

MÓDULO 5:

SOFTWARE PARA PROCESSAMENTO COMPLEXO DE EVENTO (CEP) PARA CAUSAS RAÍZES

Execução:



LRI - LABORATÓRIO DE REDES INTELIGENTES
www.lri.ufu.br
UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Alan Petrônio Pinheiro

Coordenador do projeto – UFU/LRI

Execução e pesquisa:

Gabriel Araújo Augustavo

Otávio Del Bianco Reis

1. DATA VERSÃO ORIGINAL 12-3-2025	2. DATA ÚLTIMA ATUALIZAÇÃO 23-5-2025	3. DATA COBERTA MARC/25 ATÉ MAI/25
4. TÍTULO DESTE DOCUMENTO REPORTE TÉCNICO DO SOFTWARE CEP		5a. PROCESSO SEI DO P&D -
		5b. NÚMERO PROJETO P&D
6. AUTOR(ES) GABRIEL ARAUJO AUGUSTAVO - 12111ECP017 OTAVIO DEL BIANCO REIS - 12021ECP008		5c. ETAPA DO PROJETO TODAS
		5d. TIPO DE PRODUTO DOCUMENTAÇÃO TÉCNICA DE SOFTWARE DE DISCIPLINA STR
7. ENDEREÇO AV. JOÃO NAVES DE ÁVILA, 2121, BLOCO 3N - UBERLÂNDIA - MG		8. NÚMERO DO DOCUMENTO TR-01
9. DISTRIBUIÇÃO DESTE DOCUMENTO DISTRIBUIÇÃO ABERTA A TODOS OS INTERESSADOS.		
10. NOTAS COMPLEMENTARES -		
11. RESUMO ESTE DOCUMENTO DESCREVE A MODELAGEM DOS ELEMENTOS DE SOFTWARE CEP QUE ANALISA PACOTES DE DADOS PARA DETECTAR PADRÃO DE FALHA EM DETERMINADA ÁREA. TODOS MÓDULOS CONSTITUEM UM SISTEMA SUPERVISÓRIO PARA O SETOR ELÉTRICO.		
12. PALAVRAS-CHAVE P&D; IOT; SISTEMA EM TEMPO REAL, CEP, SISTEMA SUPERVISÓRIO, ANÁLISE DE PADRÃO.		
13. CLASSIFICAÇÃO SEGURANÇA: ABERTA	14. NÚMERO DE PÁGINAS -	15. NOME DO RESPONSÁVEL PRINCIPAL E CONTATO GABRIEL ARAUJO AUGUSTAVO. EMAIL: gabriel.augustavo@ufu.br TELEFONE: (34) 9 9807-2008

HISTÓRICO DE VERSÕES DESTE TR

Tabela 1 – Histórico de versões deste reporte técnico.

Versão	Data	Modificações
1.0	março/2024	<ul style="list-style-type: none">• Principais elementos de projeto• Requerimentos básicos• Modelagem de pacotes e fluxo de pacotes• Interfaces básicas
2.0	abril/24	<ul style="list-style-type: none">• Correções gerais no documento• Melhoria na máquina de estados

SUMÁRIO

RESUMO GERAL	4
1 – INTRODUÇÃO: VISÃO GERAL DA SOLUÇÃO	4
1.1 – PROPÓSITO E ESCOPO	4
1.2 – PRODUTO: PERSPECTIVAS E FUNÇÕES	5
1.3 – RESTRIÇÕES DO PRODUTO E CONSIDERAÇÕES	6
2 – REQUISITOS	6
2.1 – CENÁRIOS DE USO	6
2.2 – REQUISITOS E VALIDAÇÃO	7
2.3.1 – VERSIONAMENTO	8
2.3.2 – DIVISÃO DE RESPONSABILIDADES	8
2.4 – ELEMENTOS DE PROJETO	9
2.4.1 – Máquina de estados	9
3 – MODELAGEM	9
3.1 – BLOCOS DE ELEMENTOS PRINCIPAIS	9
3.2 – FLUXO GERAL E FORMATO DOS PACOTES	10
3.3 – Overview do algoritmo DBSCAN	10

