**Universidade Federal de Uberlândia**
**Faculdade de Engenharia Elétrica**

ALAILTON JOSÉ ALVES JÚNIOR

# Implementação de um sistema de Proteção, Automação e Controle virtualizado para subestações digitais baseado na Norma IEC-61850

Uberlândia-MG
2023

ALAILTON JOSÉ ALVES JÚNIOR

# Implementação de um sistema de Proteção, Automação e Controle virtualizado para subestações digitais baseado na Norma IEC-61850

Trabalho de conclusão de curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Engenheiro Eletricista grau bacharelado.

Área de concentração: Proteção e Automação de Sistemas Elétricos.

Orientador: Prof. Dr. Alan Petrônio

Uberlândia-MG

2023

## AGRADECIMENTOS

Primeiramente agradeço os meus pais, Alailton e Veronilda, por todo o apoio e incentivo durante toda a minha vida.

À minha irmã Lisa, agradeço pelo companheirismo, apoio emocional e por compartilhar os café e aspirações durante os meus anos de graduação.

Ao meu orientador Alan Petrônio pelos conselhos e por ter me convidado a participar do Laboratório de Redes Inteligentes (LRI), onde tive a oportunidade de me desenvolver profissionalmente.

Aos meus amigos da graduação e membros do LRI pelos momentos descontraídos e companheirismo.

Aos docentes da Faculdade de Engenharia Elétrica por todos os ensinamentos e oportunidades que me foram oferecidas.

À Conprove Indústrias pelos ensinamentos e orientações no desenvolvimento deste trabalho.

# Sumário

# Lista de Figuras

# Lista de Tabelas

## LISTA DE ABREVIAÇÕES E SÍMBOLOS

**ASN.1** – Abstract Syntax Notation One

**DFT** – Transformada Discreta de Fourier

**DPDK** – Data Plane Development Kit

**GOOSE** – *Generic Object Oriented Substation Event*

**KVM** – Kernel-based Virtual Machine

**MU** – *Merging Unit*

**OvS** – Open vSwitch

**PDIS** – *Protection Distance*

**PIOC** – *Protection Instantaneous Overcurrent*

**PRP** – *Parallel Redundancy Protocol*

**PTOC** – *Protection Timed Overcurrent*

**PTOV** – *Protection Over-Voltage*

**PTP** – *Precision Time Protocol*

**PTUV** – *Protection Under-Voltage*

**SDN** – *Software-Defined Networking*

**SEP** – Sistema Elétrico de Potência

**SV** – *Sampled Value*

**vIED** – IED Virtual

**VM** – Máquina Virtual

**vMU** – Virtual *Merging Unit*

**RESUMO**

Este trabalho aborda a evolução das tecnologias da informação na transformação de subestações de energia, considerando o aumento das fontes renováveis e a complexidade da rede elétrica moderna. Para atender a esses desafios, foi adotado o conceito de Proteção, Automação e Controle Centralizado (PAC Virtual), baseado em *smart grids* e na norma IEC 61850. O PAC Virtual implica na virtualização de dispositivos de proteção e controle da subestação, tornando-os independentes de hardware e obtendo, assim, maior flexibilidade e redução de custos e manutenções. As contribuições deste trabalho incluem a concepção e implementação do sistema PAC descrito acima, com foco na criação de IEDs virtuais que realizam funções de proteção e automação, bem como um simulador/controlador virtual para testes básicos dos protocolos da norma. Os resultados apresentaram tempos de resposta com baixas latências similares aos equipamentos físicos, indicando expectativas promissoras para este nova abordagem virtual.
**Palavras Chave:** IEC 61850, Smart Grid, PAC Virtual, PAC Centralizado, Virtualização de IEDs.

**ABSTRACT**

This document addresses the evolution of information technologies in the transformation of power substations, considering the increase in renewable sources and the complexity of the modern electrical grid. To meet these challenges, the concept of Centralized Protection, Automation, and Control (Virtual PAC) was adopted, based on smart grids and the IEC 61850 Standard. Virtual PAC involves the virtualization of protection and control devices in the substation, making them hardware-independent, thus achieving greater flexibility and cost reduction in maintenance. The contributions of this work include the design and implementation of the PAC system described above, with a focus on creating virtual IEDs that perform protection and automation functions, as well as a virtual simulator/controller for basic a testing the standard's protocols. The results showed response times with low latencies similar to physical equipment, indicating promising expectations for this new virtual approach.
**Keywords:** IEC 61850, Smart Grid, Virtual PAC, Centralized PAC, Virtual IEDs

# 1 INTRODUÇÃO

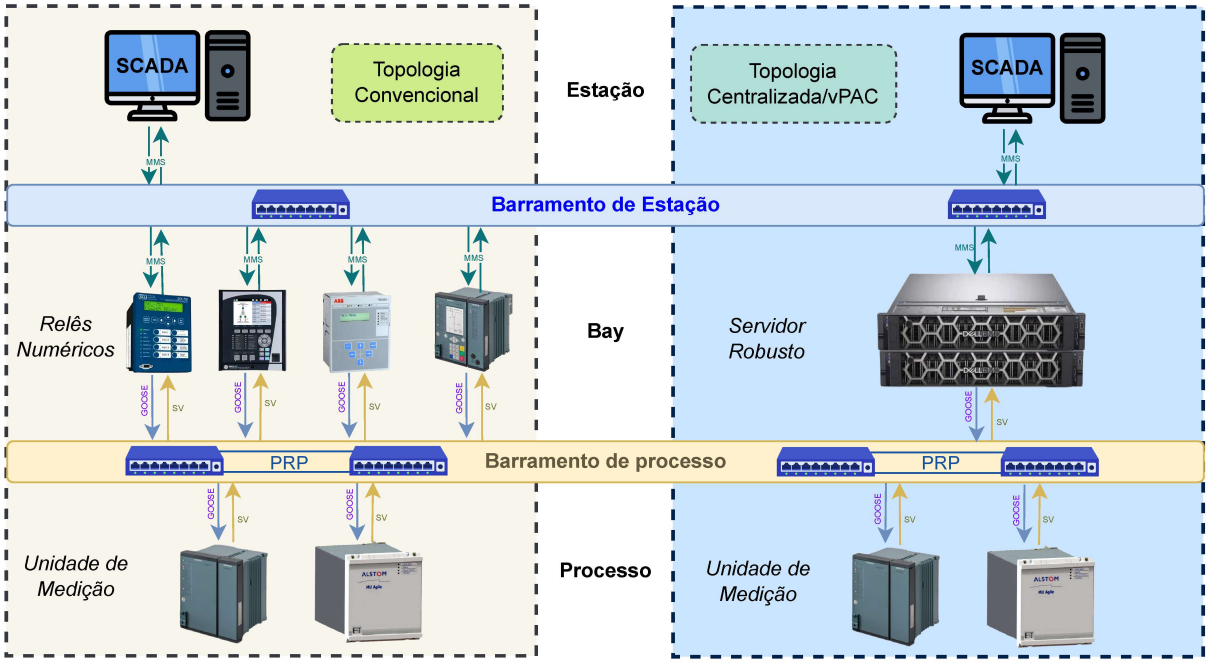
A evolução constante de novas tecnologias da informação tem desencadeado mudanças significativas na forma como concebemos, projetamos e gerenciamos elementos críticos dos sistemas de energia, em especial as subestações. Estas instalações desempenham um papel vital como pontos de acesso em sistemas elétricos, interligando linhas de transmissão, transformadores, geradores, fontes de energia renovável e diversas cargas. Contudo, a crescente integração de fontes renováveis, a expansão de novas linhas de transmissão e as demandas energéticas em constante crescimento têm tornado as subestações tradicionais limitadas para lidar com as complexidades da rede elétrica moderna.

Historicamente, as subestações foram projetadas para direcionar o fluxo unidirecional de energia, dos geradores para as cargas. No entanto, com a proliferação de sistemas de geração distribuída e microredes, o fluxo de potência está fluindo em ambas as direções, tornando o gerenciamento e proteção desses sistemas mais desafiadores. Em resposta a essas demandas, a busca por maior confiabilidade, flexibilidade e interoperabilidade tem impulsionado a adoção de tecnologias mais avançadas e a introdução do conceito de redes inteligentes nas subestações (HIGGINS et al., 2008).

Frente a esse desafio, o termo *smart grid* ganhou destaque, promovendo a necessidade de digitalização das subestações e a adoção de padrões abertos, como a norma IEC 61850. Este padrão, que estabelece protocolos de comunicação específicos para sistemas de energia, possibilita a transmissão de grandezas elétricas analógicas e sinais de controle de forma digital pela rede da subestação. Tal padronização otimiza a comunicação nesse ambiente, contribuindo para uma integração mais eficiente e uniforme (VYATKIN et al., 2010).

O tema em foco deste trabalho é uma nova topologia denominada de *Centralized Protection, Automation* e *Control* ou *Virtualized PAC*. Este modelo utiliza dos princípios de *smart grids* e a norma IEC 61850 para tornar as subestações de energia elétrica mais modernas e flexíveis. Nesta arquitetura, os equipamentos utilizados para proteção e controle da subestação, em especial os relês de proteção, não estariam mais conectados intrinsecamente ao hardware. Esses dispositivos estariam implementados em um ambiente virtualizado dentro de um servidor, conforme ilustra a Figura 1.1.

**Figura 1.1** – Comparação entre a arquitetura PAC convencional e centralizada

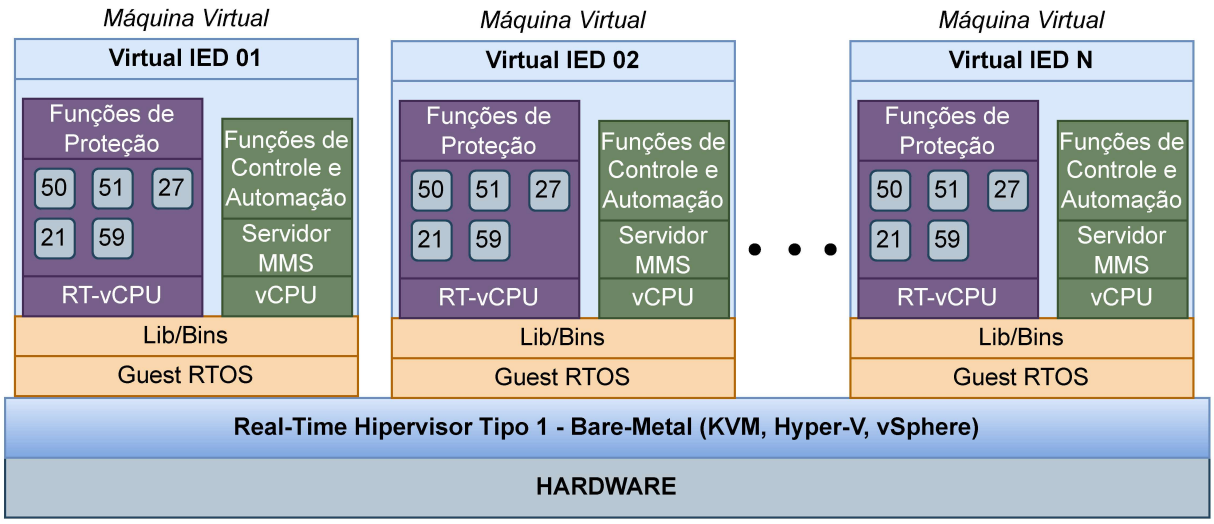



**Fonte:** autoria própria

No servidor, os IEDs virtuais encontraria-se dentro de máquinas virtuais ou containers operando apenas com os protocolos de comunicação da norma IEC 61850, reduzindo custos e facilitando o monitoramento do ciclo de vida do equipamento.

Além disso, a virtualização desses equipamentos traz diversas vantagens operacionais, como a maior flexibilidade para configurações dinâmicas e adaptativas, clonagens e redundâncias virtuais para o IED e simplicidade em atualizações e expansões da planta. Na Figura 1.2 é mostrado uma arquitetura do IED dentro de um servidor.

**Figura 1.2** – Modelo virtualizado de um relé de proteção




**Fonte:** autoria própria

Como descrito anteriormente, a expansão de novas fontes de energia distribuídas, faz com que o sistema PAC das subestações tenda a ser mais flexível e dinâmico, e isto pode se tornar uma tarefa mais trivial com a utilização de IEDs virtuais, os quais podem ser mais facilmente manipulados, escaláveis e independentes do hardware.

Porém, a transição para essa abordagem, na qual as funções de proteção e controle da subestação são virtualizadas, introduz desafios específicos na sua operação, especialmente no que diz respeito às funções de tempo crítico. Por isso, é fundamental conduzir pesquisas e avaliações para garantir que a virtualização desses sistemas atenda aos requisitos de segurança, confiabilidade e desempenho do sistema de potência.

Diante disso, este estudo presente pretende contribuir para a literatura executando estudos e análises desses sistemas de proteção virtualizados. Para isto, foi desenvolvida uma aplicação PAC centralizada em um ambiente laboratorial, tratando desde a configuração do servidor, quanto os IED e equipamentos de medição virtuais hospedados nele.

## 1.1 Objetivos

O objetivo deste trabalho é desenvolver e implementar uma prova de conceito de um sistema de Proteção, Controle e Automação (PAC) de uma subestação em um ambiente virtual, aderindo a alguns dos padrões da norma IEC 61850 explanados na seção metodológica. Neste sistema, busca-se verificar se a nova arquitetura sugerida para proteção elétrica consegue apresentar alguma viabilidade dentro do contexto do trabalho aqui definido. Em decorrência desse objetivo principal, seguem outros.

- Desenvolver o modelo básico de um IED Virtual (vIED), no qual utiliza os protocolos de comunicação especificados pela norma IEC 61850 para realizar medições das grandezas elétricas e enviar informações de estados (*trips*). Criado com o intuito de verificar o seu comportamento em um ambiente simulado e estimar a latência dele dentro de uma rede Ethernet.
- Elaborar um simulador/controlador virtual para os protocolos GOOSE e SV da norma IEC 61850, denominado de vMU. Este dispositivo tem o propósito de operar como uma ferramenta virtual de teste para os IEDs de proteção que aderem à norma, realizando condições de falta e medindo o tempo de atuação do IED.

