

# MÓDULO 4:

## SOFTWARE DE FILTRAGEM DE PARÂMETROS E EVENTOS EM TEMPO REAL DE SUBESTAÇÕES DE ENERGIA



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Alan Petrônio Pinheiro  
Professor da Disciplina – UFU/LRI

Execução:  
Bruna Isabela de Oliveira  
Bruno Pavan Ciriani da Silva

1. DATA VERSÃO ORIGINAL 28-09-2024	2. DATA ÚLTIMA ATUALIZAÇÃO 13-11-2024	3. DATA COBERTA SET/24 ATÉ NOV/24
4. TÍTULO DESTE DOCUMENTO REPORTE TÉCNICO DA APLICAÇÃO MU		5a. PROCESSO SEI DO P&D -
		5b. NÚMERO PROJETO P&D -
6. AUTOR(ES) BRUNA ISABELA DE OLIVEIRA - 12011ECP004 BRUNO PAVAN CIRIANI DA SILVA - 12011EC012		5c. ETAPA DO PROJETO TODAS
		5d. TIPO DE PRODUTO DOCUMENTAÇÃO TÉCNICA DE SOFTWARE DE DISCIPLINA SCTR
7. ENDEREÇO AV. JOÃO NAVES DE ÁVILA, 2121, BLOCO 1E - UBERLÂNDIA - MG		8. NÚMERO DO DOCUMENTO  <b>TR-01</b>
9. DISTRIBUIÇÃO DESTE DOCUMENTO DISTRIBUIÇÃO ABERTA A TODOS OS INTERESSADOS.		
10. NOTAS COMPLEMENTARES -		
11. RESUMO ESTE DOCUMENTO DESCREVE A MODELAGEM DA APLICAÇÃO QUE TEM COMO OBJETIVO FILTRAR PARÂMETROS E EVENTOS DE COMPONENTES DE SOFTWARE/HARDWARE REATRATADOS POR OUTROS REPORTES TÉCNICOS. TODOS OS MÓDULOS CONSTITUEM UM SISTEMA SUPERVISÓRIO PARA O SETOR ELÉTRICO.		
12. PALAVRAS-CHAVE P&D, IOT, SISTEMA EM TEMPO REAL, SISTEMA SUPERVISÓRIO, IDENTIFICAÇÃO DE EVENTOS ELÉTRICOS EM SUBESTAÇÕES DE ENERGIA.		
13. CLASSIFICAÇÃO SEGURANÇA:  <b>ABERTA</b>	14. NÚMERO DE PÁGINAS  -	15. NOME DO RESPONSÁVEL PRINCIPAL E CONTATO ALAN PETRÔNIO PINHEIRO. EMAIL: alan_petronio@yahoo.com.br. TELEFONE: (34) 3239-4701

# HISTÓRICO DE VERSÕES DESTE TR

Tabela 1 – Histórico de versões deste reporte técnico.

Versão	Data	Modificações
1.0	setembro/2024	<ul style="list-style-type: none"><li>• Visão geral da solução</li><li>• Principais elementos de projeto</li><li>• Requerimentos básicos</li><li>• Modelagem de pacotes e fluxo de pacotes</li></ul>
2.0	novembro/2024	<ul style="list-style-type: none"><li>• Distribuição de responsabilidades da equipe</li><li>• Correções gerais</li></ul>

## 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 .....	5
2 – REQUISITOS .....	6
2.1 – CENÁRIOS DE USO .....	6
2.2 – REQUISITOS E VALIDAÇÃO .....	7
2.3 – VERSIONAMENTO .....	8
2.4 – DISTRIBUIÇÃO DE RESPONSABILIDADES.....	8
2.5 – ELEMENTOS DE PROJETO .....	9
2.5.1 – Máquina de estados.....	9
3 – MODELAGEM .....	10
3.1 – BLOCOS DE ELEMENTOS PRINCIPAIS .....	10
3.2 – FLUXO GERAL DE PACOTES .....	10

