# ✅ STATUS ATUAL DO PROJETO – Petro\_ProtecAI\_Mini

## 🔹 Objetivo do Projeto

Simular falhas e coordenar proteção elétrica em tempo real via algoritmo de Reforço (RL), sobre a topologia IEEE 14 Barras estendida com relés, disjuntores e lógica de proteção. A aplicação visa ambientes críticos como **plataformas offshore**.

**📂 Estrutura Atual Validada**

* Diretório base: protecai\_mini/
* Pastas principais:
  + simuladores/power\_sim/
    - ieee14.py → modelo base IEEE 14
    - scripts\_simulacao.py → lógica de simulação
    - visualizar\_topologia\_protecao.py → visualização com dispositivos
  + infra/protecao/
    - protecao\_eletrica.py → estrutura OO para relés (Rele51, Rele67, Rele87, etc.)
  + tests/
    - test\_pipeline.py, test\_protecao\_eletrica.py → testes com pytest

**📈 Recursos implementados**

* Modelo IEEE 14 validado com pandapower 2.14.1
* Dispositivos de proteção definidos na **Tabela 2** (já estruturada e validada)
* Visualização inicial da topologia com relés e disjuntores
* Pipeline de CI/CD via **GitHub Actions** funcionando
* Ambiente local com execução via main.py testada e funcional
* Critérios de Recompensa e Penalidade para RL já definidos (evitar blackout, garantir seletividade, etc.)

**🧾 Referências que estão em uso**

**📄 Documento de apoio em memória:**

* Petro\_Protecai\_Mini.docx → guarda contexto geral do projeto, decisões, Tabela 2 e notas

**📊 Tabela 2 – Distribuição de Dispositivos:**

|  |  |  |
| --- | --- | --- |
| Tipo de Dispositivo | Função | Locais Sugeridos |
| Relé 50/51 | Sobrecorrente | Barras 3–4, 4–5, 5–6, 6–13 |
| Relé 67 | Direcional | Barras 2, 3, 6 |
| Relé 87T | Diferencial de Trafo | Barras 2–4, 4–5 |
| Relé 27/59 | Sub/Sobretensão | Barras 7, 9, 10, 14 |
| Disjuntores | Abertura física | Saídas de trafos e geradores |
| Fusíveis (simulados) | Proteção rápida | Barras de cargas críticas |

* O objetivo e a lógica da simulação com proteção coordenada.

**1. 🔌 Estrutura básica da rede:**

* bus: todas as 14 barras da IEEE com seus atributos
* line: conexões entre barras (com impedâncias e comprimentos)
* trafo: se incluímos transformadores de 25 MVA (confirmado)
* gen e load: presentes e consistentes com a base IEEE 14

**2. 🧠 Elementos de proteção (segundo Tabela 2):**

* **protection\_devices** ou outra chave análoga com:
  + Relé 50/51 em: barras 3–4, 4–5, 5–6, 6–13
  + Relé 67 em: barras 2, 3, 6
  + Relé 87T em: barras 2–4, 4–5
  + Relé 27/59 em: barras 7, 9, 10, 14
  + Disjuntores associados a trafos e geradores
  + Fusíveis (simulados) nas barras de carga sensível

**3. 🧭 Atributos esperados por tipo de dispositivo:**

* Para cada relé: tipo, ID, barra associada, curvas (curva 51, 67 etc.), tempo de atuação, corrente de pickup
* Para disjuntores: barra ou linha associada, delay de atuação, status (aberto/fechado)
* Para fusíveis: corrente de fusão, delay, barra de carga

**4. 🧩 Validação cruzada:**

* Não pode haver relés flutuando (sem barra associada)
* Nenhum disjuntor pode estar duplicado ou sem conexão real
* As keys do JSON devem estar completas, sem null, None ou listas vazias

**✅ Proposta de ação imediata:**

1. 📥 **Você pode me enviar o arquivo .json atual**, e eu faço:
   * Uma **auditoria completa e técnica** de sua estrutura
   * Uma **comparação com o que deveria conter**
   * Um relatório dos pontos críticos ou ausentes
2. 🔁 Com base nessa auditoria:
   * Se o JSON estiver **ok**: vamos direto corrigir o visualizar\_topologia\_protecao.py
   * Se o JSON estiver **com falhas estruturais**: então sim, partimos para a refatoração do gerar\_ieee14\_protecao\_json.py

**✅ Resumo da Auditoria Técnica do JSON IEEE14 com Proteção**

**🔹 1. Elementos da Topologia**

* ✔️ bus, line, trafo, load, gen, ext\_grid estão bem estruturados.
* ✔️ Foram inseridos **transformadores de 25 MVA** conectando barras adequadas (0→4 e 1→5).
* ✔️ As cargas estão coerentes com o modelo estendido.
* ⚠️ sgen, motor, asymmetric\_load, shunt, etc., **não possuem dados**, o que é aceitável no nosso contexto atual.

**🔹 2. Verificação de Elementos de Proteção**

* ⚠️ **Nenhuma seção dedicada aos dispositivos de proteção foi encontrada.**
  + Não há qualquer estrutura protection\_devices, rele, breaker, fusible, ou equivalente.
  + Não foram encontrados:
    - Relés 50/51, 67, 87, 27/59
    - Disjuntores
    - Fusíveis (simulados)

**🔹 3. GeoData (usado na visualização)**

* ❌ bus\_geodata e line\_geodata estão **vazios**.
  + Isso **explica o erro do script visualizar\_topologia\_protecao.py**:

yaml

CopiarEditar

Erro ao gerar coordenadas: Coordenadas geradas são inválidas.