## 1.2 Limitações e temas não abordados

Neste trabalho, não foram abordados os protocolos comunicação para redundância *Parallel Redundancy Protocol* (PRP) e sincronização *Precision Time Protocol* (PTP) citados na norma IEC 61850.

Além disso, toda a medição de tempo foi realizada de forma virtual dentro do

servidor considerando a precisão do *timer* do sistema operacional, sem a utilização de equipamentos externos para validar estes intervalos.

Ademais, a implementação do IED virtual focou unicamente no barramento de processo, no qual a latência das operações são mais críticas no ambiente virtual. Por esse motivo, não foi retratada a comunicação cliente/servidor MMS no dispositivo.

## 1.3 Estruturação do documento

No Capítulo 2 deste trabalho, são abordados o estado da arte e a fundamentação teórica. Já no Capítulo 3, é apresentado o desenvolvimento da proposta, desde os recursos tecnológicos utilizados até as aplicações desenvolvidas. No capítulo subsequente, são mostrados alguns dos resultados explorados e discussões para avaliar a arquitetura PAC supracitada. Por fim, no Capítulo 5, são apresentadas as conclusões deste estudo.

# 2 FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE

## 2.1 Estado da Arte

Atualmente, as funções de Proteção, Automação e Controle (PAC) estão integradas ao hardware, restringindo sua aplicação e expansão a esse componente específico. Com o intuito de prolongar a vida útil das aplicações e proporcionar maior flexibilidade, torna-se essencial separar as funções de aplicação e hardware. Nesse contexto, a recente brochura técnica do Grupo de Estudos B5 do CIGRE (CIGRE, Study Committee B5.60, 2022) aborda essa discrepância, apresentando dois conceitos técnicos promissores para Dispositivos Eletrônicos Inteligentes (IEDs) e sistemas centralizados de Proteção e Controle (CPC). A publicação explora requisitos conexos, limitações, arquiteturas de referência, bem como as oportunidades, desafios previstos e áreas adicionais de padronização e inovação.

O sistema PAC virtual em subestações, baseada na norma IEC 61850, não é uma novidade, já que foi primeiramente apresentado por Ferreira e Oliveira (2017), seguido por desenvolvimentos em 2018 (FERREIRA; OLIVEIRA, 2018). Esses trabalhos focaram na avaliação do desempenho do modelo, utilizando o Middleware Data Distribution Service DDS, para simular o tráfego da comunicação IEC 61850. Os resultados revelaram latências de até 5ms para 99% das amostras ensaiadas.

No mesmo ano, Wojtowicz, Kowalik e Rasolomampionona (2018) publicaram um trabalho destacando o desenvolvimento de um IED baseado em tecnologias de virtualização, concentrando-se na comunicação Cliente/Servidor da IEC 61850 e evidenciando a viabilidade da comunicação entre máquinas virtuais.

Outros estudos exploraram a arquitetura de containers para gerar tráfego de rede IEC 61850, como nos artigos de (ROSCH; NICOLAI; BRETSCHNEIDER, 2022) e (ROSCH et al., 2022). Esses autores desenvolveram aplicações simulando o tráfego na subestação, revelando um atraso médio de 21ms.

No ano de 2022, projetos iniciais começaram a surgir nas indústrias de proteção do setor elétrico, principalmente impulsionados pela aliança comercial vPAC Alliance (VPAC Alliance, 2023). Entre esses projetos, destaca-se o relé de proteção virtual apresentado pela Intel e Kalkitech, conforme detalhado por Samara-Rubio, McKenzie e Khajuria (SAMARA-RUBIO; MCKENZIE; KHAJURIA, 2022). Este último estudo demonstra os tempos de resposta do dispositivo virtual em diferentes condições de rede, além da implementação bem-sucedida de requisitos de redundância e sincronização, com resultados comparáveis aos relés físicos convencionais.

## 2.2 Norma IEC 61850

A diversidade de equipamentos presentes em uma subestação elétrica tornou necessário a adoção de um padrão de comunicação aberto para assegurar a interoperabilidade entre eles. Nesse contexto, a norma IEC 61850 surge como um padrão internacional voltado para a indústria de energia elétrica, com o propósito de estabelecer uma padronização na comunicação e controle de sistemas de automação e proteção em subestações elétricas. Elaborada pela Comissão Eletrotécnica Internacional (IEC), essa norma visa aprimorar a interoperabilidade e eficiência dos sistemas de automação em sistemas de energia elétrica, abrangendo não apenas as subestações, mas também as linhas de transmissão e usinas de energia (AFTAB et al., 2020).

A IEC 61850 é uma norma técnica que define uma estrutura de comunicação e um conjunto de protocolos de rede destinados a equipamentos e sistemas presentes em subestações. Seu principal objetivo é substituir os sistemas de comunicação tradicionais, que frequentemente se baseiam em protocolos proprietários e personalizados, por uma abordagem mais aberta e padronizada, possibilitando a interoperabilidade entre dispositivos e sistemas provenientes de diferentes fabricantes.

Essa norma estabelece o uso de protocolos de comunicação baseados em Ethernet, introduzindo novos protocolos como o MMS (*Manufacturing Message Specification*), o GOOSE (*Generic Object-Oriented Substation Event*) e o SV (*Sampled Values*). Esses protocolos são essenciais para a transmissão de informações críticas em tempo real e para o compartilhamento de dados.

### 2.2.1 Divisão da norma IEC 61850

A IEC 61850 é subdividida em 10 partes, cada uma delas dedicada a abordar aspectos específicos relacionados à automação de subestações, retratando os protocolos utilizados, requisitos de software e hardware e testes de validação. Abaixo é indicado o que é tratado em cada uma das partes, (IEC 61850-2, 2003).

- IEC 61850-1 - *Introduction and Overview*: Esta parte estabelece a introdução e a visão geral da norma;
- IEC 61850-2 - *Glossary*: O glossário define os termos e conceitos usados ao longo do documento;
- IEC 61850-3 - *General Requirements*: Esta parte estabelece os requisitos gerais que devem ser atendidos para a sua implementação no sistemas de automação de subestações elétricas;
- IEC 61850-4 - *System and Project Management*: Esta parte especifica o modelo de

informações de dispositivos e funções geralmente relacionados a usos comuns em aplicativos de automação de serviços públicos de energia;

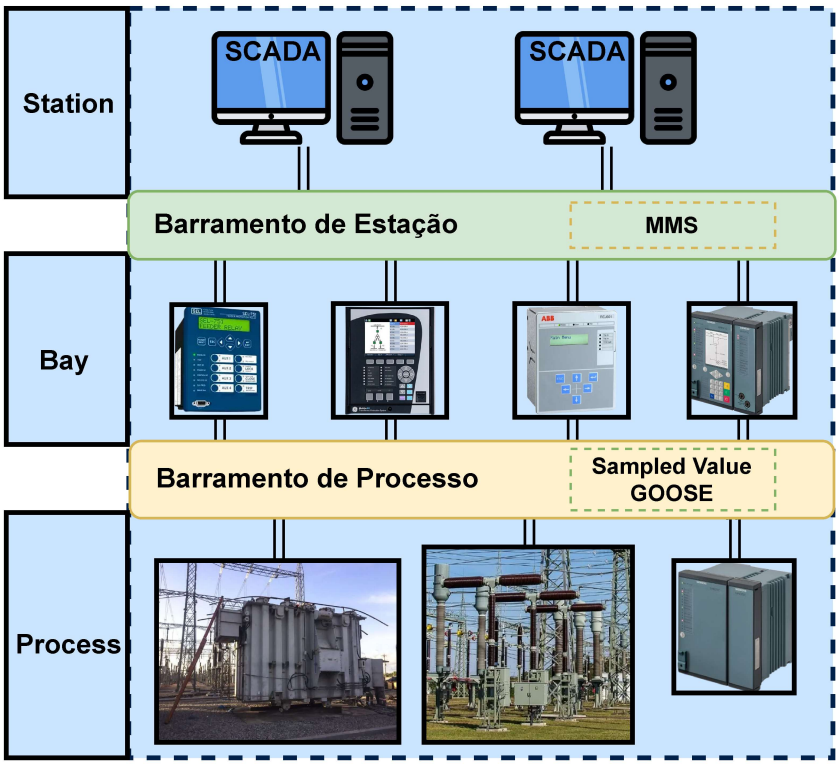
- IEC 61850-5 - *Communication Requirements for Functions and Device Models*: Aqui são definidos os requisitos de comunicação para funções e modelos de dispositivos em automação de subestações;
- IEC 61850-6 - *Configuration Description Language for Communication in Electrical Substations Related to IEDs*: Esta parte especifica uma linguagem de descrição para configurar Dispositivos Eletrônicos Inteligentes (IEDs) em Sistemas de Automação de Subestações (SAS) chamada SCL (System Configuration Description Language);
- IEC 61850-7-1 - *Basic Communication Structure for Substation and Feeder Equipment - Principles and Models*: Esta parte aborda os princípios e modelos da estrutura de comunicação para equipamentos de subestação e alimentadores;
- IEC 61850-7-2 - *Basic Communication Structure for Substation and Feeder Equipment - Abstract Communication Service Interface (ACSI)*: Esta parte lida com a interface de serviço de comunicação abstrata (ACSI) na estrutura de comunicação;
- IEC 61850-7-3: *Basic Communication Structure for Substation and Feeder Equipment – Common Data Classes*: Esta seção aborda as classes de dados comuns na estrutura de comunicação;
- IEC 61850-7-4 - B*asic Communication Structure for Substation and Feeder Equipment - Logical Node Classes and Data Classes*: Aqui são tratadas as classes de nós lógicos e classes de dados na estrutura de comunicação;
- IEC 61850-8-1 - *Specific Communication Service Mapping (SCSM) over ISO/IEC 8802-3*: Esta parte especifica o mapeamento de serviços de comunicação específicos (SCSM) sobre ISO/IEC 8802-3, que se refere ao protocolo Ethernet.
- IEC 61850-9-1 - *Specific Communication Service Mapping (SCSM) Sampled Value over Serial Unidirectional Multidrop Point to Point Link*: Aqui, são definidos os mapeamentos de serviços de comunicação específicos para a transmissão de valores amostrados em sistemas de ligação serial unidirecional multidrop ponto a ponto.
- IEC 61850-9-2 - *Specific Communication Service Mapping (SCSM) - Sampled Value over ISO/IEC 8802-3*: Esta parte aborda o mapeamento de serviços de comunicação específicos para a transmissão de valores amostrados sobre o protocolo Ethernet (ISO/IEC 8802-3).

- IEC 61850-10 - *Conformance Testing*: Esta seção especifica técnicas padronizadas para testar a conformidade de dispositivos clientes, servidores e de dispositivos de valor amostrado, bem como ferramentas de engenharia com a norma IEC 61850.

### 2.2.2 Definição dos Barramentos de Processo e Estação

Na IEC 61850, são definidos 3 níveis hierárquicos dentro da subestação. Os equipamentos no pátio da subestação, como transformadores, disjuntores e medidores, estão no nível *Process*. Já os equipamentos de proteção e controle, como IEDs e RTUs, que recebem informações das grandezas elétricas do nível *Process*, estão no nível *Bay*. Separando esses dois níveis encontra-se o Barramento de Processo, conforme demonstrado na Figura 2.1.

**Figura 2.1** – Ilustração dos barramentos de Processo e Estação na separação dos níveis *Process*, *Bay* e *Station*




**Fonte:** autoria própria

O último nível denominado *Station* está localizado acima do nível *Bay* e é separado pelo Barramento de Estação. No nível *Station* encontra-se a supervisão e parametrização dos equipamentos do nível *bay* (JANSSEN; APOSTOLOV, 2008).

Neste trabalho será desenvolvida uma aplicação virtual de um IED e uma ferramenta para ensaios de proteção, utilizando dois protocolos de comunicação da IEC 61850: (i) *Generic Object Oriented Substation Event* (GOOSE) e (ii) *Sampled Value* (SV), Por esse motivo, a topologia empregada enfoca exclusivamente o Barramento de Processo. Nos

tópicos subsequentes, será apresentada a notação ASN.1 utilizada por esses protocolos, bem como a estrutura dos pacotes GOOSE e SV.

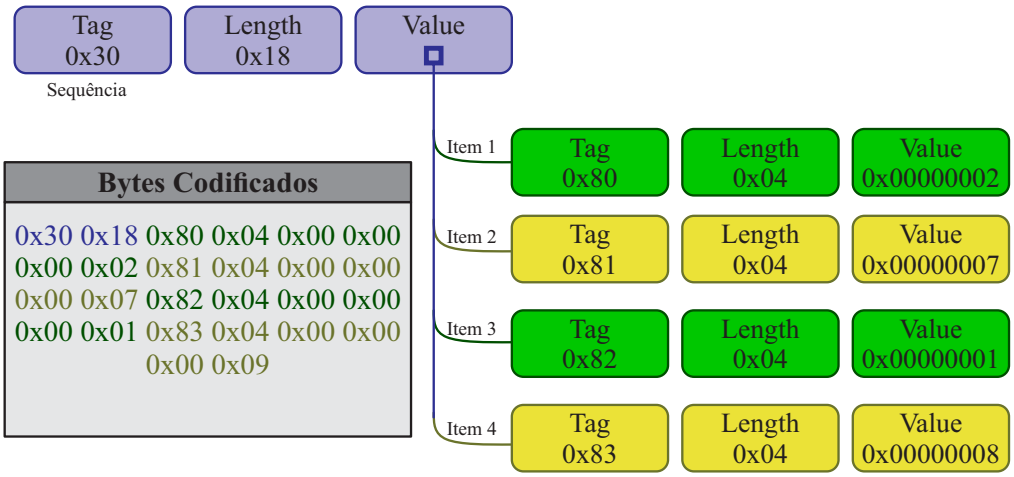
### 2.2.3 Notação ASN.1

O Abstract Syntax Notation One (ASN.1) é uma notação utilizada em protocolos de comunicação que define um modelo padronizado para codificar estruturação de dados dentro de pacotes de rede. Este modelo é utilizado pelos protocolos SV, GOOSE da IEC 61850, trazendo interoperabilidade e facilidade na codificação e decodificação deles.

A codificação de dados com base na ASN.1 utiliza uma arquitetura de 3 campos. O primeiro deles é a *Tag* que indica o tipo da estrutura de dados que deseja codificar, o segundo é o *Length* que descreve o comprimento em bytes da mensagem, e por último o *Value*, contendo toda a informação em octetos orientada por big-endian.

