

Guia — Backend & Frontend (FastAPI + Expo)

Instruções de execução (pré-requisitos, backend, frontend e troubleshooting) preservadas exatamente como fornecidas.

```
# =====
# PRE-REQUISITOS
# =====
# Confirme versões (ok se forem iguais ou maiores):
python --version - Python 3.13.9
node --version - v20.18.1
npm --version - 10.8.2

# =====
# TERMINAL A - BACKEND (FastAPI)
# Caminho: tcc/server
# =====
cd tcc/server

# 1) Crie e ative o ambiente virtual
# macOS/Linux:
# python -m venv .venv
# source .venv/bin/activate
# Windows (PowerShell):
# python -m venv .venv
# .venv\Scripts\Activate.ps1

# 2) Instale dependências
pip install --upgrade pip
pip install -r requirements.txt

# (Opcional) Se quiser sobrescrever configs (exemplos):
# macOS/Linux:
# export DB_PATH="app/data/alimentos.sqlite"
# export TABLE_NAME="alimentos"
# export COL_ID="id"
# export COL_DESC="description"
# export COL_CAT="category"
# Windows (PowerShell):
# $env:DB_PATH="app/data/alimentos.sqlite"
# $env:TABLE_NAME="alimentos"
# $env:COL_ID="id"
# $env:COL_DESC="description"
# $env:COL_CAT="category"

# 3) Rode a API (porta 8000)
uvicorn app.main:app --reload --port 8000

# Em outro terminal você pode testar:
# curl http://127.0.0.1:8000/health
# Abra no navegador: http://127.0.0.1:8000/docs

# =====
# TERMINAL B - FRONTEND (Expo/React Native)
# Caminho: tcc/client
# =====
cd tcc/client

# 1) Garanta o .env apontando para a API local (já está no repo):
```

```

# Arquivo: tcc/client/.env
# Conteúdo esperado:
# EXPO_PUBLIC_API=http://127.0.0.1:8000

# Se for testar em celular físico na mesma rede:
# coloque o IP DA SUA MÁQUINA no lugar de 127.0.0.1, ex:
# EXPO_PUBLIC_API=http://192.168.1.10:8000

# 2) Instale deps do app
npm install
# (se der conflito de dependências)
# rm -rf node_modules package-lock.json && npm cache clean --force && npm install

# 3) Rode o app
# Web (mais simples):
npx expo start --web

# OU em emulador/dispositivo:
# npx expo start --android
# npx expo start --ios # (apenas macOS com Xcode)

# =====
# DICAS / TROUBLESHOOTING
# =====
# - Porta ocupada? Troque a porta do uvicorn:
#   uvicorn app.main:app --reload --port 9000
#   e atualize tcc/client/.env => EXPO_PUBLIC_API=http://127.0.0.1:9000
# - Testando no celular? Use o IP da máquina em vez de 127.0.0.1 no .env do client.
# - Swagger disponível em: http://127.0.0.1:8000/docs
# - O banco SQLite já está incluso em: tcc/server/data/alimentos.sqlite

```

```

# === SERVER ===
```
cd C:\Users\roger\Desktop\TrabalhoFinal\TCC-II\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
```

```

em outro terminal

```

# === CLIENT ===
```
cd C:\Users\roger\Desktop\TrabalhoFinal\TCC-II\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

Um dos dois abaixo (ver conforme seu package.json):
npx expo start --web
ou

```

```
npm run dev
```
```