

Título do Projeto: The Guardian Grid – Inteligência Topológica para Despacho Operacional

Autor: Caio Marcelo Nepomuceno Senra

- **O Problema**

A operação de distribuição atual enfrenta o desafio da cegueira topológica: em momentos de contingência, múltiplos alarmes e chamados são tratados como ocorrências isoladas, mesmo quando pertencem à mesma falha estrutural da rede, levando a diagnósticos manuais lentos e deslocamentos improdutivos de equipes de campo.

O resultado é um despacho orientado a sintomas (clientes sem energia), e não à causa-raiz (ativos da rede), elevando OPEX, TMA e impactando diretamente DEC/FEC.

- **A Solução Proposta**

O Guardian Grid é um motor de inferência baseado em Teoria dos Grafos e arquitetura orientada a eventos, que analisa a conectividade da rede para identificar a provável causa-raiz de falhas em tempo real. A solução atua como uma camada de inteligência "Sidecar" (não-invasiva) sobre os sistemas legados (OMS/SCADA), permitindo que o COD transite de um modelo reativo para um modelo preditivo.

1. O Desafio: o Custo da Operação

Atualmente, a operação trata falhas em um modelo “um para um”: múltiplas reclamações em uma mesma área são interpretadas como ocorrências independentes. Na prática, isso leva ao despacho de equipes para sintomas visíveis (clientes sem luz), enquanto a causa-raiz invisível da rede, como um transformador com chave aberta, permanece.

- **O Cenário:** Quando múltiplos clientes de uma mesma rua ficam sem energia, o sistema atual tende a tratar cada reclamação como uma ordem de serviço isolada.
- **A Consequência:** O WFM (Workforce Management) gera ordens de "verificação de cliente", enviando equipes para olhar medidores individuais, quando o defeito real está no equipamento de média e/ou baixa tensão.
- **O Custo:** Isso infla o OPEX com deslocamentos improdutivos, aumenta o TMA (Tempo Médio de Atendimento) e expõe a equipe a riscos desnecessários, enquanto o defeito real na rede continua ativo.

2. Metodologia e Desafio: o mapeamento em Grafos

O desafio técnico não está na operacionalização do atendimento, mas sim na inteligência de conectar um problema semelhante a outro, utilizando a lógica de agrupamento.

Precisamos de uma camada de inteligência que consiga “segurar” a ansiedade do sistema de despachar imediatamente. Ao invés disso, é necessário que sejamos capazes de ter previsibilidade em geolocalização topológica.

Por exemplo: eu posso ser vizinho de muro do João, mas meu fio vem do poste da esquerda e o dele, do poste da direita. Se o meu poste queima, eu fico sem energia, mas o João continua com luz. Eles são vizinhos geográficos, mas estranhos topológicos. O desafio não é apenas coletar os dados, mas entender a relação que os conectam.

Para que seja possível mapear as “zonas de interferência”, podemos nos guiar por perguntas simples mas com grandes potenciais:

- Das 10 pessoas que ligaram, quantas são alimentadas pelo mesmo Transformador?
 - Estatisticamente, se 5 vizinhos caíram, qual é a probabilidade do defeito estar no fusível da esquina e não nas casas?
 - Se o padrão se repete em ruas paralelas, estamos lidando com um problema de Média Tensão (bairro) ou Baixa Tensão (rua)?
-
- **A Mudança de Paradigma:** Em vez de analisar *quem* está reclamando (o CPF), analisamos *onde* ele está conectado na árvore genealógica da rede. Qual Transformador é o "pai" dessa Unidade Consumidora?
 - **O Diferencial:** Criar uma camada lógica intermediária que intercepta os chamados *antes* de virarem Ordem de Serviço, aplicando algoritmos de Clusterização Espacial.

3. Construtos da Solução (O MVP)

A proposta que traçamos para interceptar os chamados feitos em filas “um por um”, é acionar um sistema de algoritmo que vá tratar os chamados como “agrupamento de incidentes”. Denominamos a proposta como “O Guardião”, no qual refere-se a um sistema automatizado que atua como um filtro inteligente e preditivo antes da Ordem de Serviço ser criada.