Na Figura 2.2, é demonstrado um exemplo de aplicação da notação para codificar uma sequência com 4 valores inteiros de 32 bits.

**Figura 2.2** – Exemplo de uma codificação de dados ASN.1




**Fonte:** autoria própria

### 2.2.4 Protocolo de comunicação GOOSE

O GOOSE é um protocolo de comunicação definido na parte 8 da IEC 61850 (IEC 61850-8-1, 2011). Ele é utilizado para transmitir de forma rápida informações de eventos e estados entre os dispositivos de proteção e controle na subestação. Ele é um dos principais protocolos definidos na norma e também o mais utilizado nas subestações.

Este protocolo opera diretamente na camada de enlace (*DataLink*) do modelo OSI, utilizando a arquitetura de *publish-subscribe* em *broadcast* ou *multicast* para transmitir a informação. Nesta arquitetura, quando um dispositivo envia o pacote GOOSE na rede ethernet, todos conectados a ela recebem essa informação e apenas aqueles IEDs

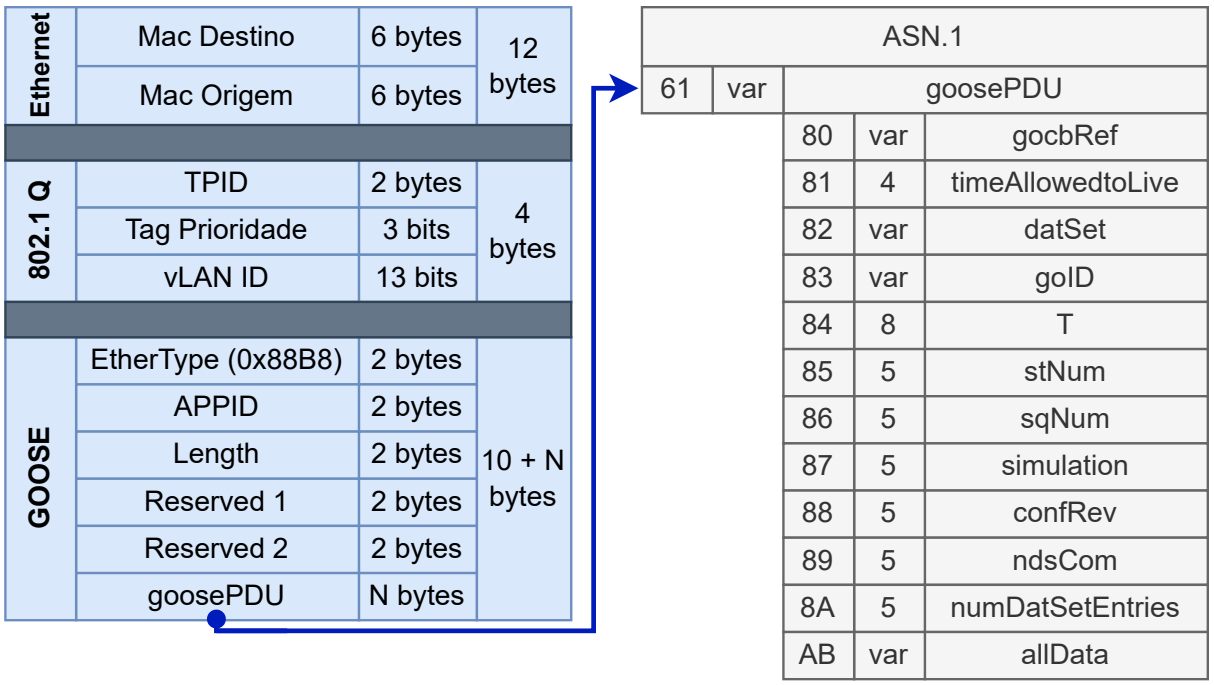
que desejam a informação contida no pacote o decodificam, filtrando-a com base no seu endereço MAC ou por outras informações contidas na mensagem.

Neste modelo, não há verificação se a mensagem chegou e por esse motivo é usualmente enviado o mesmo pacote múltiplas vezes.

O GOOSE faz parte da segunda camada da rede no modelo OSI, e por isso ele é encapsulado dentro do protocolo Ethernet, no qual contém informações sobre o MAC de destino e origem. Após a camada Ethernet, a norma permite adicionar o protocolo 802.1Q antes do GOOSE. O 802.1Q possibilita a criação de virtual LANs que segmentam virtualmente os pacotes, e também define uma *tag* de prioridades dentro deles. O Switch, equipamento no qual os pacotes são transmitidos, redireciona os pacotes para portas específicas com base na virtual LAN e também prioriza o processamento destes com base na prioridade definida na *tag*.

Na Figura 2.3 abaixo é mostrado como o pacote GOOSE é estruturado dentro de um *frame* ethernet.

**Figura 2.3** – Estrutura do pacote GOOSE




**Fonte:** (IEC 61850-8-1, 2011)

A descrição e o tipo de dado de cada campo do pacote GOOSE é apresentado na Tabela 2.1. O indicador [M] e [O], após o nome, indica se o parâmetro é opcional ou mandatório no *frame*.

**Tabela 2.1** – Atributos presente no cabeçalho do pacote GOOSE

| Nome do Atributo | Descrição | Tipo de dado |
|---|---|---|
| goCBRef [M] | Nome do Control Block associado a mensagem GOOSE | Visible-string |
| timeAllowedToLive [M] | Tempo máximo de espera até a próxima mensagem GOOSe | INT32U |
| datSet [M] | Nome do data Set no qual a mensagem GOOSE pertence | Visible-string |
| golD [O] | Identificado da mensagem GOOSE | Visible-string |
| T [M] | Estampa de tempo da última alteração nos dados do pacote | UtcTime |
| stNum [M] | Número que indica a quantidade de vezes que os dados foram alterados | INT32U |
| sqNum [M] | Número que indica a quantidade de pacotes enviados sem a alteração | INT32U |
| simulation [M] | Indica se a mensagem enviada tem fins de simulação ou não | Boolean |
| confRev [M] | Número da revisão do arquivo que no qual o protocolo está seguindo | INT32U |
| ndsCom [M] | Indica se o Control Block precisa ser revisado | Boolean |
| numDatSetEntries [M] | Número de elementos dentro do dataSet | INT32U |
| allData [M] | Dados do datset enviados pelo pacote | SEQUENCE of Data |

**Fonte:** (IEC 61850-8-1, 2011)

As informações contidas na mensagem GOOSE vem de um conjunto de dados (*datSet*) definida pela norma, esses dados são inseridos no pacote no último bloco de informação chamado *allData* mostrado na Figura 2.3. Eles são colocados em sequência seguindo a codificação ASN.1. Os tipos de dados que podem ser utilizados dentro desse bloco são mostrados na Tabela 2.2 com as suas respectivas *Tags*.

**Tabela 2.2** – Possíveis tipos de dados transmitidos pelo pacote GOOSE

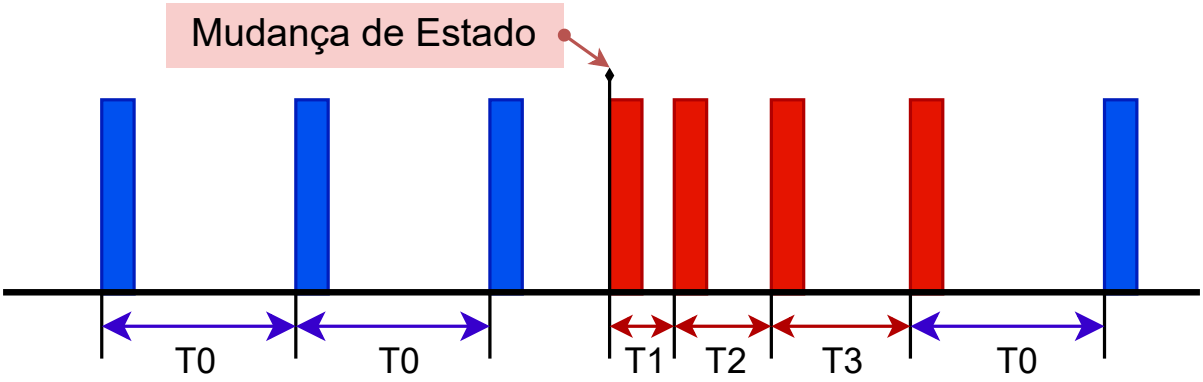
| Tipo de dado | allData ASN.I Tag | Tipo de dado | Tipo de dado |
|---|---|---|---|
| Boolean | 0x83 | CODED ENUM | 0x84 |
| INT | 0x85 | OCTET STRING | 0x89 |
| INTU | 0x86 | VISIBLE STRING | 0x8A |
| FLOAT32 | 0x87 | TimeStamp | 0x91 |
| ENUMERATED | 0x85 | Quality | 0x84 |

**Fonte:** (IEC 61850-8-1, 2011)

#### Retransmissão

As mensagens deste protocolo são enviadas periodicamente com base no tempo de vida (*timeAllowedToLive*) definido no cabeçalho do pacote. Caso ocorra um evento e os dados contidos no *allData* sejam modificados, a mensagem é enviada instantaneamente e os outros pacotes subsequentes são enviados com base em uma progressão geométrica, até atingirem o tempo do *timeAllowedToLive*/2. Este comportamento é ilustrado na Figura 2.4.

**Figura 2.4** – Retransmissão da mensagem GOOSE na ocorrência de um evento




**Fonte:** autoria própria

Na Figura 2.4, o Tempo T0 é o tempo normal entre cada transmissão do pacote, sendo este igual a *timeAllowedToLive*/2. O Tempo T1 é o menor tempo que a mensagem leva para ser reenviada após a ocorrência de um evento. Os outros intervalos T2 e T3, seguem a progressão geométrica de T1 até o tempo T0, no qual, T2 = T1*2, T3 = T2*2, assim por diante até atingirem o tempo T0.

### 2.2.5 Protocolo de comunicação *Sampled value*

O *Sampled value* é um protocolo de comunicação definido na parte 9 da IEC 61850 (IEC 61850-9-2, 2011). Ele é utilizado para transmitir dados amostrados de corrente e tensão entre os dispositivos de proteção e os equipamentos de medição.

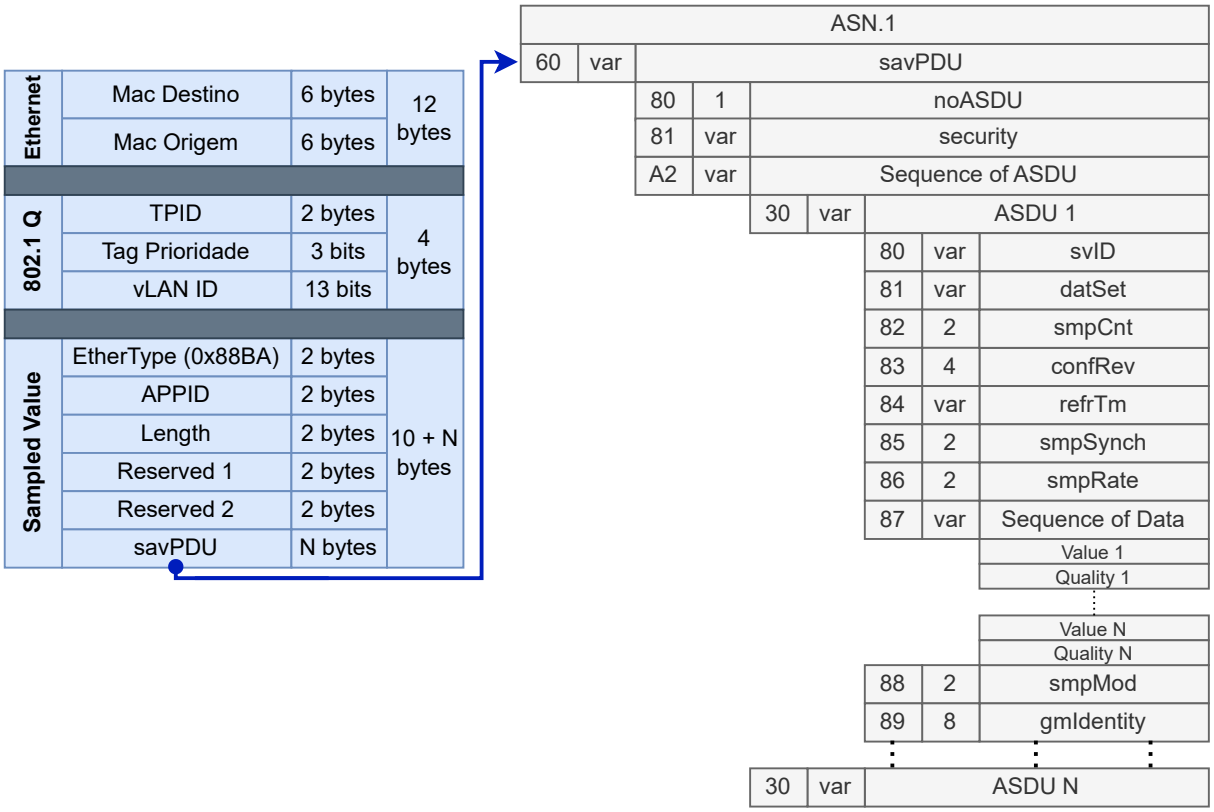
Este protocolo é similar ao GOOSE. Ambos são enviados na camada de enlace do modelo OSI e utilizam a arquitetura de publish-subscribe. Porém o SV é utilizado apenas para transmitir dados amostrados de corrente e tensão, enquanto o GOOSE é utilizado para transmitir informações de eventos e estados.

Este protocolo possui uma taxa de amostragem fixa de 80 amostras por ciclo para fins de proteção e 256 amostras por ciclo para fins de medição. Considerando a frequência da rede em 60 Hz, a quantidade de pacotes transmitidos por segundo é de 4800 e 15360 respectivamente. Esta quantia é consideravelmente elevada e pode ocupar grande parte da banda de transmissão da rede ethernet.

Para viabilizar a transmissão destes *frames* de forma mais eficiente, na normativa IEC 61869-9 (IEC 61869-9, 2016), foi estabelecido a transmissão de mais de uma amostra dentro de um mesmo *frame Sampled value*. Deste modo, para fins de proteção, a mensagem SV enviada na rede carregaria a medição de 2 amostras e no caso de uma mensagem SV para fins de medição, carregaria 8 amostras em um mesmo *frame*. Logo, a quantidade de pacotes transmitidos na rede por segundo é reduzida para 2400 e 1920, porém mantendo a mesma taxa de amostragem.

