

Trabalho conclusivo de curso

Guia para Rodar o 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, mostrando o sistema já funcionando 100% online com um vídeo explicativo de como o projeto funciona.

O projeto pode ser utilizado em:
<https://nutricomp-web.pages.dev> — sistema 100% online

Contém um vídeo explicando como funciona o site:
<https://youtu.be/1bh-YKtOu2o>

Pré-requisitos

Clonar o repositório do GitHub Repositorio:

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

Antes de iniciar, confirme se as versões instaladas no sistema correspondem às versões abaixo (o ideal é utilizar exatamente estas):

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

Caso não tenha as versões corretas, abra o VS Code com um novo terminal e instale as versões corretas:

Python

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

Node.js

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

npm

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

Terminal A — Backend (FastAPI)

No primeiro terminal, acesse o diretório do servidor:

```
cd tcc/server
```

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
```

2) Instalar dependências

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

```
pip install --upgrade pip python.exe -m
pip install --upgrade pip pip install -r requirements.txt
```

3) Executar a API (porta 8000)

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

4) Testar a API

Para verificar se a API está respondendo corretamente, utilize o comando abaixo em outro terminal, ou até mesmo no próprio terminal do backend:

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

Também é possível verificar a documentação automática da API no navegador, acessando:

```
http://127.0.0.1:8000/docs
```

Terminal B — Frontend (Expo/React Native)

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

```
cd tcc/client
```

1) Instalar dependências do aplicativo

Instale as dependências do projeto React Native/Expo com o npm:

```
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
```

2) Executar o aplicativo

Modo Web (recomendado para testes rápidos):

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

Opcionalmente, é possível executar o app em um emulador ou dispositivo físico:

```
npx expo start --android
npx expo start --ios
```

(apenas em macOS com Xcode)

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 conforme o exemplo abaixo.

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
```

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
```