## RESUMO GERAL

Este reporte técnico aborda os elementos do software(módulo) de “Filtragem de Parâmetros e Eventos em Tempo Real” que compõe a solução do sistema supervisório para o setor elétrico. Além disso, 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 um mecanismo de filtragem de medições de corrente elétrica com base em regras de identificação inseridas pelo usuário do sistema. Vale destacar que este sistema envolve comunicação tanto com outros softwares quanto hardwares e ambos constituem um sistema maior: o supervisório. 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 fazer a leitura constante de medições de corrente vindas de uma rede Ethernet, originadas do módulo de hardware “MU – Measurement Unit”, onde vários hardwares estão conectados. Esses dados são filtrados por meio de regras arbitrárias, definidas dinamicamente pelos usuários. O sistema identifica eventos com base nos dados processados, mas os pacotes UDP contendo relatórios em formato JSON só são enviados para o módulo de monitoramento visual quando há uma mudança de estado (ativação ou desativação) de um evento. Ou seja, se um evento já estiver ativo, nenhum pacote será enviado novamente até que ele seja desativado e, em seguida, reativado.

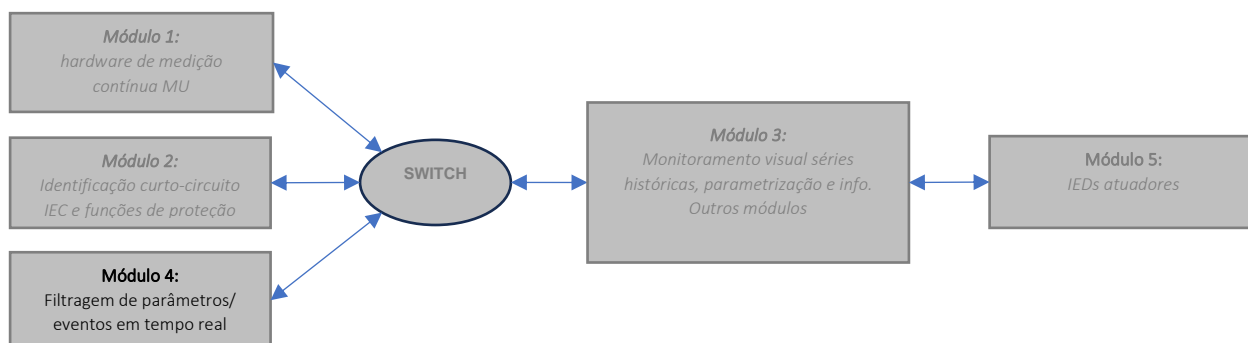


Figura 1.1.1: Visão geral de escopo.

A saber:

- 1) Módulo (hardware) de medição contínua - MU;
- 2) Módulo de identificação de curto-circuito e “IEDs atuadores”, que trabalham em complemento;
- 3) Módulo de monitoramento visual de séries históricas;
- 4) Módulo de filtragem de parâmetros e eventos em tempo real;
- 5) Módulo de complex event processing (CEP) para causas raízes.

## 1.2 – Produto: perspectivas e funções

Com base no propósito apresentado na seção 1.1, o módulo em questão tem como principais funções:

- 1) Permitir que o usuário crie eventos dinâmicos;
- 2) Fazer o processo de identificação de mudanças no estado dos eventos (ativação/desativação) com base no recebimento dos dados do módulo 1;

Emitir um relatório em formato de pacote JSON para o módulo 3.