A estrutura do frame SV diverge do GOOSE apenas no Ethertype, e no PDU associado a ele. Cada uma das amostras são armazenadas dentro de uma sequência de ASDU, conforme indicado na Figura 2.5.

**Figura 2.5** – Estrutura do pacote GOOSE




**Fonte:** (IEC 61850-9-2, 2011)

A descrição de cada parâmetro do bloco ASDU é indicado na Tabela 2.3, o indicador [M] e [O] após o nome indica se o parâmetro é opcional ou mandatório no *frame*.

**Tabela 2.3** – Informações presentes no bloco ASDU do pacote *Sampled value*

| Nome do Atributo | Descrição | Tipo de dado |
|---|---|---|
| sviD [M] | Identificador da mensagem SV | Visible-string |

**Tabela 2.3** - Informações presentes no bloco ASDU do pacote *Sampled value*

| Nome do Atributo | Descrição | Tipo de dado |
|---|---|---|
| datSet [O] | Nome do data Set no qual a mensagem SV pertence | Visible-string |
| smpCnt [M] | Contador de amostrados | INT16U |
| confRev [M] | Número da revisão do arquivo que no qual o protocolo está seguindo | INT32U |
| refrTm [O] | Tempo de quando o buffer de envio foi atualizado | INT128U |
| smpSynch [M] | Indica se a amostra está sincronizada | INT8U |
| smpRate [O] | Quantidade de amostras por período | INT16U |
| Sequence of Data [M] | Valores de corrente e tensão amostrados | INT32 |
| smpMod [O] | Formato da taxa de amostra, por ciclo ou por segundo | INT16U |
| gmIdentity [O] | Identidade do clock do Grandmaster, utilizado para sincronismo | Octet String |

**Fonte:** [IEC 61850-9-2]

## 2.3 Proteção de Sistemas Elétricos de Potência

Nesta seção, são apresentados os princípios de um sistema de proteção para Sistemas Elétricos de Potência Sistema Elétrico de Potência (SEP), detalhando alguns conceitos e equipamentos utilizados.

O sistema de proteção desempenha um papel fundamental em qualquer projeto elétrico, assegurando a segurança e confiabilidade da planta. Suas funções incluem detectar e isolar falhas ou distúrbios no sistema, visando evitar danos a equipamentos, interrupções no fornecimento de energia e garantir a segurança das pessoas.

A princípio, toda proteção de um sistema elétrico deve seguir as seguintes propriedades (FILHO; MAMEDE, 2020):

1. Confiabilidade: o nível de certeza em que um sistema de proteção atue corretamente em qualquer situação;
2. Seletividade: capacidade da proteção de localização o trecho de uma falha e isolar somente a parte afetada.
3. Velocidade: propriedade do sistema de detectar falhas no menor tempo possível
4. Sensibilidade: a proteção deve conseguir detectar faltas com a menor margem de tolerância entre a operação e não operação.

No contexto do sistema elétrico, diversos equipamentos são destinados à proteção deste. Entre esses, destacam-se os relés de proteção. Ao longo dos anos, esses dispositivos evoluíram do modelo mecânico para o microprocessado, atualmente conhecido como *Intelligent Electronic Device* (IED). Esses relés monitoram diversos parâmetros do sistema elétrico, como tensão e corrente, e são responsáveis por detectar condições anormais e emitir comandos para os dispositivos seccionadores, interrompendo o circuito, entre outras intervenções.

Os disjuntores são dispositivos de comutação que, diferentemente das chaves seccionadoras, interrompem o fornecimento de energia elétrica em condições de falhas. Eles são controlados pelos relés de proteção.

Ainda no contexto de proteção elétrica, os Transformadores de Corrente (TC) e os Transformadores de Potencial (TP) são transformadores que reduzem as correntes e tensões do sistema para níveis que podem ser monitorados pelos relés de proteção ou por sistemas de faturamento.

Com a migração das grandezas analógicas para as digitais, surgiram os equipamentos denominados *Merging Unit* (MU). Esses dispositivos são responsáveis por converter os valores de corrente e tensão do secundário dos TCs e TPs convencionais para valores digitais, transmitindo-os por meio do protocolo *Sampled value* da norma IEC 61850.

## 2.4 Virtualização

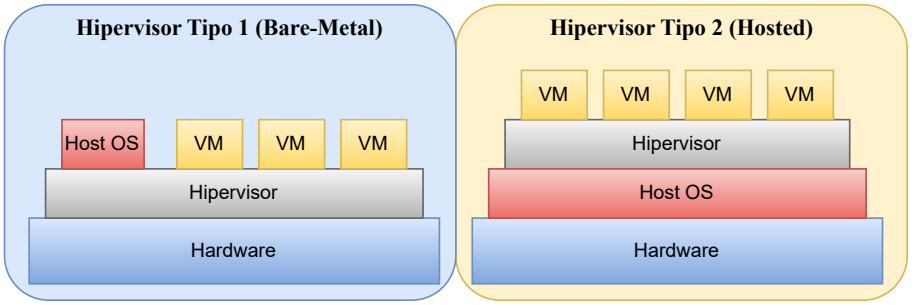
A Virtualização é uma tecnologia que permite a utilização mais eficiente dos recursos de um hardware. Ela possibilita a utilização de toda a capacidade de um único hardware, distribuindo seus recursos entre diversos ambientes virtuais isolados.

Utilizando desta tecnologia de virtualização é possível desenvolver aplicações em uma Máquina Virtual (VM) ou containers que independem do recursos físicos do hardware, proporcionando benefícios como otimização de recursos e facilidade de migração e escalabilidade do projeto. Neste sistema, a VM é um ambiente virtual que simula um computador físico com seu próprio sistema operacional, recursos e aplicativos isolados.

A VM é geralmente hospedada em servidores físico no qual são chamados de *Host*. Um *Host* pode ter várias VM operando ao mesmo tempo, no qual utilizam, distribuidamente e segundo suas necessidades individuais, os seus recursos de *hardware* como: núcleos, memória, cache, etc.

O software que permite a virtualização do dispositivo é chamado de hipervisor, ele é responsável por criar e gerenciar máquinas virtuais. Existem dois tipos de hipervisores, o tipo 1, que é executado diretamente no hardware proporcionando melhor eficiência, e o tipo 2, que é executado em um sistema operacional hospedeiro. Ambas arquiteturas são mostradas na Figura 2.6.

**Figura 2.6** – Ilustração das duas arquiteturas utilizadas em sistemas virtualizados




**Fonte:** autoria própria

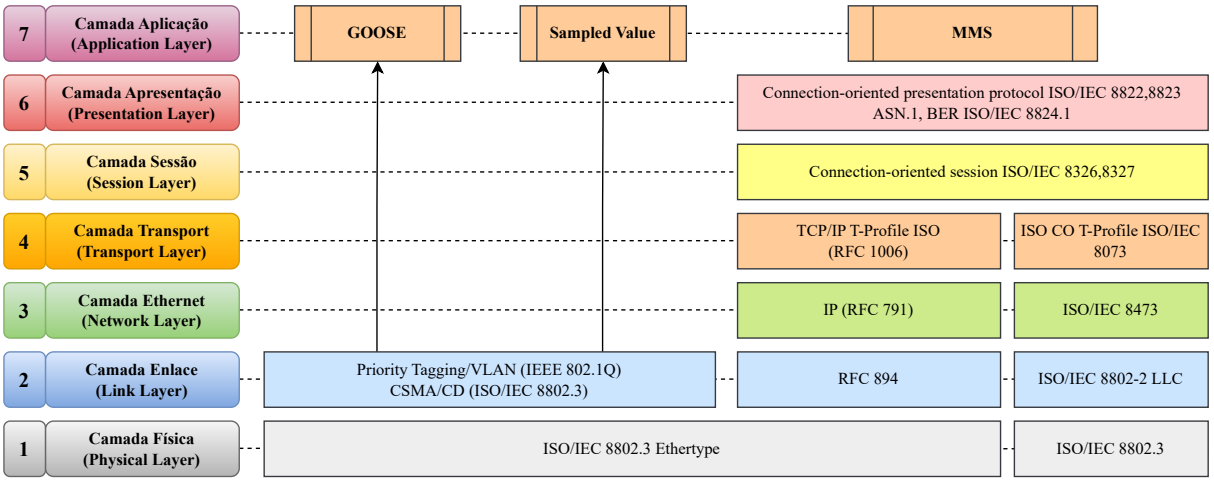
## 2.5 Redes Ethernet

Embora o propósito desta seção não seja discorrer sobre o funcionamento das redes[1], é crucial adquirir uma compreensão básica para analisar o comportamento dos pacotes especificado na norma IEC 61850.

Um aspecto relevante é a estruturação da rede de computadores em camadas. Apesar de ser uma abstração funcional, essa divisão facilita a compreensão e organização da informação. Cada vez que um pacote transita para uma camada superior, um cabeçalho é adicionado a ele. E este cabeçalho desempenha uma função específica relacionada aos serviços necessários para a transmissão dos dados.

A estrutura de camadas frequentemente adotada em redes é o Modelo OSI (ISO/IEC 7498-1:1994, 1994). Este modelo estabelece sete camadas distintas para a organização e gestão eficiente dos protocolos de comunicação. Estas camadas são: (i) Camada Física; (ii) Camada de Enlace de Dados; (iii) Camada de Rede; (iv) Camada de Transporte; (v) Camada de Sessão; (vi) Camada de Apresentação; e (vii) Camada de Aplicação. Na Figura 2.7 é mostrado a estruturação dos principais protocolos da IEC 61850 no modelo OSI.

**Figura 2.7** – Estruturação dos principais protocolos da norma IEC 61850 nas camadas da rede




**Fonte:** autoria própria

[1]Para mais informações, consultar o livro "Redes de Computadores e a Internet: Uma Abordagem Top-Down" do autor Jim Kurose

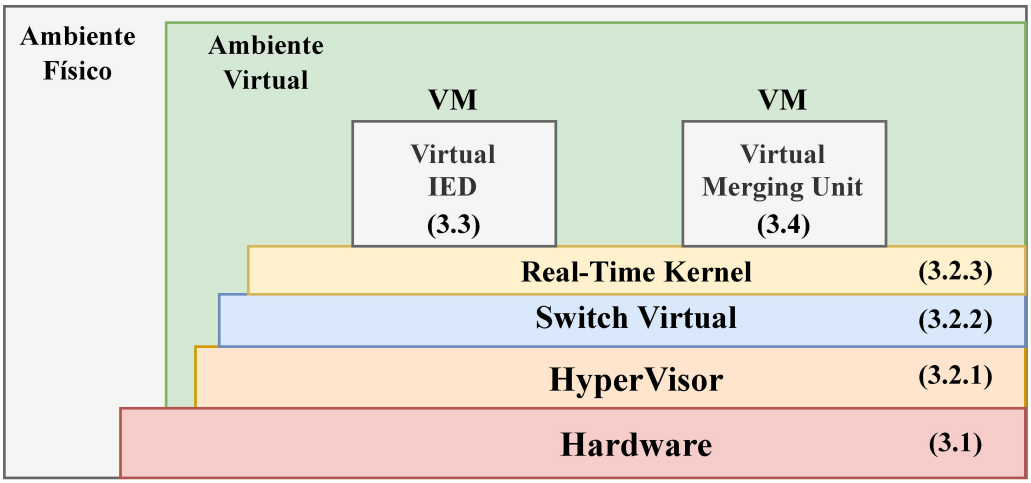
Outro ponto a considerar é o comportamento do GOOSE e do *Sampled value*, nos quais a transmissão ocorre por meio de broadcast ou multicast em alta velocidade na camada de enlace da rede. Tais protocolos apresentam suscetibilidade para congestionar as filas de dados nas portas do switch. Nesse contexto, é vantajoso segmentá-los conforme o protocolo IEEE 802.1Q (IEEE 802.1Q, 2018), proporcionando uma gestão mais eficiente do tráfego de dados e evitando sobrecargas.

O IEEE 802.1Q insere identificadores de *Virtual Local Area Network* (vLAN) dentro dos pacotes, possibilitando a criação de redes virtuais. Cada vLAN é associada a um identificador numérico, permitindo que dispositivos de rede, como o Switch, isolem e segmentem o tráfego da respectiva vLAN. Dessa forma, este protocolo possibilita separar os envios broadcast e multicast do GOOSE e *Sampled value* em grupos com diferentes vLANs, promovendo uma arquitetura mais eficiente e evitando a sobrecarga da banda de transmissão da rede.

# 3 METODOLOGIA

Neste capítulo é mostrado o desenvolvimento dos processos utilizados para a construção dos equipamentos virtuais (IED e MU) e a inserção deles no servidor. Na Figura 3.1 são apresentadas cada uma das etapas do projeto e seus respectivos capítulos.

**Figura 3.1** – Planta do projeto desenvolvido




## 3.1 Especificações e configurações do Servidor

Na realização dos testes e validação do trabalho, foi utilizado o servidor PowerEdge R730 Rack Server da empresa Dell Technologies, disponibilizado pelo Laboratório de Redes Inteligentes (LRI), mostrado pela Figura 3.2.

**Figura 3.2** – Imagem do servidor utilizado

**Fonte:** autoria própria

As especificações deste hardware são apresentadas na Tabela 3.1.

**Tabela 3.1** – Especificações do hardware do servidor utilizado

| Tipo | Quantidade | Modelo |
|---|---|---|
| Nº CPU | 40 | Intel Xeon E5-2650 v3 bits: 64, 2300 MHz |
| Network | 4 Portas 1Gib/s | Intel I350 Gigabit Network, Dell 4P I350-t |
| Memória HD | 9.82 TiB | PERC H730 Mini size |
| Memória RAM | 128 Gib | 8 x M393A2G40EB1-CRC DD4 2400 MHz |

As máquinas virtuais foram criadas com as seguintes configurações.