A proposta é ser simples, escalável e orientado a eventos reais utilizando a lógica de topologia (quem está ligado em quem), o algoritmo analisa o padrão das reclamações em tempo real. Se ele detecta que um grupo de chamados pertence ao mesmo transformador, ele bloqueia o envio de equipes para as casas e gera um único alerta prioritário.

A solução proposta é um microsserviço preditivo estruturado em três pilares:

3.1. O Mapa Virtual (Digital Twin)

Focamos na estruturação em JSON, onde cada nó da rede conhece seus dependentes diretos. Dessa forma, modelamos a rede elétrica como um Grafo Direcionado. A estrutura completa é pautada no MongoDB (NoSQL), que armazena a hierarquia energética de forma flexível e performática, estabelecendo relações claras de dependência — *quem alimenta quem* — entre Subestações, Transformadores e Unidades Consumidoras. Assim, quando um evento chega, o sistema consulta esse "mapa vivo" no banco para entender a correlação entre os pontos.

3.2. O Motor de Decisão (Core Logic)

Para conseguirmos um resultado otimizado, é preciso seguir a lógica: se mais de três residências da mesma rua abrem chamados, a necessidade de atendimento não é individual, mas sim estrutural e sistêmica. Para que isso seja realizado de forma eficiente, é necessário usar o algoritmo de inferência desenvolvido em Python + NetworkX, integrado ao barramento de eventos (Kafka).

Lógica de Agrupamento: O sistema monitora a densidade de incidentes em janelas de tempo curtas (< 10 min). Se múltiplos pontos (clientes) reportam falha simultânea, o algoritmo identifica o “problema comum” mais provável (transformador) como a causa-raiz.

4. Arquitetura, Escalabilidade e Integração com Legado

O “Guardião” foi desenhado seguindo os princípios de Escalabilidade Horizontal e Segurança, garantindo que a implementação em ambientes corporativos críticos — como Centros de Operação de Distribuição (COD) — seja segura, auditável e não-bloqueante. Afinal, entendemos que a modernização de sistemas legados exige fricção zero: nossa abordagem permite acoplar inteligência preditiva, sem romper processos atuais, propondo um ambiente de trabalho mais dinâmico, onde o analista pode observar padrões com mais conexões e focar em estratégias de previsão de dados e decisões baseadas em eventos de tempo real.

4.1. Integração "Sidecar"

Sabemos que sistemas SCADA e OMS são o coração da operação e não podem sofrer latência ou instabilidade.

- **Como funciona:** O Guardião opera em uma arquitetura de eventos (Event-Driven) via Apache Kafka. Ele atua como um consumidor passivo dos logs em tempo real.
- **Benefício:** O algoritmo processa os dados em paralelo — de forma assíncrona, sem tocar no código-fonte dos sistemas legados e sem impactar a performance do banco de dados principal. Nesse caso, há um desacoplamento total: se o Guardião parar, a operação manual continua sem interrupções, porque eles são computacionalmente independentes.

4.2. Estratégia de Rollout: O "Shadow Mode"

Para garantir a confiabilidade das decisões algorítmicas e realizar a implementação em fases sem gerar riscos à operação vigente, o sistema começa a operacionalização em Modo Sombra:

Fase 1: A Operação

O sistema roda em paralelo ao fluxo atual, atuando como um "observador silencioso".

- **Mecanismo:** O algoritmo consome os eventos do OMS/Kafka em tempo real e processa a árvore de decisão, mas o output (a Ordem de Serviço) é direcionado a um Ambiente de Simulação.
O objetivo de realizar a implementação no Modo Sombra (Shadow Mode) é garantir que o sistema processe o volume de carga real sem latência, sem interferir no despacho das equipes de campo atuais. Reduzindo o risco operacional a zero.

Fase 2: Calibragem de Precisão

Confrontamos as decisões do algoritmo e o histórico real de atendimentos.

- **Métrica de Sucesso:** Não olhamos apenas para a acurácia geral, mas focamos na redução de Falsos Positivos dentro dos chamados, e assim, podemos evitar deslocamentos improdutivos.
- **Análise:** Calibragem dos limiares de confiança (Confidence Thresholds) baseada na topologia específica da região. Nesse aspecto nos direcionamos à Lógica de Calibragem, porque temos dois cenários para a distribuição energética: Rural e Urbana. Ambas precisam de uma atenção especializada na construção do algoritmo.