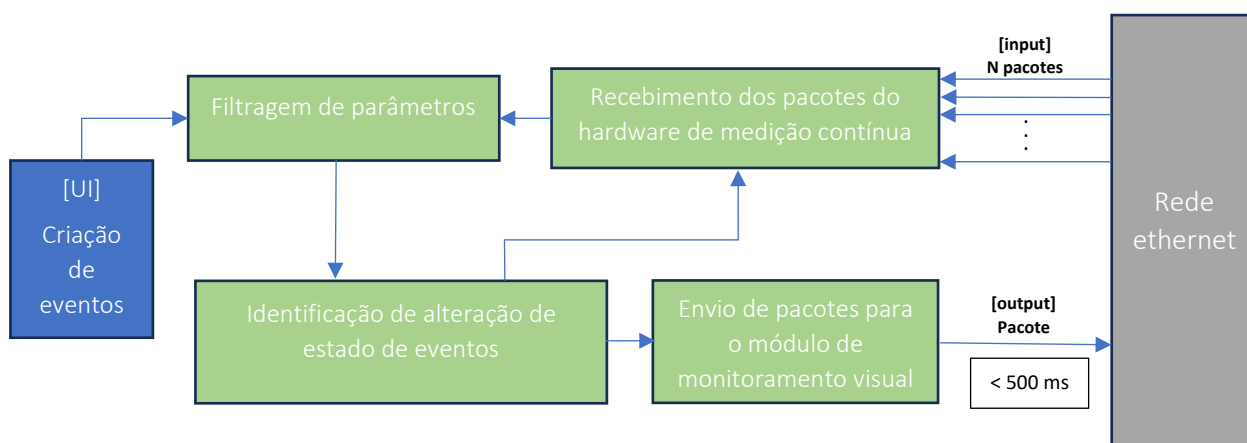


Figura 1.2.1: Principais elementos do projeto.

Para entender o sistema, começemos a análise observando a **Figura 1.2.1**. Com base nisto, descreve-se os elementos:

- **Recebimento dos Pacotes do Hardware de Medição Contínua:** Os pacotes UDP são lidos em polling e com um intervalo de recebimento de pelo menos 10 ms.
- **Criação de Eventos:** Interface onde regras são criadas. Para isto, o usuário pode inserir n regras, dinamicamente, onde deve identificar: (i) o parâmetro n (dentre as opções de variáveis produzidas pelo módulo 1 em seus pacotes), (ii) o operador de comparação (<, = ou >) e o valor numérico.
- **Filtragem de Parâmetros:** A filtragem é feita com base no input do usuário, determinando se as medições de corrente ( $I_A$ ,  $I_B$  e  $I_C$ ) se enquadram nas fórmulas definidas nos filtros.
- **Deteção de Alteração de Estado de Evento:** O sistema verifica se houve uma alteração de estado, seja uma ativação ou desativação do evento. Relatórios são gerados e enviados apenas quando uma mudança de estado é detectada, evitando envios repetitivos para eventos que permanecem inalterados.
- **Envio de Pacotes para o Módulo de Monitoramento Visual:** O pacote de envio é composto e enviado ao módulo 3 quando uma alteração de estado é detectada. O envio ocorre dentro de um tempo máximo de 500 ms após a detecção da alteração.