RESUMO GERAL

Este reporte técnico aborda os elementos do sistema de software que compõe a solução de sistema supervisorio para o setor elétrico. Em específico, ele também dá suporte a aplicação de proteção de sistemas elétricos de potência e automação de processos. Todas as aplicações voltadas exclusivamente para o setor elétrico. Em específico, o software aqui posto é responsável por realizar a identificação de anomalia em uma quantidade volumosa de dados. Vale destacar que este sistema envolve comunicação tanto com outros softwares e todos constituem um sistema maior: o supervisorio. Outros reportes técnicos da solução podem ser consultados para informações sobre a solução como um todo.

1 – Introdução: visão geral da solução

1.1 – Propósito e escopo

O propósito do sistema é realizar a análise de dados recebidos por uma unidade de telemetria de uma cidade, determinar uma razão complexa associada a um evento e comunicar a um módulo de monitoramento visual. Logo, o foco desta aplicação é a análise em tempo real dos fluxos de dados, o CEP permite correlacionar eventos dispersos para detectar padrões e anomalias que podem apontar para a origem de falhas ou problemas críticos. Tal sistema, será chamado designado processamento de eventos complexos (CEP) para causas raízes.

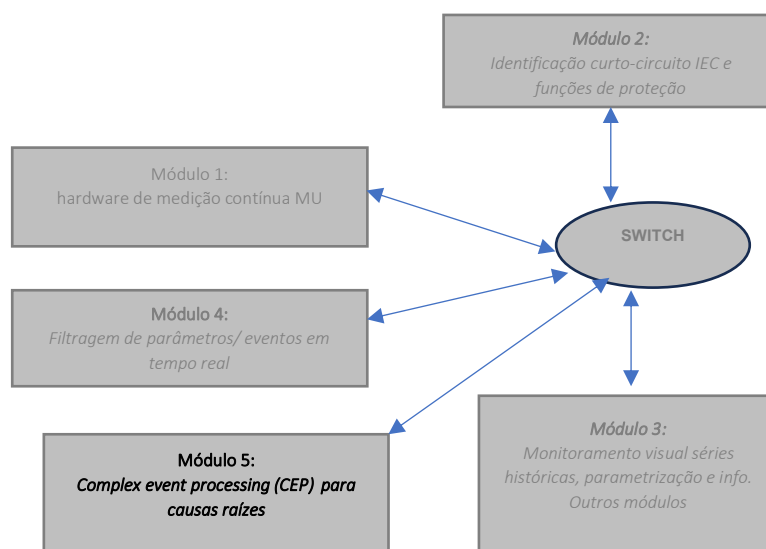


Figura 1.1.1: Visão geral de escopo.

De acordo com a **Figura 1.1.1**, temos o seguinte:

- **"MU"** – Measurement Unit: é uma unidade de medida que obtém dados de corrente em determinado espaço de tempo contínuo.

- **Identificação curto-circuito IEC e funções de proteção:** Monitorar os pacotes de medição recebidos dos módulos para identificar, em tempo real, a ocorrência de curtos-circuitos.
- **Monitoramento visual séries históricas, parametrização:** fornece uma interface gráfica que integre e visualize as informações dos demais módulos.
- **Filtragem de parâmetros/eventos em tempo real:** permitir ao usuário definir regras para a identificação de eventos a partir dos pacotes.
- **Complex event processing (CEP) para causas raízes:** focado em uma planta de energia que abrange uma cidade ou região extensa, esse módulo processa dados de telemetria para identificar eventos.

1.2 – Produto: perspectivas e funções

Este sistema tem como principal função informar ao módulo de monitoramento visual a ocorrência de um evento. Para cumprir a função, ele deve ser capaz de:

1. Processar rajadas de dados telemétricos.
2. Determinar se existe um padrão contido que revela uma falha ou problema crítico a ser reportado.

