

Guia para Execução do Projeto (Backend + Frontend)

Este guia descreve, passo a passo, como executar o backend em FastAPI e o frontend em Expo/React Native, utilizando as versões recomendadas de Python, Node.js e npm. Também apresenta o sistema já funcionando 100% online e um vídeo explicativo de como o projeto opera.

Repositório do projeto (GitHub):

<https://github.com/rogerdapalma/TCC>

1. Acesso Online ao Sistema

Aplicação web (produção):

<https://nutricomp-web.pages.dev>

Vídeo explicativo sobre o funcionamento do sistema:

<https://youtu.be/1bh-YKtOu2o>

2. Pré-requisitos

2.1 Clonar o repositório do projeto:

```
git clone https://github.com/rogerdapalma/TCC.git  
cd TCC
```

2.2 Confirmar as versões instaladas de Python, Node.js e npm (recomenda-se utilizar exatamente as versões abaixo):

```
python --version    # Python 3.13.9  
node --version     # v20.18.1  
npm --version      # 10.8.2
```

3. Instalação das versões recomendadas

3.1 Python (Windows via winget):

```
winget install -e --id Python.Python.3.13  
python --version
```

3.2 Node.js (usando NVM for Windows):

```
winget install -e --id CoreyButler.NVMforWindows  
nvm install 20.18.1  
nvm use 20.18.1  
node --version
```

3.3 npm (atualizar para a versão recomendada):

```
npm install -g npm@10.8.2  
npm --version
```

4. Terminal A — Backend (FastAPI)

No primeiro terminal, acesse o diretório do servidor:

```
cd tcc/server
```

4.1 Criar e ativar o ambiente virtual:

macOS/Linux:

```
python -m venv .venv  
source .venv/bin/activate
```

Windows (PowerShell):

```
python -m venv .venv  
.venv\Scripts\Activate.ps1
```

4.2 Instalar dependências do backend:

Com o ambiente virtual ativado, instale/atualize o gerenciador de pacotes e as dependências do projeto:

```
pip install --upgrade pip  
pip install -r requirements.txt
```

4.3 Executar a API (porta 8000):

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

4.4 Testar a API:

Para verificar se a API está respondendo corretamente, você pode utilizar o comando abaixo:

```
curl http://127.0.0.1:8000/health
```

Também é possível abrir a documentação automática da API no navegador:

<http://127.0.0.1:8000/docs>

5. Terminal B — Frontend (Expo/React Native)

Abra um segundo terminal para executar o frontend e accesse o diretório do cliente:

```
cd tcc/client
```

5.1 Instalar dependências do aplicativo:

```
npm install
```

Em caso de conflitos de dependências, a limpeza abaixo pode ajudar:

```
rm -rf node_modules package-lock.json  
npm cache clean --force  
npm install
```

5.2 Executar o aplicativo:

Modo Web (recomendado para testes rápidos):

```
npx expo start --web
```

Também é possível executar o app em emulador ou dispositivo físico:

```
npx expo start --android  
npx expo start --ios # disponível apenas em macOS com Xcode
```

6. Ajuste do banco de dados (FOODS_DB)

Se, ao iniciar o backend, o sistema não encontrar o arquivo de banco de dados *alimentos.sqlite*, é possível configurar o caminho manualmente como no exemplo abaixo.

6.1 No terminal do servidor (backend):

```
cd tcc\server  
python -m venv .venv  
.venv\Scripts\Activate.ps1  
pip install -r requirements.txt  
  
$env:FOODS_DB = (Resolve-Path ..\data\alimentos.sqlite).Path  
  
@"
FOODS_DB=$($env:FOODS_DB -replace '\\\\','\\')
"@ | Set-Content -Encoding UTF8 .env  
  
uvicorn app.main:app --reload --port 8000
```

6.2 No terminal do cliente (frontend):

```
cd tcc\client  
npm install  
$env:EXPO_PUBLIC_API = "http://127.0.0.1:8000"
```

```
@"  
EXPO_PUBLIC_API=http://127.0.0.1:8000  
"@ | Set-Content -Encoding UTF8 .env  
  
npx expo start --web
```

Após executar o comando final no cliente (*npx expo start --web*), o aplicativo será iniciado e se comunicará com a API disponível em:

<http://127.0.0.1:8000>