## 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.

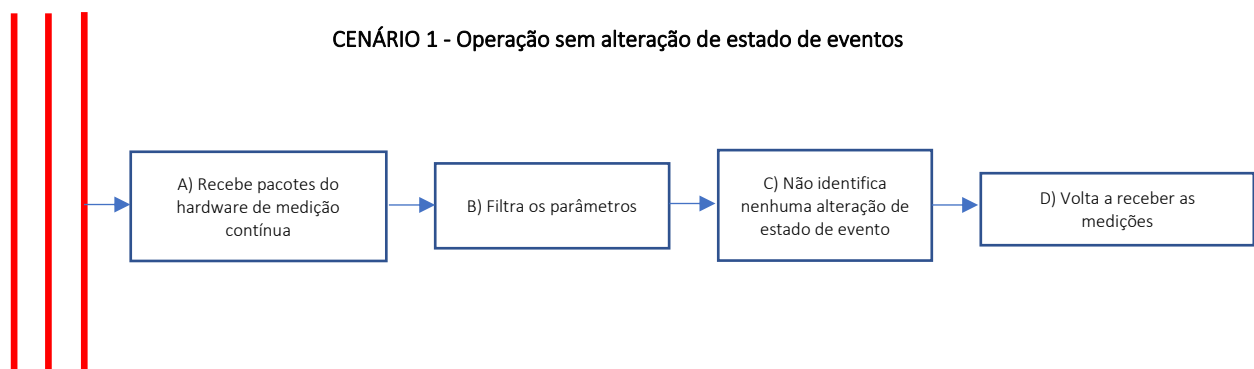
Nº	Restrição/limitação	Descrição/detalhamento
1	Intervalo de polling	O sistema deve realizar a leitura dos pacotes com um intervalo mínimo de 10 ms, garantindo que as medições sejam atualizadas com precisão e possam ser processadas imediatamente em caso de grandes variações.
2	Dependência de Conectividade UDP	A confiabilidade na entrega de pacotes depende da qualidade da rede e da conexão, especialmente em ambientes com alta latência ou perda de pacotes.
3	Gerenciamento de Conexões com Múltiplos IEDs	O sistema deve gerenciar múltiplos IEDs simultaneamente, o que exige um gerenciamento eficiente de threads para manter a performance em nível adequado.
4	Deteção de Alteração de Estado	A detecção de alterações de estado depende da precisão na leitura e no processamento dos dados em resposta a flutuações rápidas.

## 2 – Requisitos

### 2.1 – Cenários de uso

Os seguintes cenários foram identificados para este sistema.

- a) **Cenário 1 – operação sem alteração de estado de eventos:** nele, a filtragem de parâmetros não detecta nenhuma alteração de estado em eventos previamente ativos. O sistema continua monitorando as medições, aguardando novas condições que possam resultar em ativação ou desativação de eventos.



**Figura 2.1.1:** Cenário de aplicação.

- b) **Cenário 2 – operação em ativação de evento:** nele, a filtragem de parâmetros identifica a ocorrência de um evento, com uma medida de corrente se enquadrando nas fórmulas definidas pelo usuário. O sistema verifica se houve uma alteração de estado (ativação). Se sim, um pacote é enviado para o módulo de monitoramento visual.

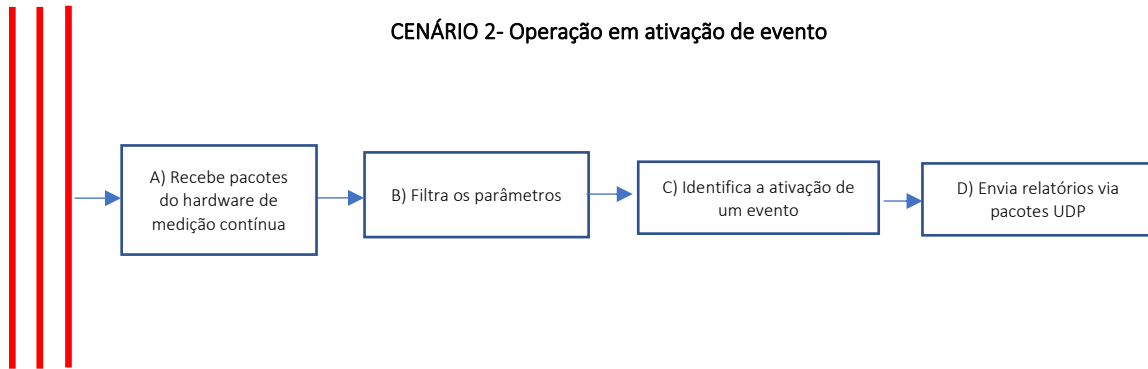


Figura 2.1.2: Cenário de aplicação.

- c) **Cenário 3 – operação em desativação de evento:** nele, a filtragem de parâmetros não identifica um novo evento, mas detecta que um evento previamente ativo foi desativado (a medida de corrente não se enquadra mais nas fórmulas). O sistema então envia um pacote para o módulo de monitoramento visual, informando a desativação.