Para tal função, temos os principais elementos essenciais a esse funcionamento:

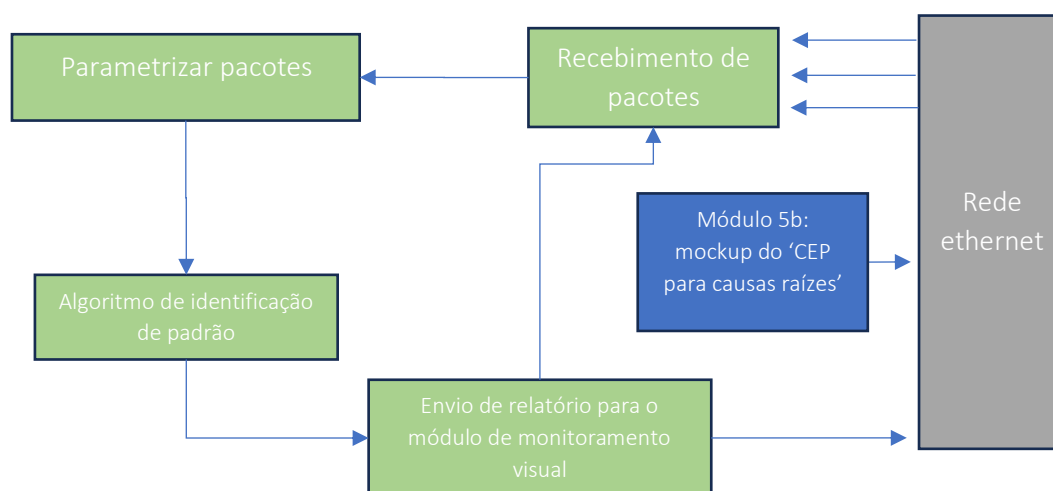


Figura 1.2.1: Principais elementos de projeto de 1 MU.

Para entender o sistema, começemos a análise observando a [Figura 1.2.1](#). Com base nisto, descreve-se os elementos:

- **Recebimento de pacotes:** os pacotes UDP são recebidos em rajadas com intervalo de 1 minuto.
- **Parametrizar pacotes:** os pacotes recebidos devem ser processados e parametrizados para aplicação do algoritmo com os dados extraídos.
- **Algoritmo de identificação de padrão:** os dados recebidos devem ser clusterizados com a aplicação do algoritmo DBSCAN.
- **Envio de relatório para o módulo de monitoramento visual:** este módulo gera um relatório a ser enviado ao módulo de monitoramento visual.
- **Módulo 5b: mockup do CEP para causas raízes:** realiza o envio de rajadas de dados de telemetria.

1.3 – Restrições do produto e considerações

A solução geral aqui prevista foi testada para condições específicas e nestas, foram identificadas as seguintes restrições ou limitações para os quais o sistema proposto não foi projetado para atuar. Estas restrições e limitações são mostradas na tabela da sequência.

Tabela 1.3.1: Restrições e limitações previstas para sistema.

Nº	Restrição/limitação	Descrição/detalhamento
1	Sensibilidade a rajadas de pacotes	O módulo recebe dados (no exemplo da cidade de Uberlândia, 300.000 pacotes) de telemetria a cada 1 minuto de cada uma das unidades consumidores de energia.
2	Tempo de emissão do alerta	O tempo para processamento de alertas pode variar consideravelmente devido ao fluxo de pacotes.
3	Precisão na identificação de eventos	O algoritmo DBSCAN requer um ajuste fino de seus parâmetros para evitar falsos positivos na detecção de falhas.
4	Conectividade e Comunicação	O sistema depende de uma infraestrutura de rede com fluxo contínuo para receber e enviar pacotes JSON via UDP, e qualquer falha na conectividade pode impactar sua eficácia.

