



**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**ENGENHARIA DA COMPUTAÇÃO**  
**PROJETO E DESENVOLVIMENTO DE SOFTWARE**  
**BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA**

**André Luis Aguiar do Nascimento**

**Hugo Samuel de Lima Oliveira**

**Leonardo Sampaio Serra**

**Lucas Emanuel Amaral Gomes**

**Wesley dos Santos Gatinho**

**FITAI – MANUAL DE INSTALAÇÃO**

# Manual de Instalação do Projeto FitAI

Este manual detalha os passos necessários para configurar e executar o projeto FitAI, que consiste em um backend (API RESTful) desenvolvido em Python com FastAPI e um aplicativo móvel desenvolvido em Flutter.

## 1. Estrutura do Projeto

O repositório G1\_FITAI contém duas pastas principais para o código:

- **Codigos/backend:** Contém o código-fonte da API RESTful.
- **Codigos/flutter\_app:** Contém o código-fonte do aplicativo móvel Flutter.

## 2. Instalação do Backend (API RESTful)

O backend é responsável pela lógica de negócio, autenticação, gerenciamento de dados e integração com modelos de IA.

### 2.1. Pré-requisitos do Backend

Certifique-se de ter os seguintes softwares instalados em sua máquina:

- **Python 3.9+:** Recomendado usar a versão mais recente compatível.
- **PostgreSQL:** Um servidor de banco de dados PostgreSQL.
- **Git:** Para clonar o repositório

### 2.2. Passos de Instalação do Backend

#### 1. Clone o repositório:

Abra seu terminal ou prompt de comando e execute:

```
git clone https://github.com/wesleygatinho/G1\_FITAI.git
```

```
cd G1_FITAI/Codigos/backend
```

#### 2. Crie e ative um ambiente virtual:

É altamente recomendável usar um ambiente virtual para gerenciar as dependências do Python.

```
python3 -m venv .venv
```

- **No Linux/macOS:**

```
source .venv/bin/activate
```

- **No Windows (Command Prompt):**

**NN.venv\Scripts\activate.bat**

- **No Windows (PowerShell):**

**.venv\Scripts\Activate.ps1**

### **3. Instale as dependências do Python:**

Com o ambiente virtual ativado, instale todas as bibliotecas necessárias listadas no **requirements.txt**:

**pip install -r requirements.txt**

As principais dependências incluem: fastapi, uvicorn, sqlalchemy, psycopg2-binary, alembic, pydantic, pydantic-settings, passlib[bcrypt], python-jose[cryptography], fastapi-users[sqlalchemy], google-generativeai, opencv-python, mediapipe, google-auth, authlib.

### **4. Configure o Banco de Dados PostgreSQL:**

- Certifique-se de que seu servidor PostgreSQL esteja em execução.
- Crie um banco de dados chamado `fitai_db`. Você pode fazer isso via `psql` ou uma ferramenta GUI como `pgAdmin`. Exemplo via `psql`:

**CREATE DATABASE fitai\_db;**

- Crie um usuário e senha para o banco de dados, se necessário, e conceda as permissões adequadas.

### **5. Configure as Variáveis de Ambiente:**

Crie um arquivo chamado `.env` na raiz da pasta `Codigos/backend` (ao lado de `main.py` e `requirements.txt`). Adicione as seguintes variáveis, substituindo os valores pelos seus dados reais:

```
DATABASE_URL="postgresql://seu_usuario:sua_senha@localhost/fitai_db"
SECRET_KEY="SUA_CHAVE_SECRETA_SUPER_LONGA_E_SEGURA"
GOOGLE_API_KEY="SUA_CHAVE_DA_API_DO_GOOGLE"
```

- **DATABASE\_URL:** A string de conexão para o seu banco de dados PostgreSQL.
- **SECRET\_KEY:** Uma chave secreta longa e aleatória usada para segurança (por exemplo, para assinar tokens JWT). Você pode gerar uma usando `openssl rand -hex 32`.
- **GOOGLE\_API\_KEY:** Sua chave de API para acessar os serviços do Google Gemini (necessária para a funcionalidade de IA).

## 6. Execute o Servidor Backend:

Com o ambiente virtual ativado e as variáveis de ambiente configuradas, inicie o servidor Uvicorn:

```
uvicorn app.main:app --reload --host 0.0.0.0 --port 8000
```

## 7. Instalação do Aplicativo Flutter

O aplicativo Flutter é a interface do usuário que interage com o backend.