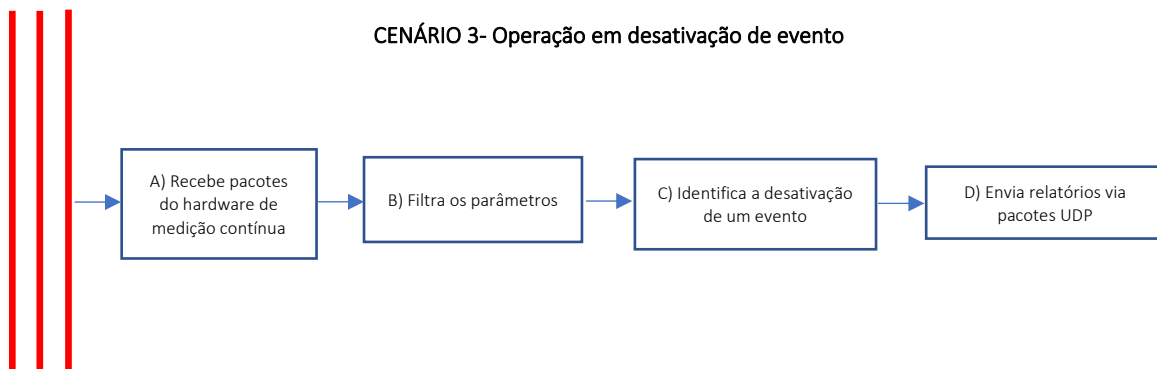


Figura 2.1.3: Cenário de aplicação.

## 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	O sistema deve continuar operando com perda de até 10% dos pacotes	Evitar falhas de comunicação entre módulos	1	Simulação de perda de pacotes em rede sob estresse.
	1.2	O sistema deve tolerar picos de carga sem afetar o desempenho (até 150% do normal).	Requisitos de tolerância a falhas	1	Testes de estresse com picos simulados de carga além da capacidade nominal.
	1.3	O sistema deve retomar operação automaticamente após falha de comunicação.	Requisitos de confiabilidade do sistema	1	Testes de failover com interrupção de rede e recuperação automática.
2 - Funcional	2.1	O sistema deve permitir a criação dinâmica de regras de filtragem pelo usuário.	Flexibilidade operacional	1	Testes de interface e criação de cenários reais para usuários finais.
	2.2	O sistema deve identificar eventos com base no recebimento dos dados de corrente ( $I_A$ , $I_B$ e $I_C$ ) do módulo 1.	Requisitos de monitoramento de corrente	1	Validação em cenários com simulações de diferentes correntes enviadas pelo módulo 1.
	2.4	O sistema deve detectar e enviar pacotes UDP ao módulo de monitoramento visual ao identificar alteração de estado.	Normas de operação em redes críticas	1	Validação por testes automatizados de eventos dinâmicos.
3 – Não funcionais	3.1	A leitura de pacotes deve ocorrer com um intervalo mínimo de 10 ms.	Norma IEC para proteção de sistemas	1	Teste de tempo de resposta e desempenho sob condições normais e estressantes.

## 2.3 – 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 (set/24)	<ul style="list-style-type: none"><li>Visão geral da solução</li><li>Principais elementos de projeto</li><li>Requerimentos básicos</li><li>Modelagem de pacotes e fluxo de pacotes</li></ul>
2.0 (nov/24)	<ul style="list-style-type: none"><li>Distribuição de responsabilidades da equipe</li><li>Correções gerais</li></ul>

## 2.4 – Distribuição de responsabilidades

As responsabilidades e contribuições dos membros da equipe são distribuídas conforme as funções e tarefas atribuídas, conforme descrito na tabela a seguir.

Tabela 2.4.1: Tabela de distribuição de responsabilidades da equipe e atribuições individuais.