**Tabela 3.2** – Configuração das máquinas virtuais

| Tipo | Quantidade |
|---|---|
| Sistema Operacional | Linux Ubuntu 22.04.2 LTS |
| Nº CPU | 2 |
| Memória HD | 40 GiB |
| Memória RAM | 4096 MiB |
| Network | Brigde Network (Open vSwitch) |

Para o controle do servidor e de suas máquina virtuais, foi desenvolvido uma aplicação Desktop para o sistema operacional Windows em C#. Este software permite criar novas máquinas ou deletá-las, além de controles básicos como ligar, desligar e reiniciar. A tela do software é mostrada na Figura 3.3.

**Figura 3.3** – Tela do software utilizada para controle das Máquinas Virtuais no Servidor


**Fonte:** autoria própria

## 3.2 Recursos tecnológicos empregados de suporte

As operações de proteção e controle dentro de uma subestação precisam atender um certo grau de velocidade e confiabilidade. Por esse motivo, um dos principais problemas da utilização de equipamentos virtualizados é garantir que estes consigam atender os requisitos de tempo e precisão no sistema em que estão inseridos.

Para alcançar os requisitos supracitados, foi utilizado um Kernel do Linux em tempo real e uma arquitetura de rede virtual, ambas tecnologias estão descritas nos capítulos a seguir.

### 3.2.1 Kernel-based Virtual Machine

A ferramente de virtualização (hipervisor) das máquina virtuais utilizado neste trabalho foi o Kernel-based Virtual Machine (KVM), uma infraestrutura de virtualização de código aberto que se integra diretamente ao kernel do Linux, transformando-o em um hipervisor do tipo I (*bare-metal*). Essa tecnologia pode ser encontrada em: linux-kvm.org.

O KVM aproveita as extensões de virtualização de hardware oferecidas por processadores modernos, como Intel VT-x e AMD-V. Nessas arquiteturas, as máquinas virtuais são executadas diretamente no hardware físico, resultando em um desempenho que se assemelha ao de sistemas físicos.

A vantagem deste hipervisor é sua capacidade de alocar recursos de hardware de forma isolada e exclusiva para cada máquina virtual. Isto garante que o desempenho de uma VM não seja afetado pelas outras, proporcionando um ambiente de virtualização estável.

### 3.2.2 Open vSwitch e DPDK

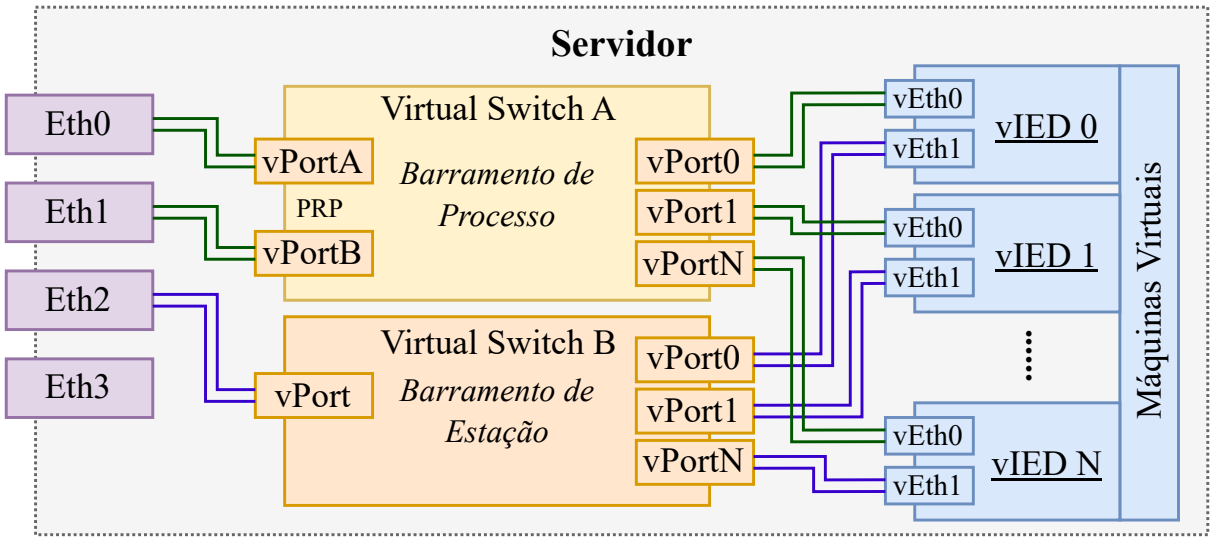
O Open vSwitch (OvS) é um software de comutação de rede virtual utilizado em servidores para distribuir e interconectar o fluxo de dados da rede Ethernet entre as máquinas virtuais e containers. Ele introduz ao ambiente virtualizado uma arquitetura similar a um Switch real, permitindo a utilização de tecnlogias como *Software-Defined Networking* (SDN) para simplificar o gerenciamento da rede, vLAN Tags para separar virtualmente os pacotes, tunelamento e balanceamento do fluxo de dados.

O software permite a criação de *Bridges* que operam como um Switch virtual, possuindo portas virtuais que interconectam o adaptador de rede físico com as máquinas virtuais.

Neste trabalho, as portas de rede físicas 0 e 1 do servidor foram conectadas ao Virtual Switch A do servidor, representando o barramento de processo. Por mais que não tenha sido implementado no projeto, foram utilizadas duas portas de rede por causa

o protocolo de redundância PRP. Ademais, a terceira porta do servidor foi utilizada para comunicação com a máquina virtual, conectada no Virtual Switch B simbolizando o barramento de estação. O arranjo descrito acima é mostrado pela Figura 3.4

**Figura 3.4** – Demonstração da funcionalidade de preempção em um RT-Kernel




A implementação da camada intermediária, OvS, entre o hardware físico e as máquinas virtuais aumenta a latência tanto no envio quanto no recebimento dos pacotes. Para contornar este problema, é utilizado em conjunto ao OvS o sistema de Data Plane Development Kit (DPDK) (DPDK, 2023).

O DPDK é um conjunto bibliotecas que permite o processamento e transferência em baixo nível dos pacotes que chegam da rede diretamente para espaço do usuário, reduzindo o tempo de latência que estes gastariam se tivesse que passar pelo o sistema de gerenciamento do Kernel antes do usuário final.

Logo, a integração do DPDK com o software OvS proporciona um melhor desempenho e eficiência no encaminhamento dos pacotes que são enviados e recebidos pelo servidor para o ambiente virtualizado. Ambas as tecnologias podem ser encontradas em: (Open vSwitch, 2023) e (DPDK, 2023).

### 3.2.3 Real-Time Kernel

O Kernel pode ser considerado o núcleo de um sistema operacional, sendo responsável por fornecer uma camada de abstração entre o hardware do dispositivo e os seus processos. ele realiza tarefas críticas, como escalonamento de processos, gerenciamento de memória, controle de acesso aos recursos de hardware e comunicação entre o software e os *drivers*.

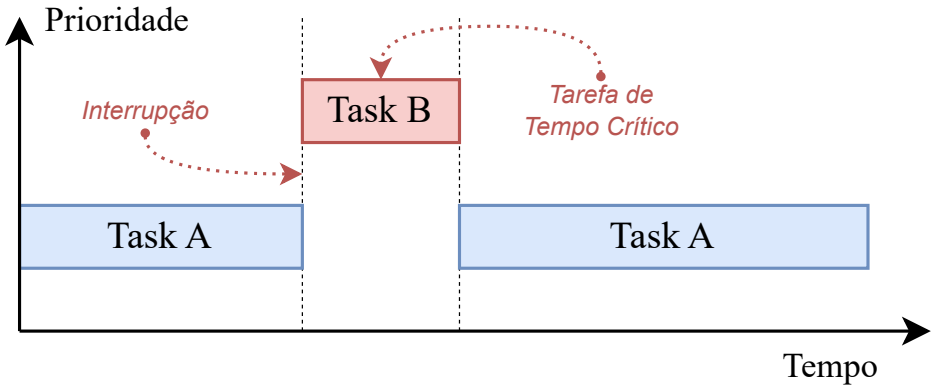
O Real-Time Kernel, também conhecido como *Fully Preemptible* Kernel, é uma extensão (*patch*) do Kernel normal que tem como objetivo proporcionar capacidades

de tempo real ao sistema operacional. Ele é disponibilizado pela Linux Foundation e encontrado em linuxfoundation.org.

Diferentemente do Kernel padrão do Linux, o Real-Time Kernel é projetado para atender a requisitos de tempo real, o que significa que ele é capaz de responder a eventos e tarefas com latências extremamente baixas e previsíveis. Isso é possível pois este sistema permite preempção determinística e garantida entre os processos executados, de forma otimizada para obter a menor latência possível na execução de tarefas de alta prioridade.

A preemptividade é capacidade de interromper um atividade em execução para executar outra com prioridade mais elevada, conforme indicado pela Figura 3.5.

**Figura 3.5** – Demonstração da preempção em um RT-Kernel




Um exemplo de aplicação desta tecnologia, no contexto deste trabalho, é quando o IED virtual está executando uma tarefa de comunicação com o sistema supervisório e ocorre a atuação do sistema de proteção. Neste cenário, a tarefa de baixa prioridade que estava sendo executada anteriormente é interrompida imediatamente e o processo de envio da mensagem de TRIP ocorre com a menor latência possível.

Outrossim, o kernel utilizado neste projeto foi compilado na versão 5.15.96 com o *patch* em tempo real na versão 5.15.96-rt61, ambos encontrados em: kernel.org. Na compilação, foi utilizada a própria configuração padrão do kernel do Ubuntu 22.04 como base e alterou-se apenas os parâmetros indicados na Tabela 3.3.

**Tabela 3.3** – Parâmetros alterados na compilação do kernel em tempo real

| Nome do Parâmetro Alterado | Valor |
|---|---|
| Preemption Model | Fully Preemptible Kernel (Real-Time) |
| Timers subsystem | High Resolution Timer Support |
| Timer tick handling | Full dynticks system (tickless) |
| Timer frequency | 1000 HZ |
| Default CPUFreq governor | performance |

## 3.3 Virtual IED

O IED Virtual desenvolvido neste trabalho teve como princípio testar e validar as funções de proteção de uma subestação com os protocolos de comunicação da norma IEC 61850 em um ambiente virtualizado. Para isso, o vIED em operação no servidor foi desenvolvido nas linguagens C e Python com foco em performance e utilizando as bibliotecas do Kernel em tempo real, para o sistema operacional Linux.

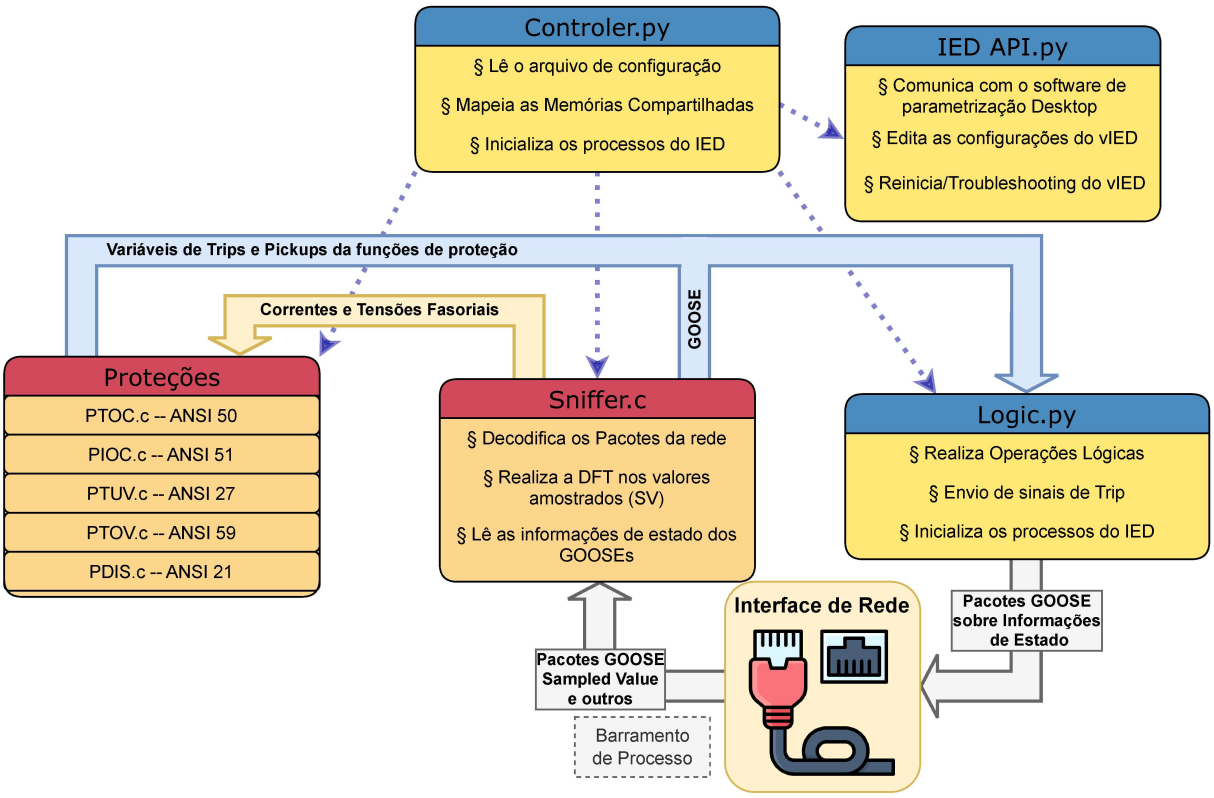
Além disso, também foi criada uma aplicação Windows em C# para a interface gráfica com o usuário, na qual permite a parametrização do IED de forma fácil e prática, a sua tela inicial é mostrada na Figura 3.6.

**Figura 3.6** – Tela inicial do software de parametrização do vIED


**Fonte:** autoria própria

As funções de baixa prioridade, como a comunicação com o aplicativo Desktop e as configurações gerais da máquina, foram implementadas em Python. Por outro lado, as aplicações de tempo crítico, como as funções de proteção e sinalizações de *trips*, foram desenvolvidas em C, utilizando as bibliotecas do Linux e funcionalidades do kernel em tempo real, conforme ilustrado na Figura 3.7.

**Figura 3.7** – Algoritmos implementados no software do IED virtual




**Fonte:** autoria própria

A descrição funcional de cada unidade é resumidamente explicada na sequência.

**Controler.py:** Código principal do vIED, ele é responsável por ler os arquivos de configuração e inicializar os serviços de proteção, Sniffer e a API do IED, além de mapear as memórias compartilhadas na RAM para que os outros serviços possam usá-la;

**IED API.py:** Algoritmo que realiza a comunicação entre a máquina virtual e o software *desktop* de configuração. Ele recebe os arquivos de configuração do vIED e envia os estados das variáveis e informações de corrente e tensão;

**Sniffer.c:** Código utilizado para capturar os pacotes SV da rede Ethernet e realizar o cálculo dos valores fasoriais das correntes e tensões, disponibilizando-os para os demais processos.

**Proteções:** Neste conjunto está presente o algoritmo de cada função de proteção que é realizado com base no valores disponibilizados pelo Sniffer, esses algoritmos serão explicados nas seções seguintes.

**Logic.py:** Neste código são realizadas as operações lógicas das variáveis do vIED e também envia os pacotes do protocolo GOOSE na ocorrência de uma mudança de estados ou quando o *timeout* é atingido.

A implementação das funções do vIED foi organizada de forma a executar cada

algoritmo em um serviço independente e em paralelo. Cada serviço possui uma prioridade correspondente à importância da função que desempenha. Para viabilizar a comunicação entre esses serviços, foi empregado o sistema de *Shared Memory* (SHM) do Linux, o qual cria arquivos virtuais com nomes específicos para compartilhar dados entre os processos.

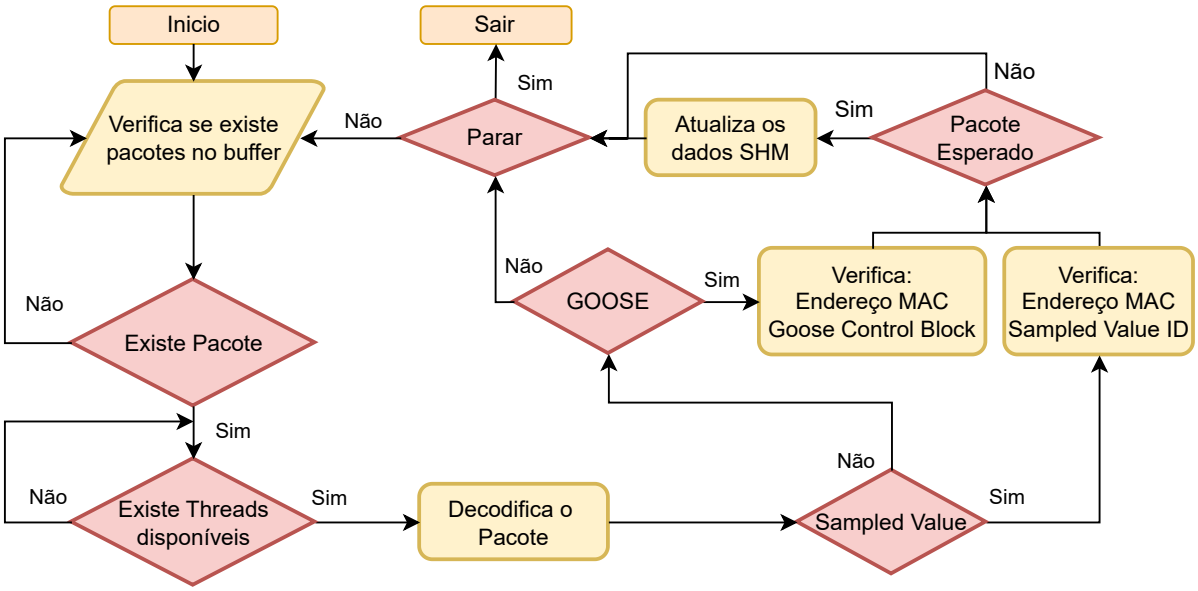
Nos tópicos a seguir serão apresentados cada um dos algoritmos implementados no vIED. Ressalta-se que os diagramas expostos são apenas exemplos da lógica utilizada nas funções. O código real implementado no vIED leva em consideração outros parâmetros não apresentados para garantir a execução da função em tempo real.

### 3.3.1 SNIFFER

Esse código é utilizado para capturar as informações dos pacotes SV e GOOSE da rede ethernet e decodificá-los para serem utilizados pelos outros processos. Além da decodificação, este algoritmo também realiza o cálculo da Transformada Discreta de Fourier (DFT) e disponibiliza os valores de corrente e tensão fasoriais para as funções de proteção através do SHM.

Com o objetivo de evitar a perda de pacotes durante o processamento, foi implementado um sistema de fila, no qual, sempre que um novo pacote é recebido através da interface de rede, ele é encaminhado para um *buffer* de funções, onde uma arquitetura de *threads* que trabalham de forma paralela, acatam esse pacotes e processam-os, conforme mostrado na Figure 3.8.

**Figura 3.8** – Algoritmo utilizado para processar os pacotes




**Fonte:** autoria própria

Os valores amostrados do pacote SV são armazenados em um buffer com 80 posições, o que equivale a um ciclo de onda completo. A cada 40 novas amostras, o algoritmo

realiza a DFT utilizando a biblioteca FFTW3 (FFTW, 2023) no conjunto completo das amostras e, em seguida, atualiza as medidas fasoriais de corrente e tensão na memória compartilhada (SHM).

No caso dos pacotes GOOSE, antes da execução do script Sniffer, o algoritmo Controler.py, mapeia previamente os endereços das variáveis do *dataset* conforme definido no arquivo de configuração. Portanto, quando o pacote é decodificado pelo Sniffer, ele apenas atualiza as informações nas variáveis que foram previamente criadas.

É importante destacar que essas operações são executadas em paralelo, o que torna necessário o uso de *mutex* (mecanismos de exclusão mútua) para evitar a corrupção dos dados, garantindo a integridade das informações compartilhadas entre os processos

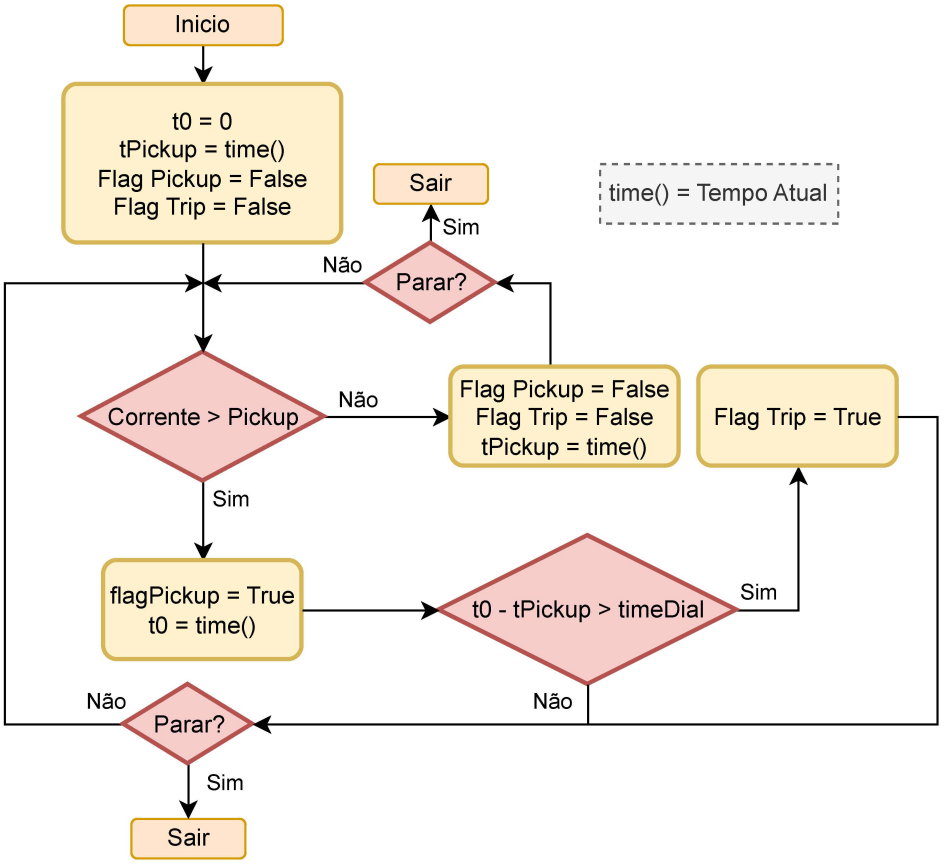
### 3.3.2 PIOC - Proteção de SobreCorrente Instantânea de Phase e Neutro

A função de proteção de sobrecorrente de tempo instantâneo (ANSI 50), também conhecida como *Protection Instantaneous Overcurrent* (PIOC) de acordo com a norma, é uma das proteções mais utilizadas e essenciais para proteger o sistema elétrico contra curto-circuitos. Esta função monitora o valor eficaz (RMS) da corrente e, caso ele exceda um limite conhecido como corrente de *pickup*, emite um sinal de comando (*Trip*) para o sistema de controle.

Na aplicação dessa proteção, foi utilizado um contador de tempo que introduz um atraso de tempo constante no envio do sinal de *Trip*, de modo a auxiliar a coordenação dessa função com outras proteções.

O código implementado para realizar essa proteção é o mesmo, tanto para as correntes de fase quanto para a de neutro, com as únicas variações sendo o valor do *pickup* e o tempo de retardo. Com isso em mente, a Figura 3.9 apresenta o diagrama de blocos do algoritmo implementado no vIED.

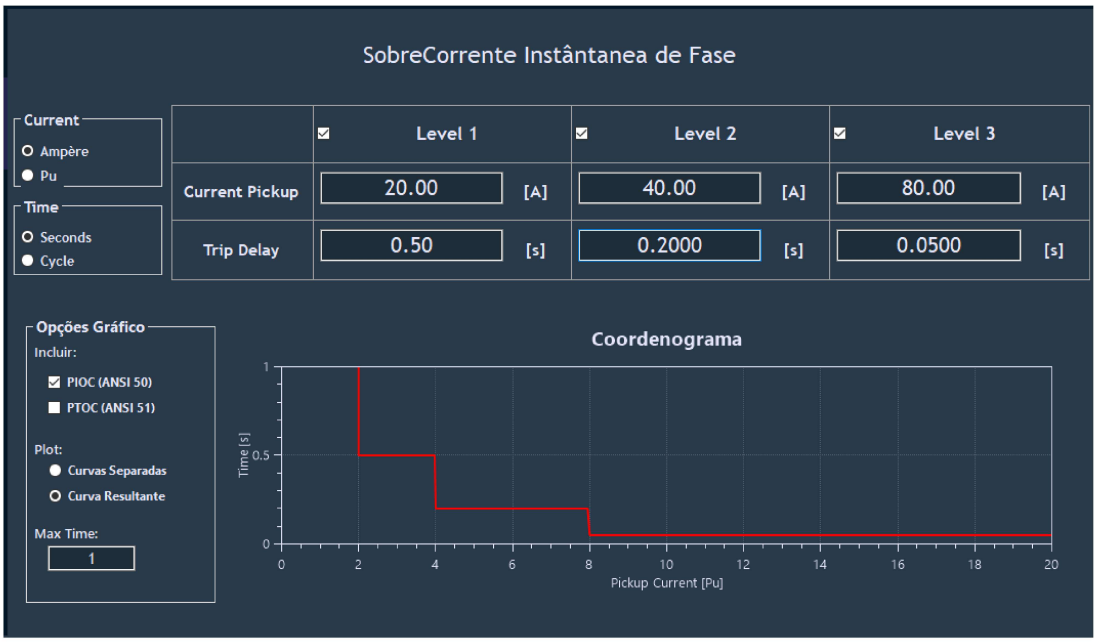
**Figura 3.9** – Lógica utilizada para implementar a função de proteção PIOC




**Fonte:** autoria própria

A parametrização dos parâmetros da função é realizada no software Desktop, através de uma interface gráfica, como pode ser visto na Figura 3.10. Nesta tela, o usuário pode definir o valor de *pickup* e o tempo de retardo.

**Figura 3.10** – Tela de parametrização da função de proteção PIOC no Software Desktop




**Fonte:** autoria própria

### 3.3.3 PTOC - Proteção de SobreCorrente de Tempo Inverso de Phase e Neutro

Essa função de proteção também é uma das utilizadas no SEP e se assemelha à PIOC. Porém esta é utilizada para proteção contra sobrecargas e outros distúrbios. Ela é carecterizada como de Tempo Inverso pois o seu tempo de atuação é inversamente proporcional a magnitude da corrente, ou seja, quanto mais alto for o valor da corrente, mais rapidamente esse função atua.

De modo análogo à PIOC, a função *Protection Timed Overcurrent* (PTOC) observa o valor RMS da corrente e compara com o valor de *pickup* pré-definido. Caso a corrente exceda esse valor de *pickup* por um período de tempo especificado pela curva de tempo inverso, a proteção envia o sinal de comando para o sistema de controle. A equação da curva desse sistema é representada pela Equação (1):

$$t_{trip} = TimeDial \times \left[ \gamma + \frac{\beta}{\left( \frac{I_{medido}}{I_{pickup}} \right)^{\alpha} - 1} \right] \tag{1}$$

O parâmetro *TimeDial* é um multiplicador que desloca verticalmente a curva, utilizado principalmente para realizar a coordenação da função. Os outros coeficientes $\gamma$, $\beta$ e $\alpha$, são parâmetros definidos pelo tipo da curva da função. Na Tabela 3.4 é apresentando o valores destes coeficientes para as curvas das famílias IEC e U.S..

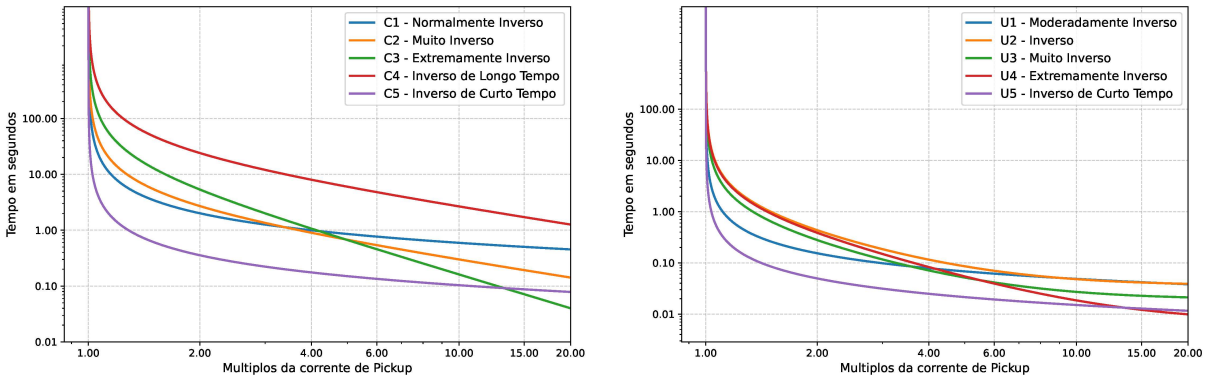
**Tabela 3.4** – Parâmetros das curvas de tempo inverso das famílias IEC e U.S.

| | **Curvas** | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|
| **Curvas IEC** | C1 (Normalmente Inverso) | 0,02 | 0,14 | 0 |
| | C2 (Muito Inverso) | 1 | 13,5 | 0 |
| | C3 (Extremamente Inverso) | 2 | 80 | 0 |
| | C4 (Inverso de Longo Tempo) | 1 | 120 | 0 |
| | C5 (Inverso de Curto Tempo) | 0,04 | 0,05 | 0 |
| **Curvas US** | U1 (Moderadamente Inverso) | 0,02 | 0,0104 | 0,0226 |
| | U2 (Inverso) | 2 | 5,95 | 0,180 |
| | U3 (Muito Inverso) | 2 | 3,88 | 0,0963 |
| | U4 (Extremamente Inverso) | 2 | 5,67 | 0,0352 |
| | U5 (Inverso de Curto Tempo) | 0,02 | 0,00342 | 0,00262 |

**Fonte:** (IEEE, 1996) e (IEC, 1989)

Na Figura 3.11 é apresentando os gráfico contendo as curvas de cada família para o *timeDial* de 0,2.

**Figura 3.11** – Gráfico de cada umas curvas de tempo inverso nas mesmas condições

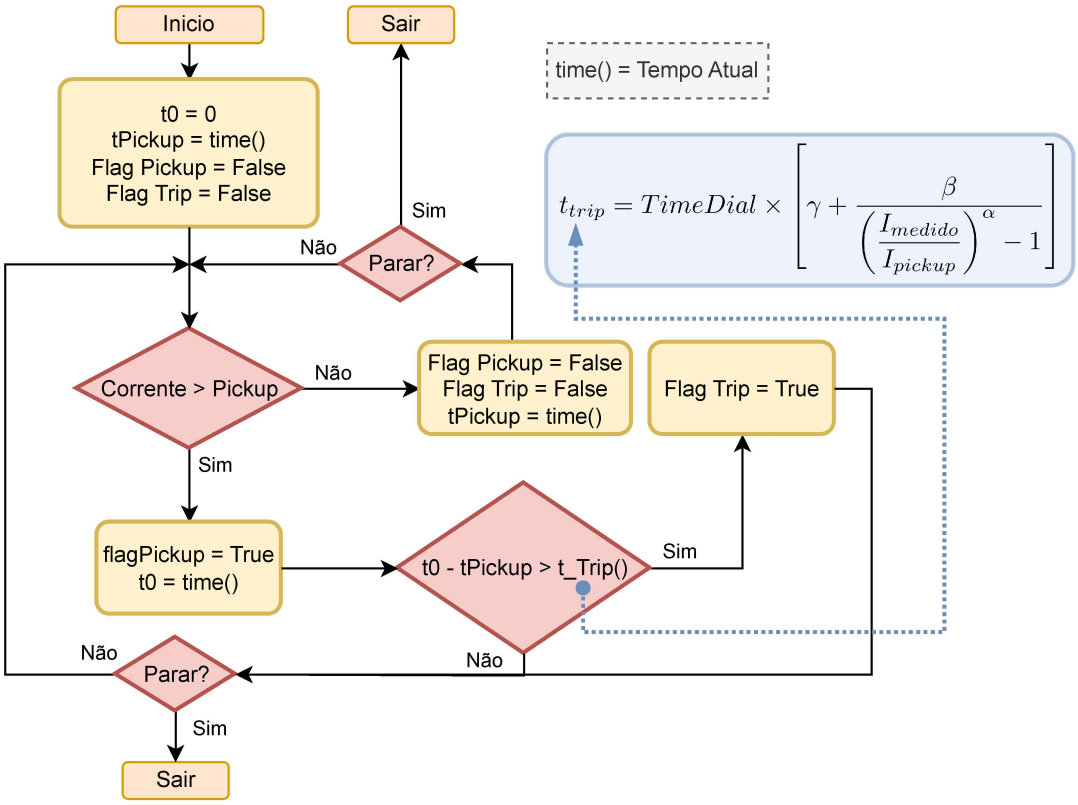



**(a)** IEC **(b)** U.S

**Fonte:** autoria Própria

A lógica implementada no algoritmo da função PTOC é semelhante à da função PIOC, diferindo apenas no tempo de atuação, no qual é utilizada uma das equações das curvas mencionadas anteriormente. A Figura 3.12 apresenta o diagrama de blocos da lógica implementada no vIED.

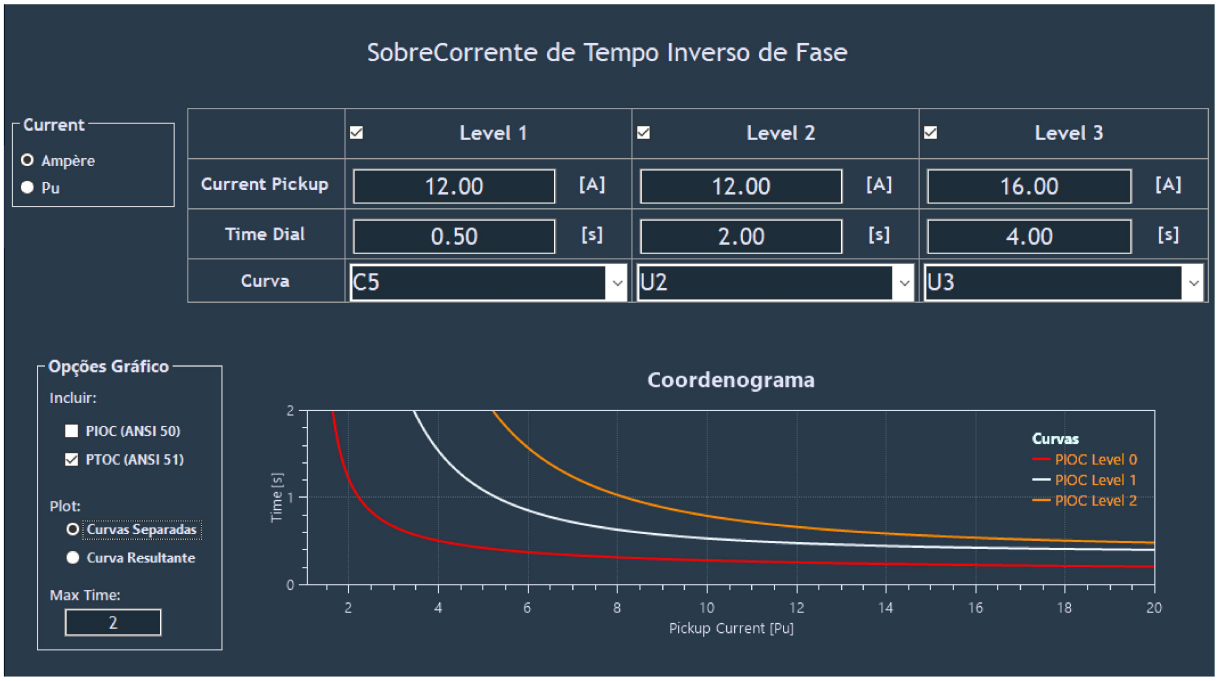
**Figura 3.12** – Lógica utilizada para implementar a função de proteção PIOC




**Fonte:** autoria própria

A interface de parametrização da função PTOC pode ser vista na Figura 3.13. Nesta tela, o usuário pode definir o valor da corrente de *pickup*, o *TimeDial* e uma das curvas supracitadas.

**Figura 3.13** – Tela de parametrização da função de proteção PTOC no Software Desktop



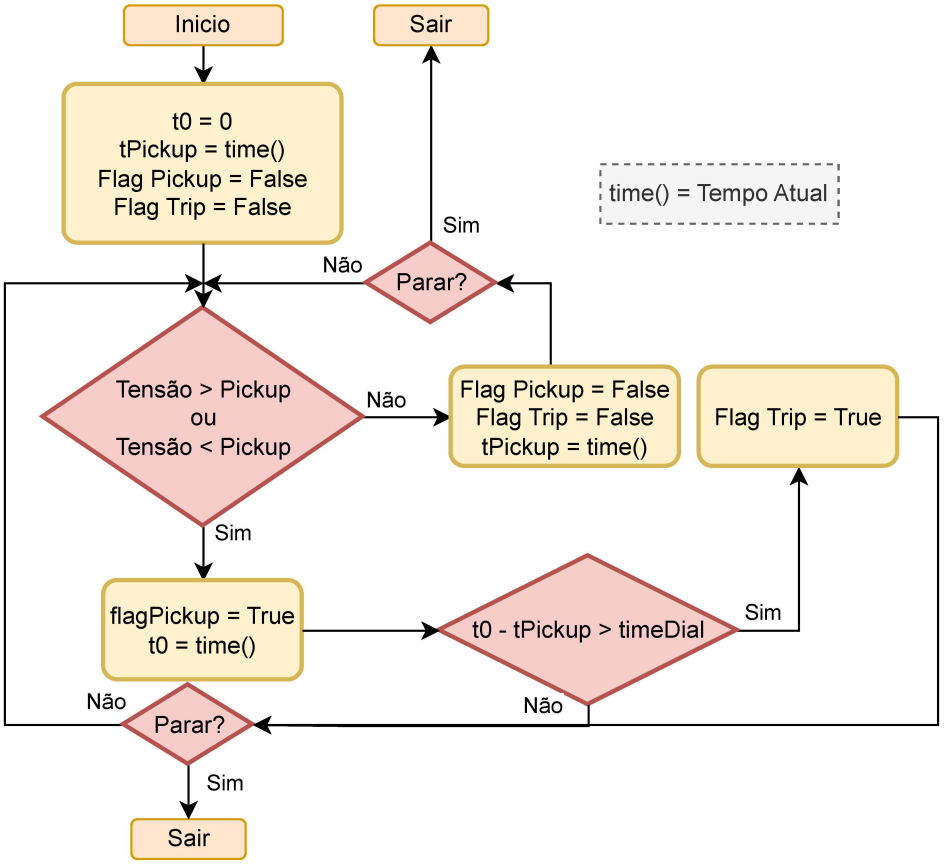

**Fonte:** autoria própria

### 3.3.4 PTUV e PTOV - Proteções de Sobre e Sub Tensão de Phase

As proteções *Protection Under-Voltage* (PTUV) e *Protection Over-Voltage* (PTOV) protegem o sistema elétrico contra sobre e sub tensões. A sua atuação pode ser em função de uma curva de tempo inverso ou possuir tempos fixos de atuação. No vIED desenvolvido, foi utilizado a abordagem de tempo de atuação fixo, similar a função PIOC.

Ambas as funções de sub e sobre tensão foram implementadas em serviços diferentes no vIED. Porém como ambas possuem algoritmos análogos, na Figura 3.14 é mostrado a logica de implementação em conjunto.

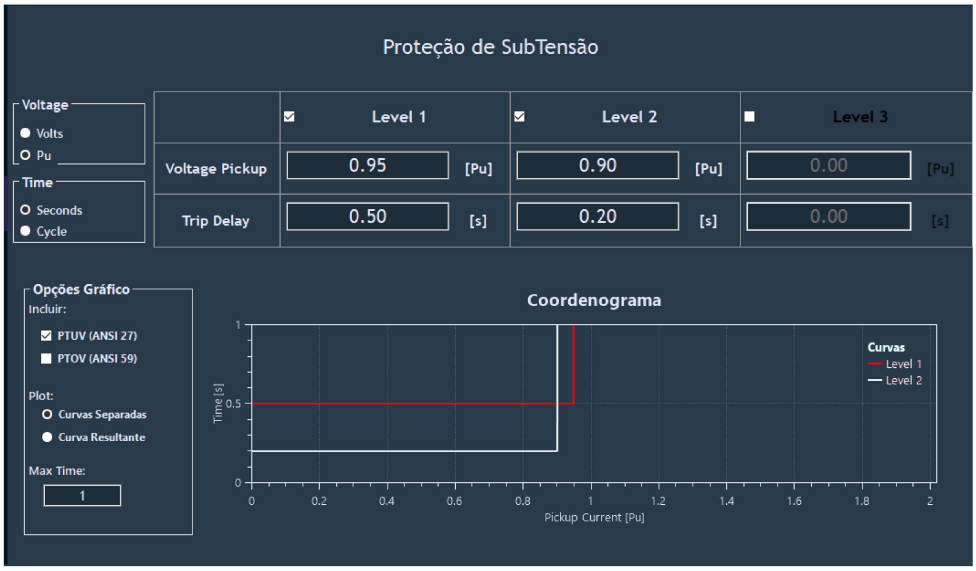
**Figura 3.14** – Lógica utilizada para implementar a função de proteção PTUV




**Fonte:** autoria própria

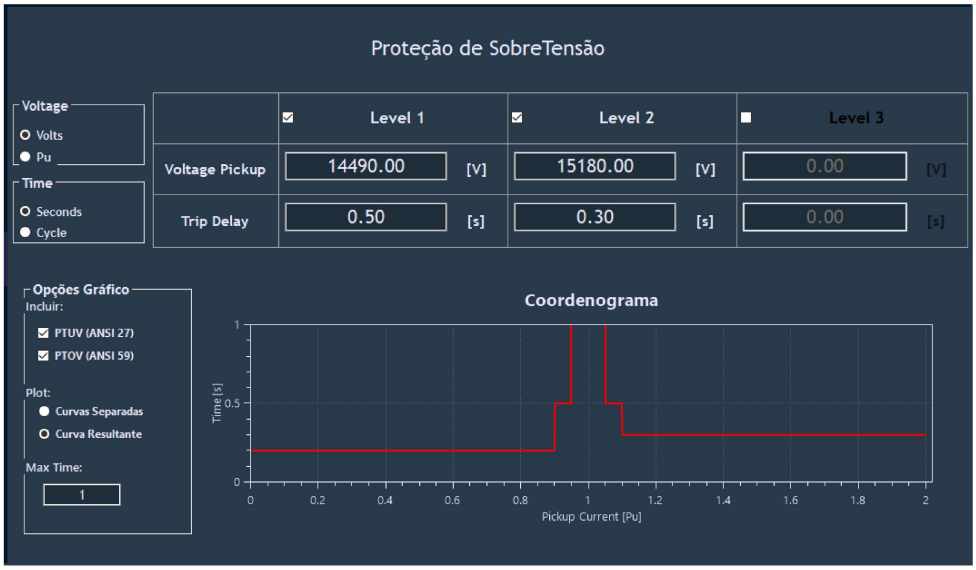
As telas do software de parametrização são indicadas nas Figuras 3.15 e 3.16. Nele é possível determinar o valor da tensão de *pickup* e o tempo de atraso da função.