2 – Requisitos

2.1 – Cenários de uso

Os seguintes cenários foram identificados para este sistema.

- a) **Operação em condições de não ocorrência de anomalia:** quando não há nenhum evento, o sistema recebe os pacotes, realiza o processamento dos dados, aplica o algoritmo de detecção de padrão e não emite nenhum alerta.

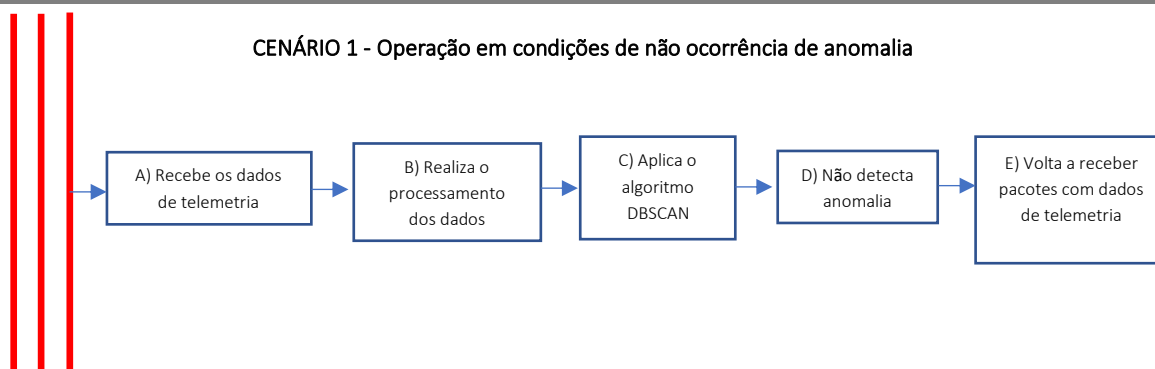


Figura 2.1.1: Cenário de aplicação.

- b) **Operação em condições de ocorrência de anomalia:** quando uma determinada área de coordenadas geográficas emite pacotes de falha em massa, o sistema detecta a ocorrência de uma anomalia, transforma em um único evento e gera um único relatório de alerta ao módulo de monitoramento visual.