Membro	Atribuições
Bruna	<ul style="list-style-type: none"><li>Elaboração dos documentos de requisitos do módulo</li><li>Revisão conjunta do código</li></ul>
Bruno	<ul style="list-style-type: none"><li>Desenvolvimento do código do projeto</li><li>Revisão conjunta do código com validação de requisitos</li></ul>



## 2.5 – Elementos de projeto

### 2.5.1 – Máquina de estados

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

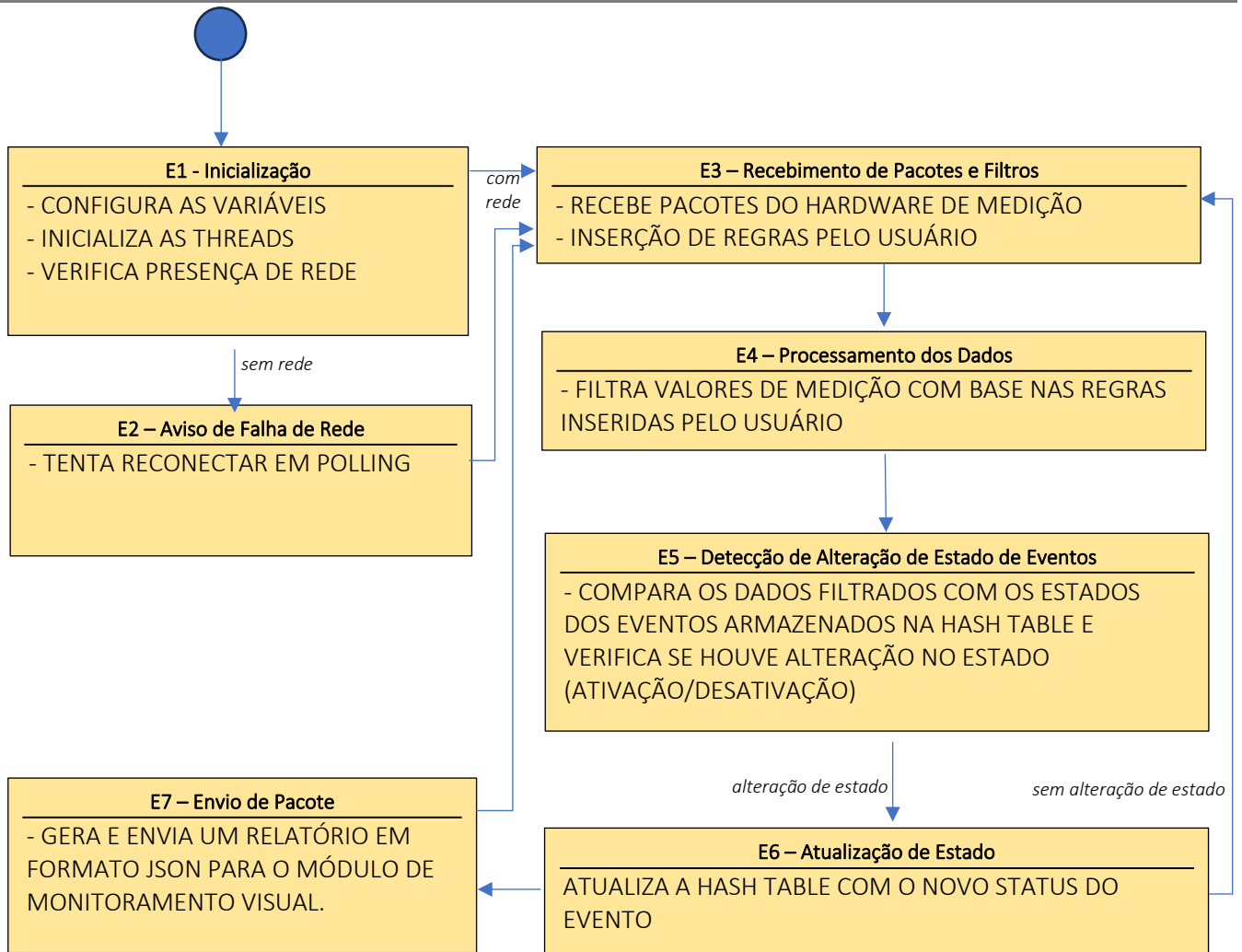


Figura 2.5.1.1: Máquina de estados do software.

