# Estudo preliminar do que esperamos minimamente do projeto Petro\_Protecai\_Mini

## Versão minimalista do projeto Petro\_Protecai

### Pontos chaves.

1. Trabalhar com IEEE 14 barras 🡪 não reproduz situação real
2. Trabalhar com LR buscando assertividade e tempo de respostas mínimo
3. Desenvolver back
4. Desenvolver Front
5. Usar PostgreSQL – com Docker
6. Trabalhar com CI/CD – GitHub Actions
7. Desenvolver API
8. Simular sistemas com PandaPower
9. Queremos que todo código seja tratado e executado a partir da raiz do projeto.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

O objetivo: **executar qualquer script Python do projeto a partir da raiz (protecai\_mini/)**, sem a necessidade de navegar até subdiretórios como simuladores/pandapower/, src/backend/, tests/, etc.

### Para alcançar esse comportamento com clareza, modularidade e robustez, devemos garantir os seguintes pontos:

#### 1. ****Importações relativas e absolutas bem definidas****

* Os scripts devem usar **importações relativas à raiz do projeto** ou configurar o PYTHONPATH corretamente.

#### 2. ****Uso de**** \_\_init\_\_.py ****para indicar pacotes****

* Cada pasta com código reutilizável (ex: simuladores/pandapower, src/backend) **deve conter um \_\_init\_\_.py**, mesmo vazio.

#### 3. ****Configuração do**** sys.path ****no ponto de entrada****

Inclua o seguinte snippet nos scripts que serão chamados diretamente (ex: scripts\_simulacao.py, test\_simulation.py):

import sys

from pathlib import Path

#### Adiciona a raiz do projeto ao sys.path

ROOT = Path(\_\_file\_\_).resolve().parents[2] # Ajuste conforme a profundidade

sys.path.append(str(ROOT))

#### 4. ****Organização dos diretórios****

Já está ótima, mas confirmando a estrutura mínima ideal: