

Adicionar o Firebase ao seu app Flutter



iOS+

✓ Android

Web

Pré-requisitos

- Instale o editor ou o ambiente de desenvolvimento integrado ([//flutter.io/get-started/editor/](https://flutter.io/get-started/editor/)) de sua preferência.
- Configure um dispositivo ou emulador para executar seu app. Os emuladores (<https://developer.android.com/studio/run/managing-avds?hl=pt-br>) precisam usar uma imagem de emulador no Google Play.
- Verifique se o app atende aos seguintes requisitos:
 - Visa o nível 21 da API (Lollipop) ou mais recente.
 - Usa o Android 5.0 ou versões mais recentes
- Instale o Flutter ([//flutter.io/get-started/install/](https://flutter.io/get-started/install/)) no seu sistema operacional específico, incluindo:
 - SDK do Flutter
 - Bibliotecas de suporte
 - Software e SDKs específicos da plataforma
- Faça login no Firebase ([//console.firebase.google.com/?hl=pt-br](https://console.firebase.google.com/?hl=pt-br)) usando sua Conta do Google.

Add Firebase to your Flutte...



Se você ainda não tem um app Flutter, conclua os Primeiros passos: test drive ([//flutter.io/get-started/test-drive/#androidstudio](https://flutter.io/get-started/test-drive/#androidstudio)) para criar um novo app Flutter usando o editor ou ambiente de desenvolvimento integrado de sua preferência.

Etapa 1: instalar as ferramentas de linha de comando necessárias

1. Caso ainda não tenha feito isso, instale a CLI Firebase (https://firebase.google.com/docs/cli?hl=pt-br#setup_update_cli).
2. Faça login no Firebase com sua Conta do Google executando o seguinte comando:

```
$ firebase login
```

3. Instale a CLI do FlutterFire executando o seguinte comando em qualquer diretório:

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

Etapa 2: configurar os apps para usar o Firebase

Use a CLI do FlutterFire para configurar seus apps do Flutter e se conectar ao Firebase.

No diretório do projeto do Flutter, execute o seguinte comando para iniciar o fluxo de trabalho de configuração do app:

```
your-flutter-proj$ flutterfire configure
```

+ O que o fluxo de trabalho `flutterfire configure` faz?

O fluxo de trabalho `flutterfire configure` faz o seguinte:

- Solicita que você selecione as plataformas (iOS, Android, Web) compatíveis com o app do Flutter. Para cada plataforma selecionada, a CLI do FlutterFire cria um novo app do Firebase no seu projeto do Firebase.

Você pode usar um projeto do Firebase ou criar um novo. Se você já tiver apps registrados em um projeto do Firebase, a CLI do FlutterFire vai tentar fazer a correspondência com base na configuração atual do projeto do Flutter.

★ **Observação:** veja algumas dicas sobre como configurar e gerenciar seu projeto do Firebase:

- Confira nossas práticas recomendadas (<https://firebase.google.com/docs/projects/dev-workflows/general-best-practices?hl=pt-br>) para adicionar apps a um projeto do Firebase, incluindo como lidar com diversas variantes.
 - Ative o Google Analytics (<https://support.google.com/firebase/answer/9289399?hl=pt-br#linkga>) no projeto. Isso possibilita uma experiência ideal usando vários produtos do Firebase, como Crashlytics e Remote Config.
- Crie um arquivo de configuração do Firebase (`firebase_options.dart`) e adicione ele ao diretório `lib/`.

★ **Observação:** esse arquivo de configuração do Firebase contém identificadores exclusivos, mas não secretos, para cada plataforma selecionada.

Acesse Noções básicas sobre projetos do Firebase

(<https://firebase.google.com/docs/projects/learn-more?hl=pt-br#config-files-objects>) para saber mais sobre esse arquivo de configuração.

- *(para Crashlytics ou Performance Monitoring no Android)* Adiciona o necessário Plug-ins do Gradle específicos do produto ao app Flutter.

★ **Observação:** para que a CLI do FlutterFire adicione o plug-in apropriado do Gradle, o plug-in do Flutter do produto já precisa estar importado no app do Flutter.

Depois dessa primeira execução de **flutterfire configure**, execute o comando novamente sempre que você:

- começar a oferecer suporte a uma nova plataforma no seu app do Flutter;
- Comece a usar um novo serviço ou produto do Firebase no seu app Flutter. especialmente se você começa a usar o recurso Fazer login com o Google, Crashlytics, Performance Monitoring ou Realtime Database.

Executar o comando novamente garante que a configuração do Firebase do seu app do Flutter esteja atualizada e, para Android, adiciona automaticamente todos os plug-ins do Gradle necessários ao app.

