

Smart Grid

Simulação de Rede Elétrica Inteligente

Pedro Santana Naves • Pedro Antônio Garcia Gonçalves

Descrição do Projeto

Descrição do Projeto

Este projeto é uma simulação em Java de uma microrrede elétrica inteligente, típica de casas em cidades conectadas. Painéis solares geram energia, que é armazenada em baterias e usada conforme a demanda. Através de uma interface gráfica simples, é possível acompanhar a produção, o consumo e o armazenamento ao longo do tempo. A simulação demonstra, de forma clara e didática, como tecnologias sustentáveis podem ser gerenciadas com eficiência, servindo como base para estudos e aplicações em cenários de Smart Cities.



Relação com Cidades Inteligentes

Relação com Cidades Inteligentes

Microrredes como base de Smart Cities:

A simulação representa uma casa capaz de gerar, armazenar e usar sua própria energia – conceito essencial para cidades mais autônomas e sustentáveis.

Gestão eficiente de energia renovável:

Mostra, de forma didática, como equilibrar produção, consumo e armazenamento usando fontes limpas.

Base para estudos e expansão urbana:

Serve como ferramenta para entender e projetar soluções energéticas em cenários urbanos mais amplos, alinhando-se aos pilares das Smart Cities.

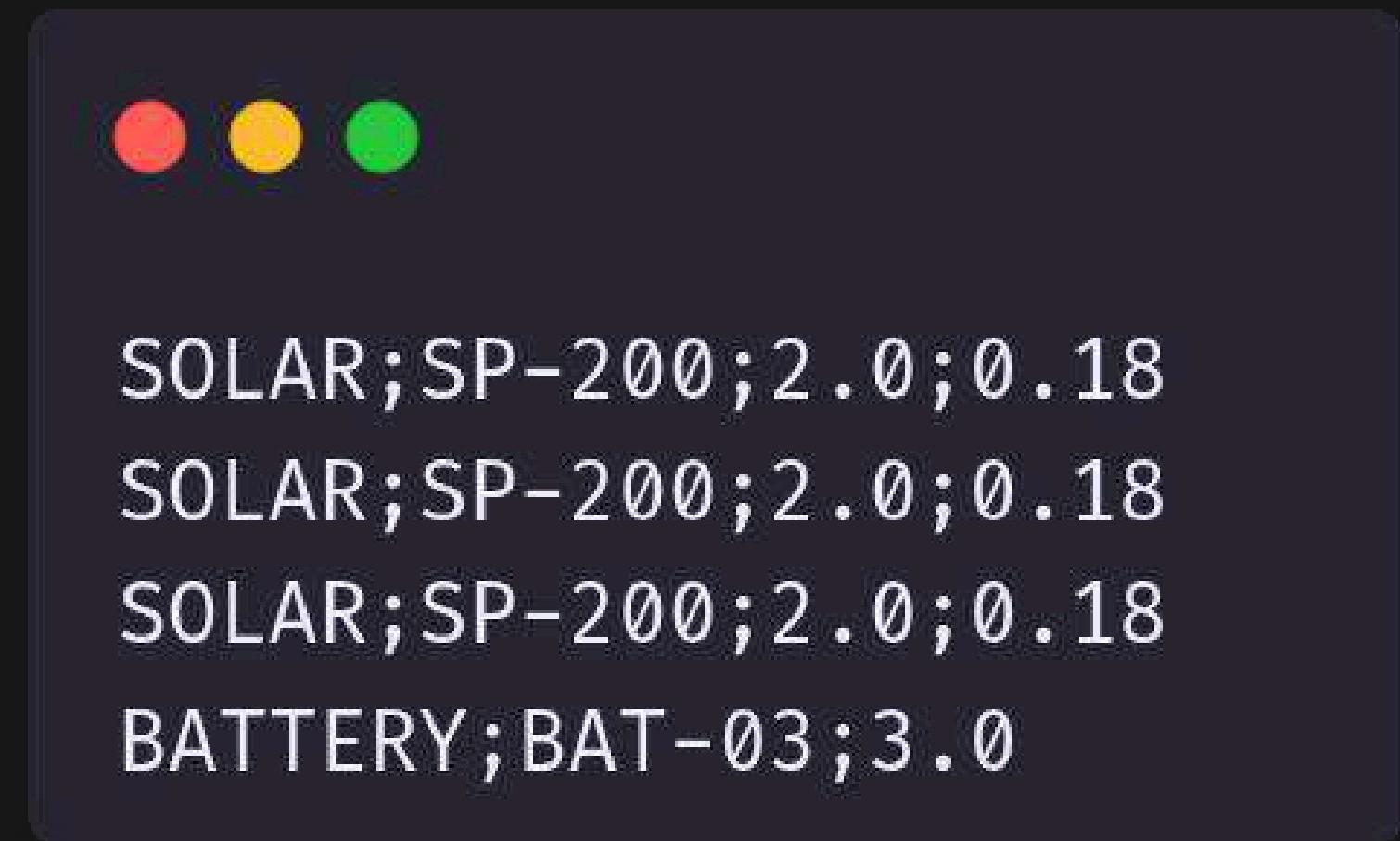
Detalhes de Implementação

Padrões de Projeto

- **Composite:** Organiza painéis e baterias numa estrutura unificada.
- **Factory Method:** Cria dispositivos a partir do arquivo CSV.
- **Flyweight:** Reutiliza configurações iguais para economizar memória.
- **Singleton:** Usa uma única instância para controlar a simulação.
- **Observer:** Atualiza a interface gráfica com dados em tempo real.

Manipulação de Arquivos

- O sistema lê o arquivo devices.csv para configurar a rede.
- Formato: tipo;modelo;área;eficiência. Exemplo:



- Procura o arquivo no diretório atual ou na raiz do projeto.

Tratamento de Exceções

- Em SmartGridApp, um bloco try-catch captura erros como arquivo não encontrado ou formato inválido.
- Em DeviceFactory, linhas inválidas no CSV são ignoradas com mensagens de aviso.
- TickerThread trata InterruptedException para garantir parada segura.

Testes

- A classe BatteryTest verifica se a bateria funciona corretamente.
- Testa limites de carga, descarga e cálculo do estado da bateria.
- Executado com java -ea, confirma se todos os testes passaram.

Interface Gráfica

- A classe SimpleView mostra numa janela:
 - Número do tick.
 - Energia gerada consumida.
 - Carga da bateria (%).
 - Economia (R\$).
- Usa um layout simples e atualiza automaticamente com novos dados.

Threads

- A classe TickerThread atualiza a simulação a cada segundo.
- Chama cálculos e atualiza a interface sem travar o sistema.
- Pode ser parada de forma segura quando necessário.

Diagrama UML do Projeto

Diagrama de Classes UML

Devido as dimensões da imagem, foi optado por inserir o diagrama UML do projeto fora dos slides. Ele pode ser encontrado junto ao GitHub do projeto, na raíz do projeto, entitulado como “uml.pdf”

- Link para o arquivo no GitHub: <https://github.com/pedrosnaves/SmartCitySim/blob/main/uml.pdf>