Fase 3: Operacionalização com Humanos (Human-in-the-Loop)

O sistema entra em produção como uma ferramenta de suporte à decisão (DSS — Decision Support System).

- **Fluxo:** O Guardião analisa o incidente e apresenta ao Controlador do COD uma "Sugestão de Despacho Otimizada" com uma pontuação de confiança. Por exemplo, ao terminar de computar os possíveis erros dentro do grafo, será disparado uma mensagem. Exemplo: *98% de chance de ser o Transformador T-45.*
- **Poder de Decisão:** O humano mantém a autoridade final do despacho, como aceitar "aceitar" ou "recusar" que a operação vá até o transformador localizado, mas já temos o trabalho cognitivo pesado de análise de causa-raiz realizado pelo algoritmo.

Lógica de Calibragem: Para que a operacionalização do software seja sustentável e eficiente, precisamos pensar no modelo preditivo-estatístico para áreas urbanas e rurais. Para cada cliente que abre um chamado dentro do sistema, é identificado um padrão de geolocalização baseado na distribuição de energia através do transformador, ou seja, para o cálculo ser perfeito (ou quase), precisamos estabelecer um padrão de threshold.

Para definir se um incidente é isolado ou sistêmico, o algoritmo calcula em tempo real a taxa de saturação do ativo. Ou seja, calculamos os chamados recebidos, mais os alarmes de sensores, e depois dividimos pelo número total de clientes conectados no transformador.

— Cenário Urbano (Alta Densidade):

Em áreas densamente povoadas, o risco operacional é o "falso positivo". Por exemplo, despachar uma equipe pesada para um disjuntor interno que desarmou em um único prédio. Isso gera custos elevados e deslocamento improdutivo.

- **A Regra:** O algoritmo eleva o *Threshold* (Limiar) para 90%.
- **A Lógica:** Exigimos confirmação estatística robusta. Se um transformador alimenta 100 clientes, o sistema só gera o despacho automático se houver uma saturação massiva de reclamações, confirmando que a falha é na rede de distribuição e não em uma unidade isolada.

— Cenário Rural (Baixa Densidade):

Em áreas onde a densidade demográfica é baixa e as redes são extensas, o problema não

é o volume de dados, mas a dispersão. Manter a régua em 90% aqui tornaria o sistema "cego" para falhas graves.

- **A Regra:** Adaptamos a sensibilidade, reduzindo o *Threshold* para 20% a 30%.
- **A Lógica:** Como os vizinhos são distantes, a ausência de dados massivos não pode impedir o atendimento. Basta que duas unidades vizinhas em uma rede longa abram chamado para que o sistema reconheça o padrão como uma falha estrutural. Como um cabo partido, por exemplo, priorizando a rapidez no restabelecimento de energia.

4.3. Escalabilidade via Particionamento Geográfico

Processar a rede elétrica de um estado inteiro em um único grafo pode ser custoso computacionalmente.

- **A Solução:** Utilizamos *Sharding*. O sistema divide o processamento em microsserviços containerizados (Docker), onde cada instância cuida de uma Regional ou Subestação específica.
- **Resultado:** O sistema é linearmente escalável. Processar 100 mil ou 10 milhões de unidades consumidoras torna-se apenas uma questão de alocar mais *containers*, mantendo a latência de resposta na casa dos milissegundos.
- **A Lógica:** O sistema foi projetado para operar sob uma lógica de encapsulamento total. Cada módulo do software funciona como uma unidade autônoma e autossuficiente. Isso significa que ele carrega consigo todas as dependências necessárias para sua execução, criando um ambiente de processamento interno independente. Ao ser implantado, o sistema não interfere nas configurações do servidor hospedeiro, nem depende de instalações prévias. Essa abordagem garante que a lógica de decisão e os cálculos matemáticos ocorram em um ambiente controlado e blindado, eliminando conflitos com sistemas legados e garantindo que, se houver falha no módulo, o restante da operação da empresa permaneça intacto e seguro.