26
```

```
flutter pub add firebase_messaging
```

```
flutter pub add firebase_auth
```

```
flutter pub add cloud_firestore
```

Vamos criar uma pastar chamada “servicos” dentro da lib.

Dentro crie o arquivo “conexoes_firebase.dart”

```
conexoes_firebase.dart no services
1  import 'package:firebase_auth/firebase_auth.dart';
2
3  class Conexao{
4
5      final FirebaseAuth _firebaseConexao = FirebaseAuth.instance;
6
7      Future<String?> cadastrarUsuario({
8          required String email,
9          required String senha,
10         required String nome,
11     }) async {
12         try {
13             UserCredential userCredential = await _firebaseConexao.createUserWithEmailAndPassword(email: email, password: senha);
14             await userCredential.user!.updateDisplayName(nome);
15         } on FirebaseAuthException catch (e) {
16             switch (e.code) {
17                 case 'email-already-in-use':
18                     return 'O email já está em uso.';
19             }
20
21             return e.code;
22         }
23
24         return null;
25     }
26 }
27
```

Vamos mudar nosso arquivo “registro_tela.dart”, observar o que está selecionado para fazer as alterações.

```
class _RegistroTelaState extends State<RegistroTela> {
    // Agora os controladores ficam dentro da classe State, onde podem ser gerenciados
    final TextEditingController _nomeController = TextEditingController();
    final TextEditingController _emailController = TextEditingController();
    final TextEditingController _senhaController = TextEditingController();
    final TextEditingController _confirmaSenhaController = TextEditingController();

    Conexao conexao = Conexao();
}
```

```

    SizedBox(height: 16),
    ElevatedButton(
      onPressed: () {
        if (_senhaController.text ==
            _confirmaSenhaController.text) {
          conexao.cadastrarUsuario(
            nome: _nomeController.text,
            email: _emailController.text,
            senha: _senhaController.text,
          )
            .then((String? erro) {
              if (!context.mounted) return;

              if (erro != null) {
                final snackBar = SnackBar(
                  content: Text(erro),
                  backgroundColor: Colors.red,
                ); // SnackBar
                ScaffoldMessenger.of(
                  context,
                ).showSnackBar(snackBar);
              } else {
                Navigator.pop(context);
              }
            });
        } else {
          const snackBar = SnackBar(
            content: Text('As senhas não correspondem'),
            backgroundColor: Colors.red,
          ); // SnackBar

          ScaffoldMessenger.of(context).showSnackBar(snackBar);
        }
      },
      child: const Text('Cadastrar'),
    ), // ElevatedButton
  ),

```

Vamos mexer no arquivo “conexoes_firebase.dart”

```

27   Future<String?> entrarUsuario(
28     {required String email, required String senha}) async {
29     try {
30       await _firebaseConexao.signInWithEmailAndPassword(
31         email: email, password: senha);
32     } on FirebaseAuthException catch (e) {
33       switch (e.code) {
34         case 'user-not-found':
35           return 'Usuário não encontrado';
36         case 'wrong-password':
37           return 'Senha incorreta';
38       }
39       return e.code;
40     }
41     return null;
42   }
43 }
44

```

Agora vamos no arquivo "tela_login.dart"

```
    SizedBox(height: 16),
    ElevatedButton(onPressed: () async {
      String? resultado = await conexao.entrarUsuario(
        email: _emailController.text,
        senha: _senhaController.text
      );
      if (resultado == null) {
        print("Login realizado com sucesso!");
      } else {
        if (!context.mounted) return; // Verificação de segurança obrigatória
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text(resultado)),
        );
      }
    },
    child: Text( 'Entrar' )), // ElevatedButton
  ],
);
```