## 3 – Modelagem

### 3.1 – Blocos de elementos principais

Na sequência é mostrado um conjunto de diagramas de blocos para exemplificar a arquitetura do sistema.

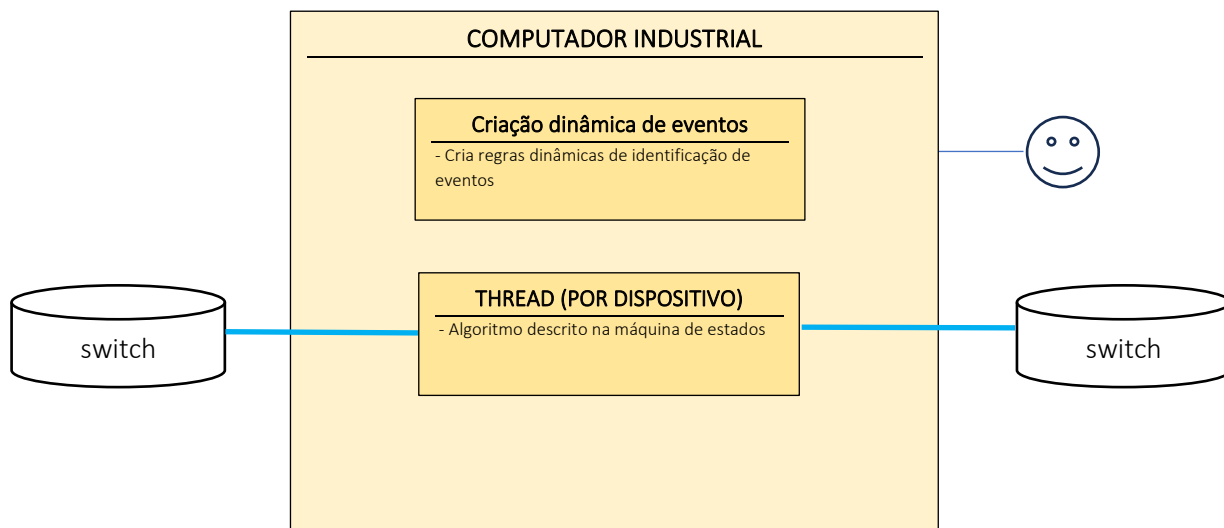
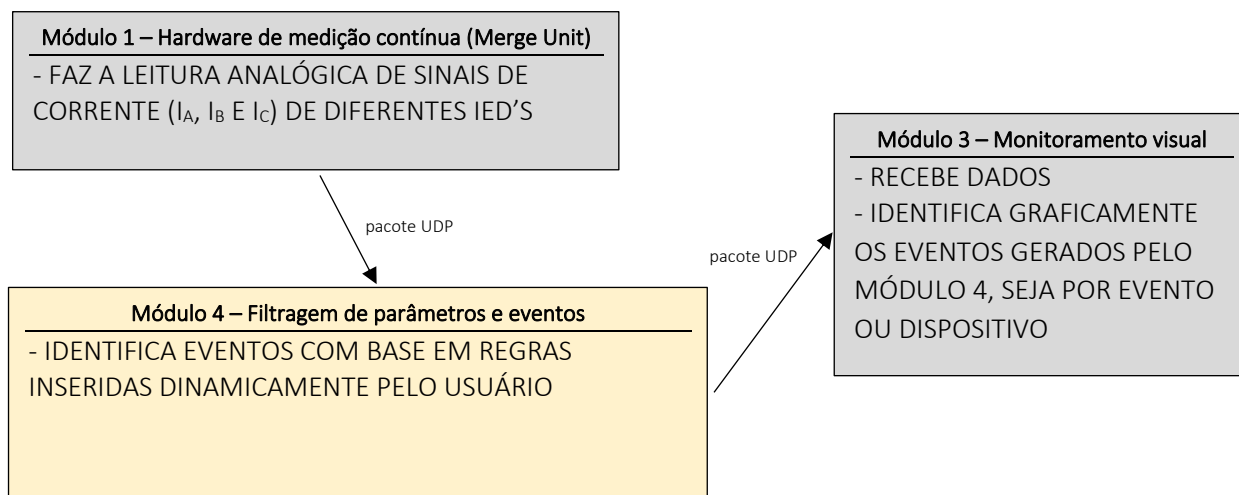