Etapas 3: inicializar o Firebase no app

1. No diretório do projeto do Flutter, execute o seguinte comando para instalar o plug-in principal:

```
your-flutter-proj$ flutter pub add firebase_core
```

2. No diretório do projeto do Flutter, execute o seguinte comando para garantir que a configuração do app criado com o Flutter do Firebase esteja atualizada:

```
your-flutter-proj$ flutterfire configure
```

3. No arquivo `lib/main.dart`, importe o plug-in principal do Firebase e o arquivo de configuração gerado antes:

```
import 'package:firebase_core/firebase_core.dart';  
import 'firebase_options.dart';
```

4. Ainda no arquivo `lib/main.dart`, inicialize o Firebase usando o objeto `DefaultFirebaseOptions` exportado pelo arquivo de configuração:

```
WidgetsFlutterBinding.ensureInitialized();  
await Firebase.initializeApp(  
  options: DefaultFirebaseOptions.currentPlatform,  
);  
runApp(const MyApp());
```

5. Recrie o aplicativo do Flutter:

```
your-flutter-proj$ flutter run
```

Se preferir usar um projeto de demonstração, inicie o Emulador do Firebase (<https://firebase.google.com/docs/emulator-suite?hl=pt-br>) e no arquivo `lib/main.dart`, inicialize o Firebase usando `demoProjectId` (deve começar com `demo-`):

```
await Firebase.initializeApp(  
  demoProjectId: "demo-project-id",  
);
```

Etapa 4: adicionar plug-ins do Firebase

É possível acessar o Firebase no app do Flutter usando os vários plug-ins do Firebase para o Flutter (`#available-plugins`), um para cada produto do Firebase, como Cloud Firestore, Authentication, Analytics etc.

Como o Flutter é um framework multiplataforma, cada plug-in do Firebase é aplicável a plataformas da Apple, do Android e da Web. Por isso, qualquer plug-in do Firebase adicionado ao seu app do Flutter será usado pelas versões para Apple, Android e Web do seu app.

Veja como adicionar um plug-in do Firebase ao Flutter:

1. No diretório do projeto do Flutter, execute o seguinte comando:

```
your-flutter-proj$ flutter pub add PLUGIN_NAME 
```

2. No diretório do projeto do Flutter, execute o seguinte comando:

```
your-flutter-proj$ flutterfire configure
```

A execução desse comando garante que a configuração do Firebase do seu app criado com o Flutter esteja atualizada e, para o Crashlytics e o Performance Monitoring no Android, adicione os plug-ins necessários do Gradle ao app.

3. Após a conclusão, recrie seu projeto do Flutter:

```
your-flutter-proj$ flutter run
```

Está tudo pronto! Agora seus apps do Flutter estão registrados e configurados para usar o Firebase.

Plug-ins disponíveis

Produto	Nome do plug-in	iOS	Android	Web	Outra da Apple (como o macOS)	Windows
<u>Analytics</u> (https://firebase.google.com/docs/analytics/get-started?platform=flutter&hl=pt-br)	firebase_analytics	✓	✓	✓	beta	
<u>App Check</u> (https://firebase.google.com/docs/app-check/flutter/default-providers?hl=pt-br)	firebase_app_check	✓	✓	✓	beta	
<u>Authentication</u> (https://firebase.google.com/docs/auth/flutter/start?hl=pt-br)	firebase_auth	✓	✓	✓	beta	b
<u>Cloud Firestore</u> (https://firebase.google.com/docs/firestore/quickstart?hl=pt-br)	cloud_firestore	✓	✓	✓	beta	b
<u>Cloud Functions</u> (https://firebase.google.com/docs/functions/get-started?hl=pt-br)	cloud_functions	✓	✓	✓	beta	
<u>Cloud Messaging</u> (https://firebase.google.com/docs/cloud-messaging/flutter/client?hl=pt-br)	firebase_messaging	✓	✓	✓	beta	
<u>Cloud Storage</u> (https://firebase.google.com/docs/storage/flutter/start?hl=pt-br)	firebase_storage	✓	✓	✓	beta	b

