

Moralar

Manual de instalação para Aplicativo TTS Profissional

Tabela de conteúdo

Tabela de conteúdo	1
1. Detalhes do projeto	2
1.1 Módulos usados pelo aplicativo	2
1.2 Unidades necessárias para seu funcionamento adequado	4
1.2.1 Verifique a raiz do projeto:	4
1.2.2 Verifique se temos as bibliotecas externas do projeto:	4
2 Requisitos para a instalação local	5
2.1 Requisitos antes da instalação	5
2.2 Conhecimento prévio	5
3. Como executar o programa localmente	6
3.1 Trazendo as bibliotecas por meio do terminal	6
3.2 Instalando o emulador a partir do Android Studio	6
3.3 Configuração do ponto de extremidade da API	7
3.4 Abrindo o emulador que instalamos	7
4. compilação do projeto	9
4.1 Compilação para APK	9

1. Detalhes do projeto


Aplicativo TTS Moralar Profissional: É um aplicativo móvel feito em [Flutter](#) onde os consultores do tipo **TTS Professional** poderão revisar os detalhes de uma família, com todas as informações importantes para reagendar, cancelar agendas, extrair relatórios de partidas, revisar indicadores no painel e muito mais.

1.1 Módulos usados pelo aplicativo

- a. **Dashboard:** é usado para analisar indicadores como:
 - Número total de famílias registradas
 - Porcentagem de estágios de agendamento
 - Número de propriedades disponíveis para compra
 - Número de propriedades vendidas

- b. **Famílias:** é o módulo principal do aplicativo, no qual podemos extrair um relatório em Excel das famílias registradas. Também podemos ver os detalhes da família, onde encontraremos todas as informações coletadas da família, como
 - Ver as agendas familiares programadas
 - Verifique se você respondeu ao questionário.
 - Verifique se você concluiu as pesquisas
 - Verifique se você está matriculado em algum curso

- c. **Agendas:** Este é um módulo em que seremos capazes de:
 - Filtre os compromissos por status, tipo de compromisso etc.
 - Visualize os detalhes de um compromisso com sua cor de status correspondente e informações detalhadas sobre o ocupante e os detalhes do compromisso.
 - Poderemos remarcar e cancelar o compromisso.

- 
- d. **Correspondências:** Neste módulo, seremos capazes de:
- Filtrar por número de registro, nome do proprietário, cpf, código de construção etc.
 - Extrair um relatório de jogo para o Excel
 - Exibir informações sobre cada partida
- e. **Questionários:** Aqui você pode ver a lista de todos os questionários e também responder ao questionário que não foi preenchido.
- f. **Editar perfil:** Aqui você pode alterar as informações básicas do seu perfil profissional.
- g. **Alterar senha:** nesse módulo, você pode alterar a senha de login do aplicativo tts professionals.

1.2 Unidades necessárias para seu funcionamento adequado

Esta seção discutirá o que você precisa ter pronto antes de fazer uma instalação em seu computador local.

1.2.1 Verifique a raiz do projeto:

a. Opção A:

- Clone o projeto que já passou pelo repositório, acesse a ramificação "Development" e entre na raiz do projeto, onde a pasta se chama ***moralar-apptts-flutter***.

Opção B:

- Acesse a pasta inputs e entre na pasta ***moralar-apptts-flutter***, dentro dessa pasta estará a raiz do projeto.

- b. Depois de sabermos qual é a raiz do projeto, precisamos verificar se o arquivo README.md está dentro dessa pasta, para sabermos que é a raiz do projeto, o que é essencial ter em mente ao executar o projeto localmente.

1.2.2 Verifique se temos as bibliotecas externas do projeto:

É essencial ter essas pastas, que mencionaremos mais adiante, pois dependemos dessas bibliotecas para que o projeto seja executado corretamente.

- Etapa 1: verifique se, fora da raiz do projeto, temos essas pastas:
 - **moralar-widgets**: essa biblioteca personalizada contém os widgets (componentes) reutilizáveis do aplicativo.
 - **mega-flutter**: essa biblioteca personalizada contém as funcionalidades de autenticação e segurança do aplicativo.
- Etapa final: insira o arquivo ***pubspec.yaml*** localizado na raiz do projeto ***moralar-apptts-flutter*** e verifique se, na seção **de dependências**, você tem acesso aos caminhos mencionados na etapa 1.

