

INSTALAÇÃO NO WINDOWS

Sítio: <https://docs.flutter.dev/get-started/install/windows#system-requirements>

Requisitos de sistema

Para instalar e executar o Flutter, seu ambiente de desenvolvimento deve atender a estes requisitos mínimos:

- **Sistemas operacionais** : Windows 10 ou posterior (64 bits), baseado em x86-64.
- **Espaço em disco** : 1,64 GB (não inclui espaço em disco para IDE/ferramentas).
- **Ferramentas** : O Flutter depende da disponibilidade dessas ferramentas em seu ambiente.
 - [Windows PowerShell 5.0](#) ou mais recente (este é pré-instalado com o Windows 10)
 - [Git para Windows](#) 2.x, com a opção **Usar Git no Prompt de Comando do Windows** .

Se o Git para Windows já estiver instalado, verifique se você pode executar `git` comandos no prompt de comando ou no PowerShell.

Obtenha o SDK Flutter

1. Baixe o seguinte pacote de instalação para obter a versão estável mais recente do Flutter SDK:

[flutter_windows_3.0.5-stable.zip](#)

Para outros canais de lançamento e compilações mais antigas, consulte a página de [lançamentos do SDK](#) .

2. Extraia o arquivo zip e coloque o conteúdo `flutter` no local de instalação desejado para o Flutter SDK (por exemplo, `C:\src\flutter`).

Aviso: Não instale o Flutter em um caminho que contenha caracteres especiais ou espaços.

Aviso: Não instale o Flutter em um diretório `C:\Program Files\` que exija privilégios elevados.

Se você não quiser instalar uma versão fixa do pacote de instalação, pule as etapas 1 e 2. Em vez disso, obtenha o código-fonte do [repositório Flutter](#) no GitHub e altere branches ou tags conforme necessário. Por exemplo:

content_copy

```
C:\src>git clone https://github.com/flutter/flutter.git -b stable
```

Agora você está pronto para executar comandos Flutter no Console Flutter.

Atualize seu caminho

Se você deseja executar comandos do Flutter no console normal do Windows, siga estas etapas para adicionar o Flutter à `PATH` variável de ambiente:

- Na barra de pesquisa Iniciar, insira 'env' e selecione **Editar variáveis de ambiente para sua conta**.
- Em **variáveis do usuário**, verifique se há uma entrada chamada **Path**:
 - Se a entrada existir, anexe o caminho completo para `flutter\bin` usando `;` como separador dos valores existentes.
 - Se a entrada não existir, crie uma nova variável de usuário nomeada `Path` com o caminho completo para `flutter\bin` como seu valor.

Você precisa fechar e reabrir todas as janelas de console existentes para que essas alterações entrem em vigor.

Observação: a partir da versão dev do Flutter 1.19.0, o SDK do Flutter contém o `dart` comando junto com o `flutter` comando para que você possa executar programas de linha de comando Dart com mais facilidade. Baixar o SDK do Flutter também baixa a versão compatível do Dart, mas se você baixou o SDK do Dart separadamente, certifique-se de que a versão do Flutter `dart` seja a primeira no seu caminho, pois as duas versões podem não ser compatíveis. O comando a seguir informa se os comandos `flutter` e `dart` originam do mesmo `bin` diretório e, portanto, são compatíveis.

content_copy

```
C:\>where flutter dart
C:\path-to-flutter-sdk\bin\flutter
C:\path-to-flutter-sdk\bin\flutter.bat
C:\path-to-dart-sdk\bin\dart.exe           :: this should go after
C:\path-to-flutter-sdk\bin\` commands
C:\path-to-flutter-sdk\bin\dart
C:\path-to-flutter-sdk\bin\dart.bat
```

Como mostrado acima, o comando `dart` do SDK Flutter não vem primeiro. Atualize seu caminho para usar comandos de `C:\path-to-flutter-sdk\bin\comandos` anteriores `C:\path-to-dart-sdk\bin\` (neste caso). Depois de reiniciar seu shell para que a alteração tenha efeito, executar o `where` comando novamente deve mostrar que os comandos `flutter` e `dart` do mesmo diretório agora vêm primeiro.