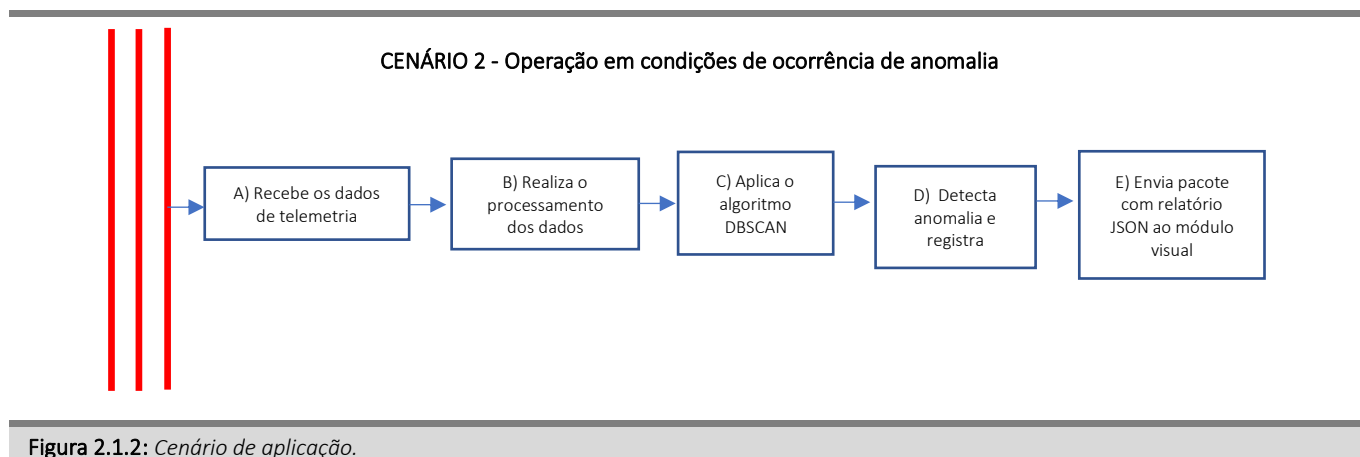
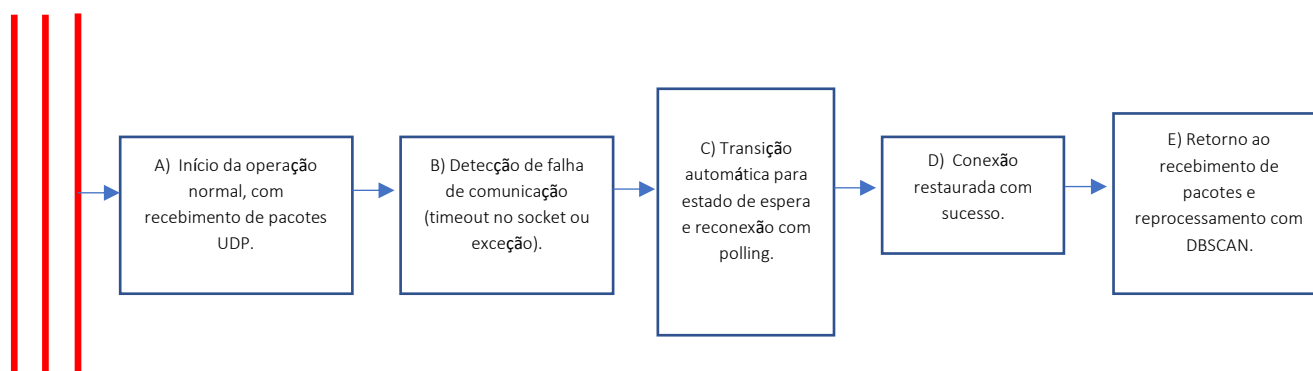


Figura 2.1.2: Cenário de aplicação.

- c) **Operação com falha de rede e recuperação automática:** este cenário representa uma situação: onde o módulo CEP perde temporariamente a conectividade de rede, interrompendo o recebimento de pacotes via UDP. O sistema detecta a falha e entra em modo de retentiva, mantendo-se em espera com tentativas periódicas de reconexão via polling. Após a restauração da conexão, o sistema retorna automaticamente ao estado de escuta ativa, reestabelecendo o fluxo de pacotes e reiniciando o processo de detecção de padrões normalmente, sem perda de estado interno.



2.2 – Requisitos e validação

Com base nas entrevistas com os clientes, equipe de engenharia e avaliações de cenário de uso, desenvolveu-se na sequência a seguinte lista de requerimentos, vista na tabela da sequência.

Tabela 2.2.1: Mapa de requerimentos.

Classe/Componente	Nº req.	Requisito	Origem requisito	Prio	Tipo validação
-------------------	---------	-----------	------------------	------	----------------

1 - Robustez	1.1	Suportar uma rajada de pacotes UDP com variação de 20% do fluxo normal	Tolerância a flutuações da rede	1	Simulação de rajada de pacotes.
	1.2	O sistema deve ter tolerância de até 10% de perda de pacotes	Tolerância a flutuações da rede	1	Simulação de fluxo de pacotes.
2 - Funcional	2.1	O sistema deve analisar os pacotes e identificar causas raízes de eventos básicos de queda de energia	Processamento de Eventos	1	Implementação de ocorrência de quedas de energia.
	2.2	O CEP deve criar e enviar um único alarme quando detectar um evento que afete múltiplas unidades consumidoras	Geração de Alertas	1	Ajuste nos parâmetros do algoritmo DBSCAN.
	2.3	O sistema deve ser capaz de registrar o envio de alarmes	Registro de Alertas	2	Simulação de eventos em tempos diferentes.
3 – Não funcionais	3.1	A identificação e agrupamento dos eventos devem ocorrer com latência mínima para envio de pacote UDP com relatório JSON	Tempo de resposta tolerável	1	Teste do tempo total médio da identificação de um alerta.