2 Requisitos para a instalação local

Nesta seção, discutiremos como executar nosso aplicativo flutter a partir de uma máquina Windows local, tendo em mente que você deve atender aos pré-requisitos de conhecimento e ter concluído com êxito o índice 1.2.

2.1 Requisitos antes da instalação

Você deve ter essas versões ou versões superiores instaladas para poder executar o projeto no computador local.

- a. **Dart:** [3.6.1](#)
- b. **Flutter:** [3.27.3](#)
- c. **Android studio:** [qualquer versão](#) compatível com o SDK ($\geq 2.19.0$ $< 3.0.0$)
- d. **Visual studio code insiders:** [qualquer versão](#)
- e. Extensões de código Vs:
 - Dart
 - Flutter
 - Utilitários Flutter & dart (opcional)
 - Ajudantes de flutter (opcional)
 - Flutter Intl (opcional)
 - Snippets do Flutter (opcional)
- f. Configuração de variáveis de ambiente [\(guia\)](#)

2.2 Conhecimento prévio

1. Conhecer os comandos [do Dart pub](#).
2. Conheça os comandos [de pub do Flutter](#).
3. Saber como executar um emulador no Android Studio.
4. Saber como instalar um emulador e diferenciar se o emulador é api 33 ou api 34 etc.
5. Ter conhecimento de como ver os erros de saída no código vs. caso ocorra um erro durante a compilação [\(guia\)](#).
6. Como depurar um projeto flutter a partir do código vs., [guia completo](#)

3. Como executar o programa localmente

Nesta seção, explicaremos de forma resumida como depurar o projeto flutter no código vs. Se você quiser revisar o guia completo em mais detalhes, consulte este [link](#).

3.1 Trazendo as bibliotecas por meio do terminal

- Etapa 1: Execute o Vs code insiders e insira a raiz do projeto.
- Etapa 2: Vá para a guia Exibir -> Terminal.
- Etapa final: execute os seguintes comandos para ***moralar-widgets, mega-flutter e moralar-appts-flutter***. *Verifique também* se eles foram concluídos com êxito:
 - `dart pub get`
 - `flutter pub get`

3.2 Instalando o emulador a partir do Android Studio

- Etapa 1: Abra o Android Studio.
- Etapa 2: crie uma nova decisão no gerenciador de dispositivos.
- Etapa 3: Escolha qualquer telefone, o que usamos para o desenvolvimento do Moralar é o **Pixel 7 Pro**.
- Etapa 4: Aplique o botão seguinte.
- Etapa 5: Escolha o nível de API **nº 34** (UpSideDownCake).
- Etapa 6: Aplique o botão seguinte.
- Etapa final: verifique a configuração e clique no botão Finish (Concluir).

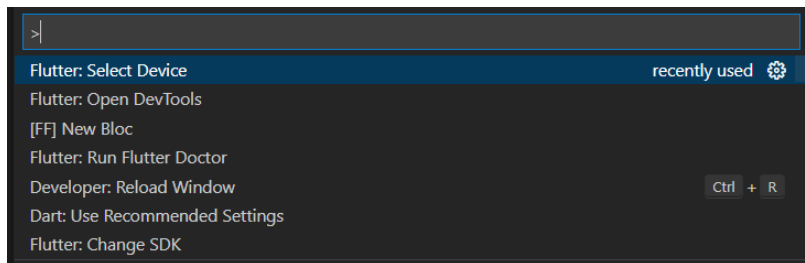
3.3 Configuração do ponto de extremidade da API

Nesta seção, configuraremos o servidor e a API para os quais faremos todos os tipos de solicitações https do nosso aplicativo flutter.

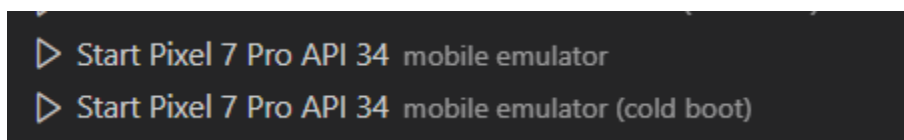
- Etapa 1: Iremos para a pasta da biblioteca personalizada do moralar, que chamamos de ***moralar-widgets-lib***.
- Etapa 2: Insira o arquivo na pasta lib chamado ***moralar_widgets.dart***.
- Etapa 3: Localize a variável ***baseUrlForHTTP*** e altere o domínio da url, mas deixe o caminho `/api/v1` intacto.
- Etapa 4: Localize a variável ***baseUrlAssets*** e altere o domínio do URL, mas deixe o caminho `/content/upload` intacto.
- Etapa 5: Execute os seguintes comandos para ***moralizar-widgets***. *Verifique também se eles foram concluídos com êxito:*
 - `dart pub get`
 - `flutter pub get`

3.4 Abrindo o emulador que instalamos

- Etapa 1: Abra o vs code insiders a partir da raiz do projeto ***moralar-apptts-flutter***.
- Etapa 2: execute o atalho de teclado `Ctrl+shift+P` do vs code insiders, onde essa lista será exibida.