content_copy

```
C:\>where flutter dart
C:\dev\src\flutter\bin\flutter
C:\dev\src\flutter\bin\flutter.bat
C:\dev\src\flutter\bin\dart
C:\dev\src\flutter\bin\dart.bat
C:\dev\src\dart-sdk\bin\dart.exe
```

No entanto, se você estiver usando `PowerShell`, nele `where` é um alias de `Where-Object` comando, então você precisa usar `where.exe` em vez disso.

content_copy

```
PS C:\> where.exe flutter dart
```

Para saber mais sobre o `dart` comando, execute `dart -h` a partir da linha de comando ou consulte a página da [ferramenta de dardo](#).

Corre `flutter doctor`

Em uma janela de console que tenha o diretório Flutter no caminho (veja acima), execute o seguinte comando para ver se há alguma dependência de plataforma necessária para concluir a configuração:

content_copy

```
C:\src\flutter>flutter doctor
```

Este comando verifica seu ambiente e exibe um relatório do status de sua instalação do Flutter. Verifique a saída com cuidado para outros softwares que você possa precisar instalar ou outras tarefas a serem executadas (mostradas em **negrito**).

Por exemplo:

content_copy

[–] Cadeia de ferramentas Android – desenvolver para dispositivos Android

- SDK do Android em D:\Android\sdk

X Android SDK não possui ferramentas de linha de comando; baixe em <https://goo.gl/XxQghQ>

- Tente reinstalar ou atualizar seu Android SDK, visite <https://docs.flutter.dev/setup/#android-setup> para obter instruções detalhadas.

As seções a seguir descrevem como executar essas tarefas e concluir o processo de configuração. Depois de instalar as dependências ausentes, você pode executar o `flutter doctor` comando novamente para verificar se configurou tudo corretamente.

Observação: se `flutter doctor` retornar que o plug-in Flutter ou o plug-in Dart do Android Studio não está instalado, vá para [Configurar um editor](#) para resolver esse problema.

Aviso: a ferramenta Flutter pode ocasionalmente baixar recursos dos servidores do Google. Ao baixar ou usar o SDK do Flutter, você concorda com os [Termos de Serviço do Google](#).

Por exemplo, quando instalado a partir do GitHub (ao contrário de um arquivo pré-empacotado), a ferramenta Flutter fará o download do SDK do Dart dos servidores do Google imediatamente na primeira execução, pois é usado para executar a `flutter` própria ferramenta. Isso também ocorrerá quando o Flutter for atualizado (por exemplo, executando o `flutter upgrade` comando).

A `flutter` ferramenta usa o Google Analytics para relatar estatísticas de uso de recursos e enviar [relatórios de falhas](#). Esses dados são usados para ajudar a melhorar as ferramentas Flutter ao longo do tempo.

A análise da ferramenta Flutter não é enviada na primeira execução. Para desabilitar o relatório, execute `flutter config --no-analytics`. Para exibir a configuração atual, use `flutter config`. Se você optar por não receber análises, um evento de desativação será enviado e nenhuma informação adicional será enviada pela ferramenta Flutter.

As ferramentas Dart também podem enviar métricas de uso e relatórios de falhas para o Google. Para controlar o envio dessas métricas, use as seguintes opções na [dart ferramenta](#):

- `--enable-analytics`: permite análises anônimas.
- `--disable-analytics`: Desativa a análise anônima.

A [Política de Privacidade do](#) Google descreve como os dados são tratados por esses serviços.

Configuração do Android

Observação: o Flutter depende de uma instalação completa do Android Studio para fornecer suas dependências da plataforma Android. No entanto, você pode escrever seus aplicativos Flutter em vários editores; uma etapa posterior discute isso.

Instale o Android Studio

1. Baixe e instale [o Android Studio](#) .
2. Inicie o Android Studio e vá até o 'Assistente de configuração do Android Studio'. Isso instala o Android SDK mais recente, as ferramentas de linha de comando do Android SDK e as ferramentas de compilação do Android SDK, que são exigidas pelo Flutter ao desenvolver para Android.
3. Execute `flutter doctor` para confirmar que o Flutter localizou sua instalação do Android Studio. Se o Flutter não conseguir localizá-lo, execute `flutter config --android-studio-dir <directory>` para definir o diretório no qual o Android Studio está instalado.