**📌 Conclusão: o JSON está incompleto**

|  |  |  |
| --- | --- | --- |
| Item | Situação | Observação |
| Topologia IEEE 14 | ✅ OK | Barras, linhas, trafos e cargas estão corretos |
| Dispositivos de proteção | ❌ AUSENTES | Nenhum relé, disjuntor ou fusível no JSON |
| Coordenadas (bus/linha) | ❌ VAZIAS | Impede renderização gráfica na visualização |
| Estrutura para simulação | ⚠️ Parcial | Pode ser usada para cálculo de fluxo, mas **não** para proteção automatizada |

**🛠️ Próximo passo obrigatório**

Precisamos **refatorar agora mesmo** o script gerar\_ieee14\_protecao\_json.py para que ele:

1. ✅ Mantenha toda a estrutura atual da rede.
2. ➕ **Adicione a seção protection\_devices** com base na Tabela 2:
   * reles: com tipos (51, 67, 87, 27, 59), parâmetros e barras associadas
   * disjuntores: por linha ou por barra, com delay/tempo
   * fusíveis: em barras de carga sensível
3. ➕ **Inclua bus\_geodata e line\_geodata básicos** para visualização funcionar.

**✅ VALIDAÇÃO FINAL DO ARQUIVO ieee14\_protecao.json**

**1. Topologia Elétrica**

* ✅ Rede base IEEE 14 barras está presente e **íntegra**.
* ✅ Transformadores substituíram corretamente as conexões (0,4) e (1,5), com **parâmetros realistas** para aplicações em plataformas de petróleo.
* ✅ Dados de bus, line, trafo, gen, load, etc., estão consistentes e sem erros estruturais.

**2. Coordenadas para Visualização**

* ✅ bus\_geodata e line\_geodata corretamente gerados com layout circular.
* ✅ Todos os bus possuem coordenadas, todas as line têm coords válidas.

**3. Dispositivos de Proteção**

**🔸 reles:**

|  |  |  |
| --- | --- | --- |
| Tipo | Elementos | Detalhes |
| 51 | 4 | Protegem pares de barras em sequência |
| 67 | 3 | Localizados em barras com geração (fluxo reverso possível) |
| 87T | 2 | Referenciam corretamente trafo, com element\_type e element\_id |
| 27/59 | 4 | Aplicados em barras críticas (7, 9, 10, 14) |

✅ Todos os relés contêm campos corretos e seguem nomenclatura padronizada.

**🔸 disjuntores:**

* ✅ Todos os disjuntores **possuem**:
  + element\_type: "bus" ou "gen"
  + element\_id: índice real no net.bus ou net.gen
  + Campos adicionais como barra ou gerador presentes conforme o tipo

**🔸 fusiveis:**

* ✅ Todos possuem element\_type = "bus" e element\_id consistente.
* ✅ Sem duplicidades ou inconsistências.

**4. Padrão e Organização**

* ✅ Todos os campos seguem nomenclatura coerente com o projeto (ex: RELE\_87T\_B2\_B4, DISJ\_G1, FUSIVEL\_B9).
* ✅ Estrutura de net.protection\_devices estende corretamente o objeto pandapowerNet.

**📌 CONCLUSÃO FINAL**

|  |  |
| --- | --- |
| Critério | Status |
| Estrutura elétrica | ✅ OK |
| Transformadores | ✅ OK |
| Coordenadas de visualização | ✅ OK |
| Dispositivos de proteção | ✅ OK |
| Campos extras (element\_\*) | ✅ OK |
| Padrão do projeto | ✅ OK |

🟢 **APROVADO com louvor**: O .json final está **pronto para ser consumido pelo script de visualização** e por **módulos de IA/coordenação de proteção**.

## ⚙️ Resumo Final da Análise dos Três Scripts

|  |  |
| --- | --- |
| Script | Ação Recomendada |
| config\_protecao.py | ❌ **Remover** (substituído por JSON) |
| ieee14.py | ⚠️ **Refatorar** para usar ieee14\_protecao.json |
| scripts\_simulacao.py | ✅ **Manter** com pequenos ajustes futuros |

**✅ Resumo do Escopo**

* Usar a rede IEEE 14 modificada apenas com os **elementos relevantes para testes de coordenação da proteção**.
* **Remover** conexões e elementos da rede original que não estejam sendo usados para proteção, controle ou atuação.
* Incluir **apenas 2 transformadores estratégicos**, entre as barras 0–4 e 1–5.
* Garantir representação **fiel, visual e funcional** no .json e na topologia.
* O sistema final deve ser capaz de simular falhas, observar comportamento da proteção, e permitir redistribuição segura via RL.

## Escopo validado e memorizado com rigor:

* Apenas **os elementos definidos pelo projeto** devem compor a rede modificada.
* Nenhum transformador ou linha "herdada" da IEEE 14 original pode permanecer se **não estiver explicitamente definido para os testes de proteção**.
* A topologia no .json precisa refletir **exatamente o cenário de teste**, ou seja, ser:
  + enxuta,
  + coordenada,
  + adequada à lógica de redistribuição de energia por RL.
* O script visualizar\_topologia\_protecao.py deve **construir o grafo apenas com os elementos efetivamente existentes e necessários no JSON**, e garantir integridade visual, sem artefatos soltos, duplicações nem erros de conectividade.

## 🧠 Memória ativa

* Requisitos de **qualidade de software**: modularidade, docstrings padrão Google, PEP8, robustez
* Local correto de arquivos:
* protecai\_mini/simuladores/power\_sim/data/ieee14\_protecao.json
* protecai\_mini/docs/ieee14\_topologia\_protecao.png
* Pronto para avançar para:
  + Geração de **eventos de falha simulada**
  + Lógica de **atuação dos dispositivos**
  + Criação da **estratégia de Reforço**