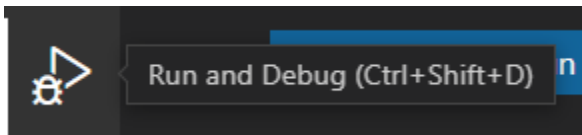


- Etapa 3: Clique na opção: Flutter: Select Device.
- Etapa 4: Escolha a opção do emulador que instalamos anteriormente; neste caso, o que usamos é: Start Pixel 7 Pro Api 34.

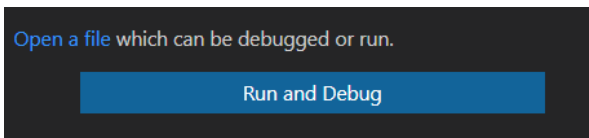


- Etapa 5: aguarde a abertura do emulador de dispositivo móvel e conclua a configuração básica do emulador (a configuração do dispositivo será feita apenas uma vez).

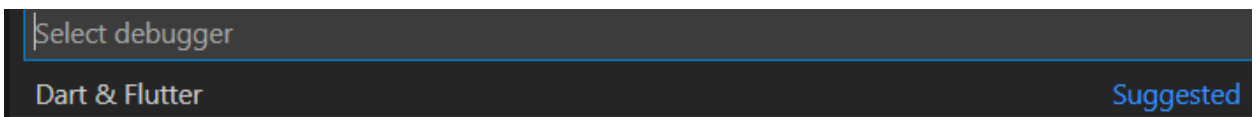
- Etapa 6: vá para a seção Run and debug (Executar e depurar) do vs code insiders.



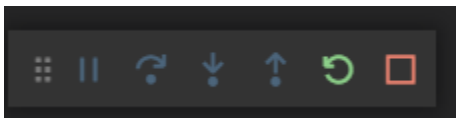
- Etapa 7: Execute o botão executar e depurar.



- Etapa 8: Você obterá essa lista e escolherá Dart & Flutter.



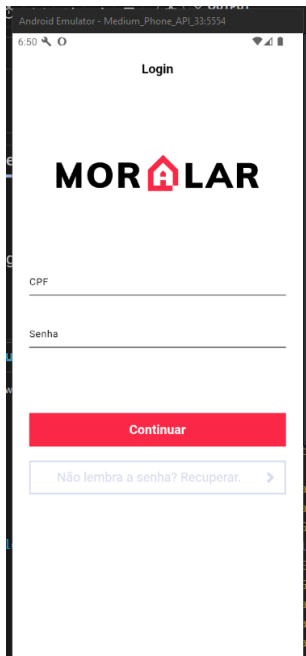
- Etapa 9: Se tudo estiver ok, teremos estes botões de depuração.



- Etapa 10: Se os botões anteriores não estiverem ocultos, podemos verificar na guia View -> Output (Exibir -> Saída) que o projeto está sendo carregado e que, após alguns minutos, o aplicação que queríamos verificar será executado no emulador.

Observação: fique atento se aparecerem letras vermelhas na saída e verifique se há erros de instalação.

- Etapa final: se tudo estiver correto, veremos o login do Moralar em nosso emulador.



4. compilação do projeto

4.1 Compilação para APK

Nesta seção, falaremos sobre como compilar nosso aplicativo flutter em um apk do Android.

- Etapa 1: Abra o terminal a partir da raiz do nosso projeto.
- Etapa 2: Executar os comandos
 - `flutter pub get`
 - `flutter build apk`
- Etapa 3: Aguarde a geração do apk e acesse o caminho que indica que o processo de compilação foi concluído com êxito.

```
Running Gradle task 'assembleRelease'...  
✓ Built build\app\outputs\flutter-apk\app-release.apk (23.2MB)
```