Produto	Nome do plug-in	iOS	Android	Web	Outra da Apple (como o macOS)	Windows
<u>Crashlytics</u> (https://firebase.google.com/docs/crashlytics/get-started?platform=flutter&hl=pt-br)	firebase_crashlytics	✓	✓		beta	
<u>Data Connect</u> (https://firebase.google.com/docs/data-connect/flutter-sdk?hl=pt-br)	firebase_data_connect	✓	✓	✓		
<u>Dynamic Links</u> (https://firebase.google.com/docs/dynamic-links/flutter/create?hl=pt-br)	firebase_dynamic_links	✓	✓			
<u>In-App Messaging</u> (https://firebase.google.com/docs/in-app-messaging/get-started?platform=flutter&hl=pt-br)	firebase_in_app_messaging	✓	✓			
<u>Firebase Instalações</u> (https://firebase.google.com/docs/projects/manage-installations?hl=pt-br)	firebase_app_installations	✓	✓	✓	beta	
<u>Download de modelos de ML</u> (https://firebase.google.com/docs/ml/flutter/use-custom-models?hl=pt-br)	firebase_ml_model_downloader	✓	✓		beta	
<u>Performance Monitoring</u> (https://firebase.google.com/docs/perf-mon/flutter/get-started?hl=pt-br)	firebase_performance	✓	✓	✓		
<u>Realtime Database</u> (https://firebase.google.com/docs/database/flutter/start?hl=pt-br)	firebase_database	✓	✓	✓	beta	
<u>Remote Config</u> (https://firebase.google.com/docs/remote-config/get-started?platform=flutter&hl=pt-br)	firebase_remote_config	✓	✓	✓	beta	
<u>Vertex AI in Firebase</u> (https://firebase.google.com/docs/vertex-ai/get-started?platform=flutter&hl=pt-br)	firebase_vertexai	✓	✓	✓	beta	

Cuidado: o Firebase no Windows não se destina a casos de uso de produção, apenas fluxos de trabalho de desenvolvimento local.

Testar um aplicativo de exemplo com o Analytics

Como todos os pacotes, o plug-in `firebase_analytics` vem com um programa de exemplo

([//github.com/firebase/flutterfire/tree/master/packages/firebase_analytics/firebase_analytics/example](https://github.com/firebase/flutterfire/tree/master/packages/firebase_analytics/firebase_analytics/example))

1. Abra um app Flutter que você já configurou para usar o Firebase. Veja instruções nesta página.

2. Acesse o diretório `lib` do aplicativo e exclua o arquivo `main.dart` atual.

3. No repositório de programas de exemplo

([//github.com/firebase/flutterfire/tree/master/packages/firebase_analytics/firebase_analytics/example/lib](https://github.com/firebase/flutterfire/tree/master/packages/firebase_analytics/firebase_analytics/example/lib))

do Google Analytics, copie e cole os dois arquivos abaixo no diretório `lib` do seu app:

- `main.dart`
- `tabs_page.dart`

4. Execute seu app Flutter.

5. Acesse o projeto do Firebase do seu app no console do Firebase

([//console.firebase.google.com/?hl=pt-br](https://console.firebase.google.com/?hl=pt-br)) e clique em **Analytics** no menu de navegação à esquerda.

a. Clique em **Painel** ([//support.google.com/firebase/answer/6317517?hl=pt-br](https://support.google.com/firebase/answer/6317517?hl=pt-br)). Se o Analytics estiver funcionando corretamente, o painel mostrará um usuário ativo no painel "Usuários ativos nos últimos 30 minutos". Pode levar algum tempo para que esse painel seja preenchido.

b. Clique em **DebugView** (<https://firebase.google.com/docs/analytics/debugview?hl=pt-br>)
. Ative o recurso para ver todos os eventos gerados pelo programa de exemplo.

Saiba mais sobre como configurar o Analytics nos guias para iniciantes para **iOS+**

(<https://firebase.google.com/docs/analytics/get-started?platform=ios&hl=pt-br>), **Android**

(<https://firebase.google.com/docs/analytics/get-started?platform=android&hl=pt-br>) e **Web**

(<https://firebase.google.com/docs/analytics/get-started?platform=web&hl=pt-br>).

Próximas etapas

- Aprenda a usar o Firebase Flutter Codelab (<https://firebase.google.com/codelabs/firebase-get-to-know-flutter?hl=pt-br>) na prática.
- Prepare-se para lançar seu aplicativo:
 - Configure os alertas de orçamento (<https://firebase.google.com/docs/projects/billing/avoid-surprise-bills?hl=pt-br#set-up-budget-alert-emails>) para seu projeto no console Google Cloud.
 - Monitore o painel *Uso e faturamento* ([//console.firebase.google.com/project/_/usage?hl=pt-br](https://console.firebase.google.com/project/_/usage?hl=pt-br)) no console do Firebase para ter uma visão geral do uso do projeto em vários serviços do Firebase.
 - Consulte a checklist de lançamento do Firebase (<https://firebase.google.com/support/guides/launch-checklist?hl=pt-br>).

Exceto em caso de indicação contrária, o conteúdo desta página é licenciado de acordo com a Licença de atribuição 4.0 do Creative Commons (<https://creativecommons.org/licenses/by/4.0/>), e as amostras de código são licenciadas de acordo com a Licença Apache 2.0 (<https://www.apache.org/licenses/LICENSE-2.0>). Para mais detalhes, consulte as políticas do site do Google Developers (<https://developers.google.com/site-policies?hl=pt-br>). Java é uma marca registrada da Oracle e/ou afiliadas.

Última atualização 2025-04-09 UTC.