



Esse documento foi elaborado pelo [Programador Aventureiro](#) e é um guia de configuração da estrutura de projetos Flet para distribuição e personalização da aplicação final.

Dúvidas envie no nosso grupo do Telegram [@programadoraventureiro](#)

Preparando o App para distribuição

Estrutura geral do projeto

- /assets/
 - icon.png
 - manifest.json
- main.py
- requirements.txt

Ícones

- Os ícones podem ser nos formatos **.png**, **.bpm**, **.jpg** ou **.webp**.
- Para usar ícones diferentes em cada plataforma, salve as variações desejadas dentro de **assets** diretamente:
 - **iOS:** **icon_ios.png** (ou qualquer formato de imagem suportado). O tamanho mínimo recomendado da imagem é 1024 px. A imagem não deve ser transparente (ter canal alfa). Padrão é icon.png com o canal alfa removido automaticamente.
 - **Android:** **icon_android.png** (ou qualquer formato de imagem suportado). O tamanho mínimo recomendado da imagem é 192 px. Padrão é icon.png.
 - **Web:** **icon_web.png** (ou qualquer formato de imagem suportado). O tamanho mínimo recomendado da imagem é 512 px. Padrão é icon.png.

- **Windows:** `icon_windows.png` (ou qualquer formato de imagem suportado). ICO será produzido com tamanho de 256 px. Padrão é `icon.png`. Se o arquivo `icon_windows.ico` for fornecido, ele será apenas copiado para `windows/runner/resources/app_icon.ico` sem modificações.
- **macOS:** `icon_macos.png` (ou qualquer formato de imagem suportado). O tamanho mínimo recomendado da imagem é 1024 px. Padrão é `icon.png`.
- Se nenhum ícone for passado ele usará o padrão do Flet e se apenas `icon.png` estiver disponível então será usado para todos os dispositivos.

Tela inicial

Você pode personalizar a imagem que é exibida no carregamento do aplicativo para cada plataforma (splash screen):

- **iOS (claro):** `splash_ios.png` (ou qualquer formato de imagem suportado). Padrão é `splash.png` e então `icon.png`.
- **iOS (escuro):** `splash_dark_ios.png` (ou qualquer formato de imagem suportado). Padrão é a splash clara do iOS, então `splash_dark.png`, então `splash.png` e então `icon.png`.
- **Android (claro):** `splash_android.png` (ou qualquer formato de imagem suportado). Padrão é `splash.png` e então `icon.png`.
- **Android (escuro):** `splash_dark_android.png` (ou qualquer formato de imagem suportado). Padrão é a splash clara do Android, então `splash_dark.png`, então `splash.png` e então `icon.png`.
- **Web (claro):** `splash_web.png` (ou qualquer formato de imagem suportado). Padrão é `splash.png` e então `icon.png`.
- **Web (escuro):** `splash_dark_web.png` (ou qualquer formato de imagem suportado). Padrão é a splash clara da web, então `splash_dark.png`, então `splash.png` e então `icon.png`.

Além disso pode escolher qual a cor exibida no fundo do app na tela inicial, para isso use os comando abaixo no momento de gerar o bundle com `flet build`:

- A opção `--splash-color` especifica a cor de fundo para uma tela de inicialização no modo claro. O padrão é `#ffffff`.
- A opção `--splash-dark-color` especifica a cor de fundo para uma tela de inicialização no modo escuro. O padrão é `#333333`.

Bibliotecas

- O arquivo `requirements.txt` deve conter apenas o nome das bibliotecas usadas no projeto.

Não use o comando ``pip freeze > requirements.txt`` para criar o ``requirements.txt`` para o aplicativo que será executado em dispositivos móveis. Ao executar o comando ``pip freeze`` em um ambiente de desktop, o ``requirements.txt`` terá dependências que não são destinadas a funcionar em um dispositivo móvel, como `watchdog`.

Escolha manualmente as dependências no `requirements.txt` para incluir apenas as dependências diretas necessárias para o seu aplicativo, além de flet.

Arquivos externos

Você pode incorporar arquivos externos ao seu projeto com Flet da maneira que preferir dentro do diretório **assets**, porém recomendo seguir a estrutura abaixo:

- **images:** Use para salvar todas as imagens que serão usadas no seu aplicativo.
- **fonts:** Fontes personalizadas para usar no projeto.
- **audios:** Arquivos de áudio como músicas de fundo e sons de interação.
- **media:** Demais arquivos de mídia como vídeos.

Exemplo de estrutura:

```
/assets/  
  /images/  
    - /avatars/  
      - default.jpg  
    - background.jpg  
    - welcome.png  
  /fonts/  
    - SuperMario.ttf  
  /audios/  
    - click.wav  
    - error.wav  
    - success.wav  
  /media/  
    - intro.mp4
```

Organização para aplicativos modulares

Ao desenvolver um aplicativo profissional com muitas telas é recomendado criar subdiretórios no seu projeto para deixá-lo modular. Essa organização pode ser seguida para separar os módulos do seu projeto:

- **assets:** Use este diretório para armazenar arquivos estáticos, como imagens, ícones, fontes, etc.
- **components:** Este diretório é usado para armazenar componentes reutilizáveis da sua aplicação, como botões, cartões, formulários, etc.
- **pages:** Aqui você pode colocar os componentes que representam páginas inteiras da sua aplicação. Cada arquivo pode corresponder a uma rota específica da aplicação.
- **utils:** Neste diretório, você pode colocar funções utilitárias que são usadas em toda a aplicação, como funções de formatação de data, funções de cálculo, etc.
- **services:** Armazene aqui os serviços da aplicação, como lógica de negócios, chamadas de API, manipulação de dados, etc. Cada subdiretório pode corresponder a um serviço específico ou a um

domínio da aplicação.

Lembre de colocar um arquivo `__init__.py` dentro de cada diretório (exceto o `assets`), para transformá-los em módulos do seu projeto.

Crie um projeto com a estrutura mínima e personalize

Você pode criar um novo projeto com a estrutura mínima necessária para gerar seu aplicativo com Flet usando o comando: `flet create meu_projeto`, onde `meu_projeto` é o nome do diretório que será criado para armazenar seu projeto.

O resultado será a estrutura abaixo:

```
/assets/  
    icon.png  
main.py  
requirements.txt
```