Configure seu dispositivo Android

Para se preparar para executar e testar seu aplicativo Flutter em um dispositivo Android, você precisa de um dispositivo Android executando o Android 4.1 (API de nível 16) ou superior.

1. Ative **as opções do desenvolvedor** e a **depuração USB** no seu dispositivo. Instruções detalhadas estão disponíveis na [documentação do Android](#) .
2. Somente Windows: instale o [driver USB do Google](#) .
3. Usando um cabo USB, conecte seu telefone ao computador. Se solicitado em seu dispositivo, autorize seu computador a acessar seu dispositivo.
4. No terminal, execute o `flutter devices` comando para verificar se o Flutter reconhece seu dispositivo Android conectado. Por padrão, o Flutter usa a versão do Android SDK em que sua `adb` ferramenta está baseada. Se você quiser que o Flutter use uma instalação diferente do Android SDK, você deve definir a `ANDROID_SDK_ROOT` variável de ambiente para esse diretório de instalação.

Configurar o emulador do Android

Para se preparar para executar e testar seu aplicativo Flutter no emulador do Android, siga estas etapas:

1. Habilite a [aceleração de VM](#) em sua máquina.
2. Inicie o **Android Studio** , clique no ícone do **AVD Manager** e selecione **Criar dispositivo virtual...**
 - Em versões mais antigas do Android Studio, você deve iniciar o **Android Studio > Tools > Android > AVD Manager** e selecionar **Create Virtual Device...** . (O submenu **Android** só está presente quando dentro de um projeto **Android**.)
 - Se você não tiver um projeto aberto, você pode escolher **Configurar > AVD Manager** e selecionar **Criar dispositivo virtual...**
3. Escolha uma definição de dispositivo e selecione **Avançar** .
4. Selecione uma ou mais imagens do sistema para as versões do Android que você deseja emular e selecione **Avançar** . Uma imagem x86 ou x86_64 é recomendada.
5. Em Desempenho Emulado, selecione **Hardware - GLES 2.0** para habilitar a [aceleração de hardware](#) .

6. Verifique se a configuração do AVD está correta e selecione **Concluir** .

Para obter detalhes sobre as etapas acima, consulte [Gerenciando AVDs](#) .

7. No Android Virtual Device Manager, clique em **Executar** na barra de ferramentas. O emulador é inicializado e exibe a tela padrão para a versão do sistema operacional e o dispositivo selecionados.

Concordar com as licenças do Android

Antes de poder usar o Flutter, você deve concordar com as licenças da plataforma Android SDK. Esta etapa deve ser feita após a instalação das ferramentas listadas acima.

1. Certifique-se de que você tenha uma versão do Java 8 instalada e que sua `JAVA_HOME` variável de ambiente esteja configurada para a pasta do JDK.

As versões do Android Studio 2.2 e superiores vêm com um JDK, então isso já deve ser feito.

2. Abra uma janela de console elevada e execute o comando a seguir para começar a assinar licenças.

```
content_copy  
$ flutter doctor --android-licenses
```

3. Revise os termos de cada licença cuidadosamente antes de concordar com eles.
4. Quando terminar de concordar com as licenças, execute `flutter doctor` novamente para confirmar que você está pronto para usar o Flutter.

Configuração do Windows

Requisitos adicionais do Windows

Para o desenvolvimento da área de trabalho do Windows, você precisa do seguinte, além do SDK do Flutter:

- [Visual Studio 2022](#) Ao instalar o Visual Studio, você precisa da carga de trabalho “Desktop development with C++” instalada para construir janelas, incluindo todos os seus componentes padrão.

Observação: o **Visual Studio** é diferente do *Visual Studio Code*.

Para obter mais informações, consulte [Suporte de desktop para Flutter](#)

Configuração da Web

Flutter tem suporte para construção de aplicações web no `stable` canal. Qualquer aplicativo criado no Flutter 2 é criado automaticamente para a web. Para adicionar suporte à Web a um aplicativo criado antes da estabilidade da Web, siga as instruções em [Como criar um aplicativo Web com o Flutter](#) ao concluir a configuração acima.