### 3.1. Pré-requisitos do Aplicativo Flutter

- **Flutter SDK** (versão 3.0.0 ou superior): Instale o Flutter SDK seguindo as instruções oficiais do Flutter.
- **Dart SDK** (versão 3.0.0 ou superior): Geralmente vem junto com o Flutter SDK.
- **Um emulador Android/iOS ou dispositivo físico:** Configure um ambiente de desenvolvimento Flutter (Android Studio com emulador, Xcode com simulador, ou um dispositivo físico conectado).
- **Backend em execução:** O servidor backend deve estar rodando e acessível.

### 3.2. Passos de Instalação do Aplicativo Flutter

#### 1. Navegue até a pasta do aplicativo:

Se você já clonou o repositório, navegue para a pasta do Flutter:

```
cd G1_FITAI/Codigos/flutter_app
```

## 2. Verifique a instalação do Flutter:

Execute para garantir que seu ambiente Flutter esteja configurado corretamente:

**flutter doctor**

Resolva quaisquer problemas relatados pelo flutter doctor antes de prosseguir.

## 3. Instale as dependências do Flutter:

Execute este comando para baixar todos os pacotes listados no pubspec.yaml:

**flutter pub get**

As principais dependências incluem: http, provider, shared\_preferences, fl\_chart, camera, flutter\_markdown, flutter\_launcher\_icons, flutter\_native\_splash, table\_calendar, image\_picker, intl, google\_sign\_in, cupertino\_icons, collection.

## 4. Configure a Conexão com a API do Backend:

Abra o arquivo `Codigos/flutter_app/lib/services/api_service.dart`. Você precisará atualizar a constante `_localIp` para apontar para o endereço IP da máquina onde o seu backend está rodando.

- **Se o backend estiver rodando no mesmo computador e você estiver usando um emulador Android:**

Normalmente, 10.0.2.2 é o alias do host loopback do seu computador dentro do ambiente do emulador Android.

**static const String \_localIp = "10.0.2.2"; // Para emulador Android**

- **Se o backend estiver rodando no mesmo computador e você estiver usando um simulador iOS:**

Você pode usar localhost ou o IP da sua máquina.

**static const String \_localIp = "localhost"; // Para simulador iOS**

- **Se você estiver usando um dispositivo físico (Android ou iOS):**

Você precisará encontrar o endereço IP local da sua máquina (ex: 192.168.1.X) e usá-lo aqui. Certifique-se de que ambos os dispositivos estejam na mesma rede.

**static const String \_localIp = "SEU\_IP\_DA\_MAQUINA\_AQUI"; // Ex: "192.168.1.100"**

## 5. Execute o Aplicativo Flutter:

Com um emulador em execução ou um dispositivo físico conectado e reconhecido pelo flutter doctor, execute o seguinte comando na raiz da pasta Codigos/flutter\_app:

**flutter run**

O Flutter compilará e implantará o aplicativo no dispositivo/emulador selecionado.

## Reconhecimentos e Direitos Autorais

@autor:

ANDRE LUIS AGUIAR DO NASCIMENTO, LEONARDO SAMPAIO SERRA, HUGO SAMUEL DE LIMA OLIVEIRA, LUCAS EMANOEL AMARAL GOMES, WESLEY DOS SANTOS GATINHO

@contato: [andre.aguiar@discente.ufma.br](mailto:andre.aguiar@discente.ufma.br), [leonardo.sampaio@discente.ufma.br](mailto:leonardo.sampaio@discente.ufma.br),  
[hugo.samuel@discente.ufma.br](mailto:hugo.samuel@discente.ufma.br), [lucas.amaral@discente.ufma.br](mailto:lucas.amaral@discente.ufma.br),  
[wesley.gatinho@discente.ufma.br](mailto:wesley.gatinho@discente.ufma.br)

@data última versão: 01/07/2025

@versão: 1.0.0

@outros repositórios: [https://github.com/wesleygatinho/G1\\_FITAI.git](https://github.com/wesleygatinho/G1_FITAI.git)

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

### Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina PROJETO E DESENVOLVIMENTO DE SOFTWARE, sob a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2025.1, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da Licença MIT, conforme descrito abaixo, e, portanto, é compatível com a GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Licenciado sob a Licença MIT. Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar no Software sem restrição, incluindo sem limitação os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, e permitir pessoas a quem o Software é fornecido a fazê-lo, sujeito às seguintes condições:

Este aviso de direitos autorais e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>.