2.3.1 – Versionamento

Os recursos do software são distribuídos em versões conforme estimado pela tabela na sequência.

Tabela 2.3.1: Tabela de recursos do sistema e versão.

Versão	Recurso
1.0 (mar/24)	<ul style="list-style-type: none">• Visão geral da solução• Principais elementos de projeto• Requerimentos básicos• Modelagem de pacotes e fluxo de pacotes
2.0 (abril/24)	<ul style="list-style-type: none">• Correções gerais• Melhoria na máquina de estados

2.3.2 – Divisão de responsabilidades

Tabela 2.3.2: Divisão geral de responsabilidades do projeto.

Responsável	Partição
Gabriel Araújo Augustavo	<ul style="list-style-type: none">- Implementação do receptor UDP e fila concorrente de entrada de pacotes.- Desenvolvimento do fluxo de envio de alertas (JSON).- Criou a interface gráfica- Envio de clusters para o módulo 3- Escrita da documentação técnica das seções 1.1, 1.2, 2.1, 2.2 e 3.2.
Otávio Del Bianco Reis	<ul style="list-style-type: none">- Implementação do paralelismo com múltiplas threads.- Aplicação do algoritmo DBSCAN.- Criação do modelo de máquina de estados e sua simulação.- Escrita da documentação das seções 1.3, 2.4, 3.1 e 3.3.

Ambos os membros participaram da **revisão geral do código-fonte e do documento**, além de definirem conjuntamente os critérios de validação, modelagem dos pacotes e integração com os demais módulos do sistema supervisorio.

2.4 – Elementos de projeto

2.4.1 – Máquina de estados

Baseado nos cenários identificados e requerimentos construídos, tem-se a seguinte proposição para a máquina de estados:

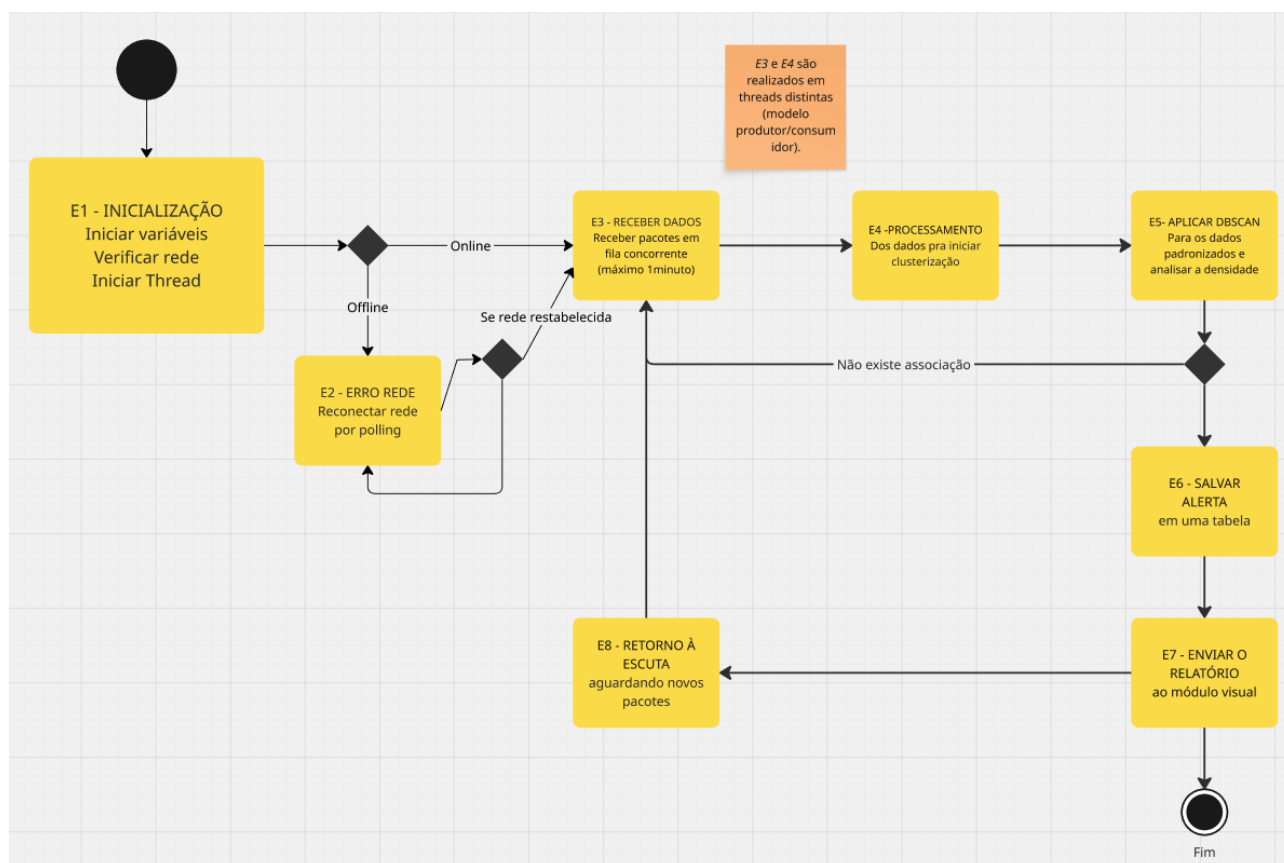


Figura 2.4.2.1: Máquina de estados de uma MU.

3 – Modelagem

3.1 – Blocos de elementos principais

Na sequência é mostrado um conjunto de diagramas de blocos para exemplificar a arquitetura do sistema. Cada bloco é um objeto e estes são os principais objetivos previstos na solução. As setas indicam o fluxo das informações.

Figura 1.2.1: Componentes básicos do Digital Twins.

3.2 – Fluxo geral e formato dos pacotes

Primeiro, o sistema recebe por meio de um thread, dados telemétricos de uma cidade (no exemplo Uberlândia), com em média 300 mil dados. Os pacotes recebidos para processamento possuem uma estrutura JSON com os seguintes campos:

Tabela 3.2.1: *Formato do pacote de dados recebido*

Campo	Valores	Significado
Lat	string	Coordenada de latitude geográfica da ocorrência
Long	string	Coordenada de longitude geográfica da ocorrência
codErro	int	Identificador do tipo de erro recebido

Assim, os dados são colocados em uma estrutura chamada fila concorrente pela thread receptora, aqui denominada de *produtora*. Após o recebimento de tais dados em até 60 segundos, estes são extraídos por threads consumidoras, que possuem buffers próprios. No final do processamento dos dados, os buffers servem de entrada para o funcionamento do algoritmo DBSCAN, que realiza a *clusterização*, que é o processo de separação dos dados em conjuntos e analisar a densidade de cada conjunto. Então, caso exista um padrão relevante, este será armazenado em uma tabela contendo os seguintes campos:

Tabela 3.2.2: *Formato do pacote de alerta cujo evento foi detectado*

Campo	Valores	Significado
Lat	string	Coordenada de latitude geográfica média obtida pela clusterização
Long	string	Coordenada de longitude geográfica média obtida pela clusterização
codErro	int	Identificador do tipo de erro do alerta
tempoAlerta	string	Data temporal contendo o tempo de quando a ocorrência de evento foi detectada

Após o salvamento das informações de alerta da anomalia, deve-se enviar o pacote com JSON dessas informações ao módulo 3 de monitoramento visual.