Figura 3.1.1: Componentes básicos do módulo.

### 3.2 – Fluxo geral de pacotes

A figura na sequência ilustra resumidamente os pacotes que são trocados, em diferentes circunstâncias, entre o módulo de filtragem de parâmetros e eventos e demais módulos.





**Figura 3.2.1:** Diagrama do fluxo geral de pacotes

Como observado, o módulo 1 realiza a leitura contínua de sinais de corrente ( $I_A$ ,  $I_B$  e  $I_C$ ) e envia dados em formato JSON via UDP broadcasting para a rede. Esse módulo, disponível na forma de hardware (IED) ou como software, envia medidas em intervalos regulares de pelo menos 10ms.

O módulo 4 recebe os pacotes do módulo 1 e aplica regras de filtragem definidas pelo usuário para identificar eventos específicos, como quando a corrente em uma fase excede um determinado valor. O módulo atualiza o estado dos eventos em uma hash table – assim como é mostrado na **Tabela 3.2.1.1** – conforme necessário. Somente quando ocorre uma ativação ou desativação de um evento, o módulo 4 emite um relatório em formato JSON para o módulo 3 – detalhado na **Tabela 3.2.1.2**. Esse envio é realizado dentro de um tempo máximo de 500 ms após a detecção da alteração, garantindo que o módulo 3, responsável pelo monitoramento visual, receba apenas informações relevantes e em tempo real sobre os eventos gerados, permitindo uma visualização integrada e dinâmica dos dados de todos os módulos.

**Tabela 3.2.1.1:** Hash table para armazenamento do estado dos eventos por dispositivo.

Tabela 3.2.1.1.1: Hash table para armazenamento do estado dos eventos por dispositivo.						
Campo	Valores	Significado				
<device_id>	list	Identificação do IED (Intelligent Electronic Device)  Contém um array que tem o seguinte formato: <table><tr><td>filter</td><td>string</td></tr><tr><td>state</td><td>bool</td></tr></table>	filter	string	state	bool
filter	string					
state	bool					

A tabela acima descreve a estrutura dos dados armazenados na hash table, onde a chave é o próprio `<device_id>` que representa cada dispositivo (IED). Para cada dispositivo, são armazenados múltiplos eventos, sendo que cada evento é representado por dois campos: *filter* e *state*. O campo *filter* define a condição ou critério de monitoramento, como " $I_A > 2$ ", enquanto o campo *state* indica o estado do evento, que pode ser True (ativo) ou False (inativo).

É importante destacar que os campos *filter* e *state* são fixos, mas os valores associados a eles representam os critérios e o estado de cada evento. A chave `<device_id>` é única para cada dispositivo, e para cada um desses dispositivos, podem existir múltiplos eventos, com seus respectivos filtros e estados. Esses dados estão organizados de forma hierárquica, onde cada dispositivo contém sua lista de eventos, com o filtro correspondente e o estado do evento.

**Tabela 3.2.1.2:** Formato do pacote de dados gerado pelo módulo 4.

Campo	Valores	Significado
id	int	Identificação do IED (Intelligent Electronic Device)
filter	string	Representa a regra ou o critério que acionou o evento, como " $I_A > 30$ "
threshold_achieved	bool	Indica se o evento foi ativado (True) ou desativado (False)