3.3 – Overview do algoritmo DBSCAN

O DBSCAN (Density-Based Spatial Clustering of Applications with Noise) é um algoritmo de agrupamento baseado em densidade. Ele identifica clusters (grupos de dados) em um conjunto de pontos e separa os pontos considerados ruído (outliers), sem exigir que o número de clusters seja definido previamente. No caso desse módulo, o DBSCAN será usado para identificar falhas elétricas em diferentes locais analisando 300.000 pacotes de telemetria por minuto. Ele analisa quais falhas estão próximas umas das outras, agrupando eventos em clusters de falha e se muitos eventos estiverem concentrados em uma mesma região, o sistema gera um alerta consolidado.

Portanto, o DBSCAN foi escolhido para o Módulo 5 porque é um algoritmo altamente eficiente na identificação de falhas elétricas em larga escala, permitindo detectar padrões complexos sem necessidade de definir um número fixo de clusters. Sua abordagem baseada em densidade possibilita distinguir falhas realmente relevantes de eventos isolados, tratando esses últimos como ruído e evitando falsos alarmes. Isso é essencial para garantir que apenas incidentes significativos sejam reportados, reduzindo o volume de notificações desnecessárias e melhorando a precisão do sistema. Além disso, o DBSCAN é capaz de se adaptar

dinamicamente às mudanças no comportamento da rede elétrica, pois sua metodologia permite que novos padrões de falhas sejam identificados sem a necessidade de reconfiguração manual.

O DBSCAN foi paralelizado com base no **particionamento dos dados por região geográfica**, processando cada subgrupo em **threads separadas**, o que permite manter o tempo de resposta baixo mesmo com grandes volumes. Após o processamento, os resultados são fundidos, garantindo integridade dos alertas.

Segue abaixo um protótipo do código em forma de desenho técnico:

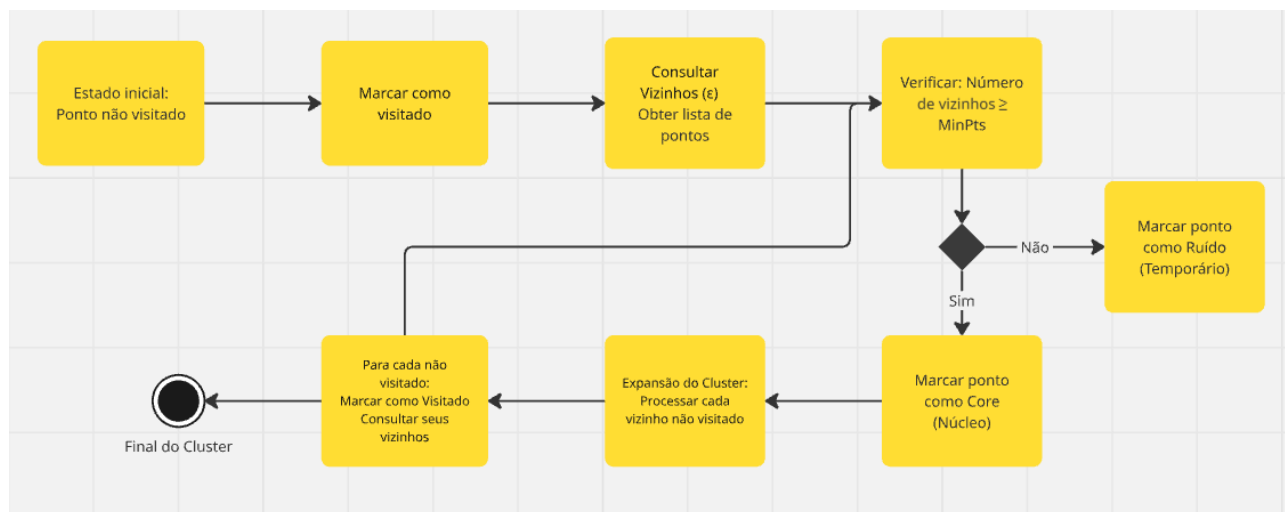


Figura 3.3.1: Protótipo do funcionamento do algoritmo DBSCAN.