**Figura 3.15** – Tela de parametrização da função de proteção de Subtensão




**Fonte:** autoria própria

**Figura 3.16** – Tela de parametrização da função de proteção de SobreTensão



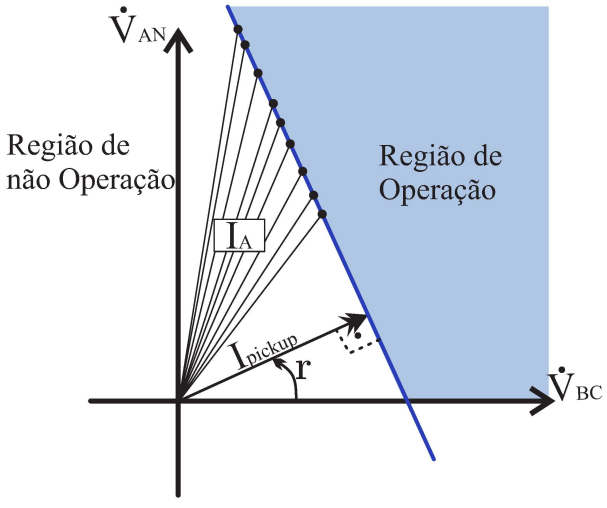

**Fonte:** autoria própria

### 3.3.5 PDIR - Proteção direcional de Sobrecorrente

A função de direcionalidade, utilizada para coordenar a atuação do relé em sistemas radiais, inserindo ao sistema proteção características direcionais para as funções de sobrecorrente de modo que o relê atue apenas para faltas que ocorram a sua frente.

A operação dessa proteção ocorre quando a corrente medida na fase ultrapassa um certo valor de operação que é ponderado em relação a defasagem angular entre a sua corrente e um outro referencial. Neste trabalho foi utilizado apenas a polaridade de 90° para representar o referencial da corrente. Neste sistema, a corrente de uma fase é ponderado em relação à defasagem angular entre as tensões das outras duas fases. Este modelo é indicado pela Figura 3.17, considerando o sistema equilibrado e a análise na fase A.

**Figura 3.17** – Representação gráfica da zona de operação da proteção direcional de corrente



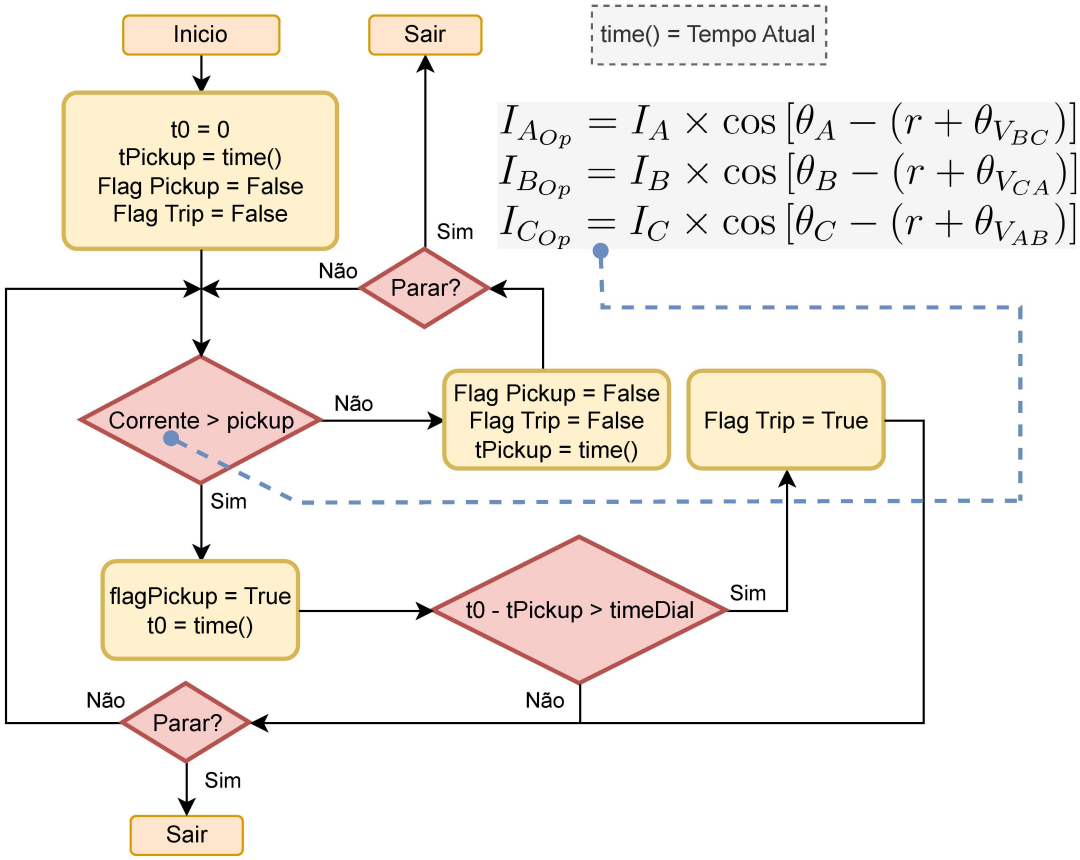

**Fonte:** autoria própria

No gráfico é perceptível que para correntes com ângulo distante do ângulo de torque $r$, o seu valor precisa ser maior para alcançar a região de operação em azul. Pois, a corrente de operação é o valor da corrente de fase projetada na mesma direção da corrente de *pickup*, o seu valor pode ser estimado pela Equação 2.

$$\begin{aligned} I_{A_{Op}} &= I_A \times \cos\left[\theta_A - (r + \theta_{V_{BC}})\right] \\ I_{B_{Op}} &= I_B \times \cos\left[\theta_B - (r + \theta_{V_{CA}})\right] \\ I_{C_{Op}} &= I_C \times \cos\left[\theta_C - (r + \theta_{V_{AB}})\right] \end{aligned} \tag{2}$$

Para o relé atuar, a corrente de operação deve ser maior que a corrente de *pickup* defina na função. A algoritmo implementado para realizar essa função no vIED é indicado na Figura 3.18

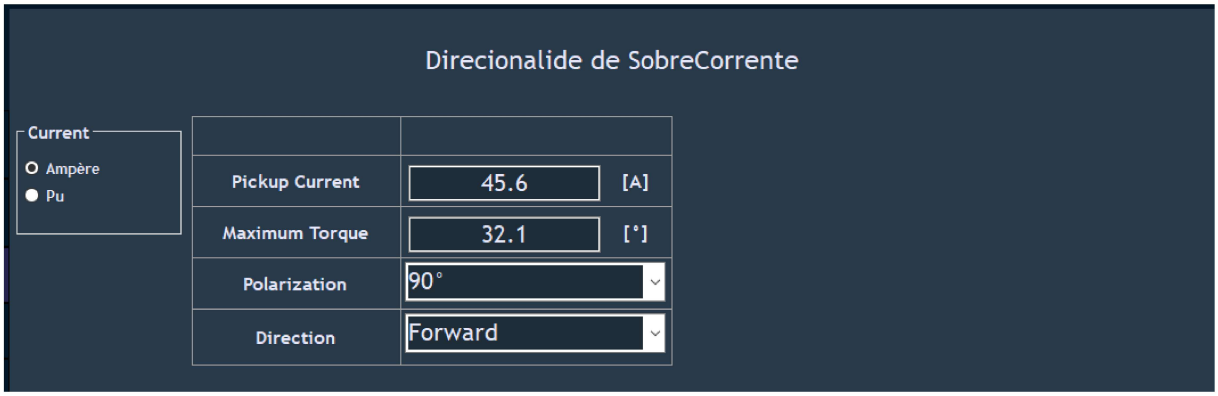
**Figura 3.18** – Lógica utilizada para implementar a função de proteção PDIR




**Fonte:** autoria própria

A parametrização da função leva em consideração a corrente de *pickup*, o torque máximo, a sua foram de polarização e se a direção que deseja olhar, para frente (forward) ou para trás do relé (reverse). Esses parâmetros são indicados na Figura 3.19.

**Figura 3.19** – Tela de parametrização da função de proteção PDIR no Software Desktop




**Fonte:** autoria própria

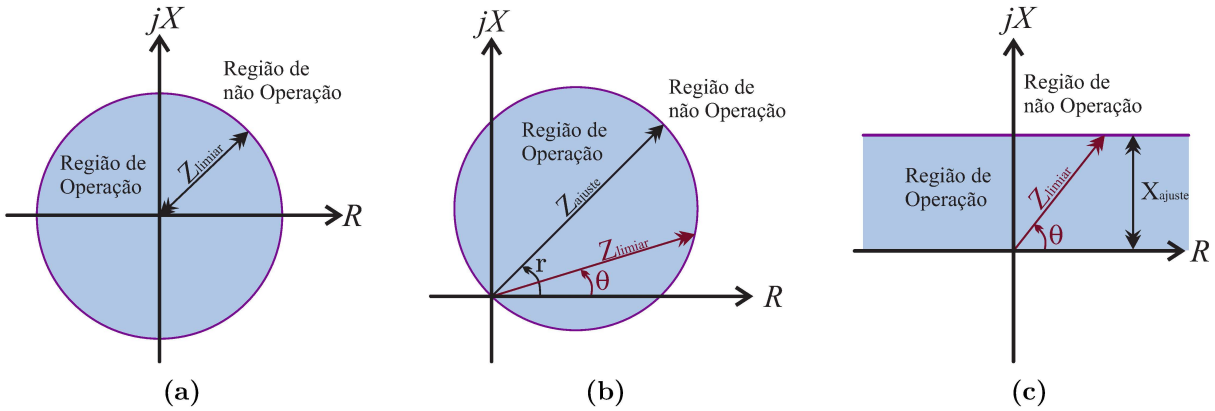
### 3.3.6 PDIS - Proteção de Distância

A proteção de distância, *Protection Distance* (PDIS), é amplamente utilizada em linhas de transmissão por causa da sua capacidade de detectar faltas elétricas com base na impedância vista pelo relê e pela grande quantidade de redundâncias que a sua característica insere ao sistema.

Essa função é parametrizada com base na impedância da linha, de modo que na ocorrência de uma falta dentro da sua região de alcance, o valor da impedância vista pelo IED é reduzida drasticamente.

Existem 4 tipos convencionais dessa proteção: Impedância, Admitância, Reatância e Quadrilateral. Neste trabalho foi utilizado apenas os 3 primeiros tipos que estão retratados no plano $R \times jX$ da Figura 3.20.

**Figura 3.20** – Zonas de operação dos tipos de proteção de distância implementados, (a) Impedância, (b) Admitância, (c) Reatância



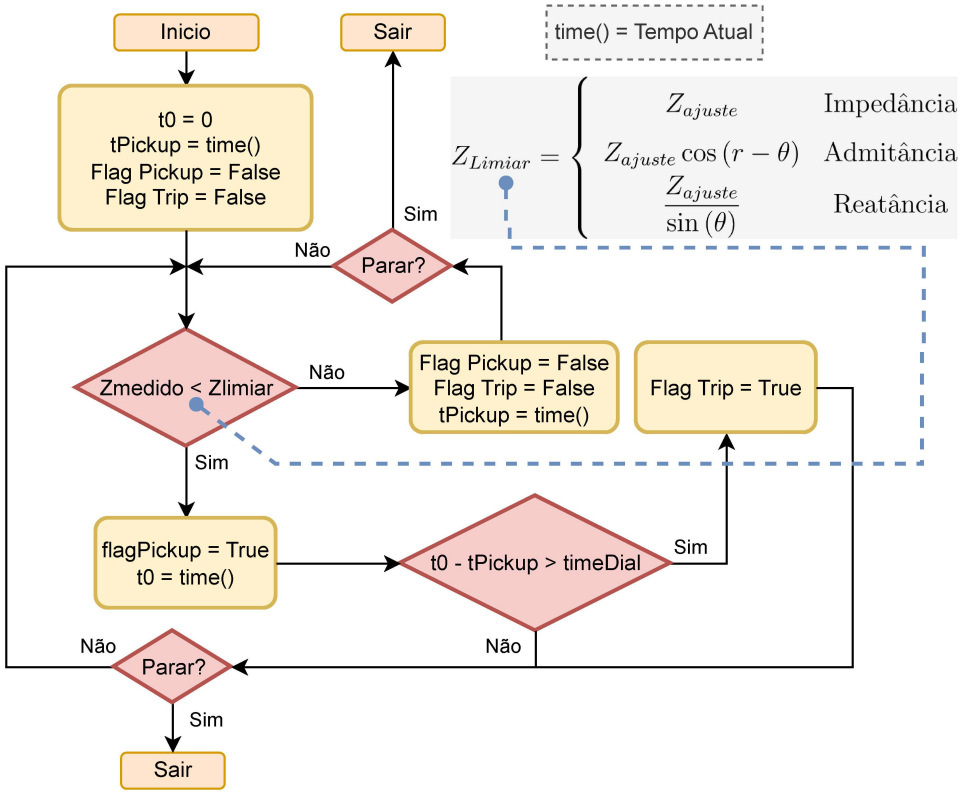

**Fonte:** autoria própria

IED opera quando a impedância medida encontra-se dentro da região de operação indicada na figura. As equações que definem a impedância limiar que delimita essas regiões são mostradas pela Equação (3) (HOROWITZ; PHADKE, 2014).

$$Z_{Limiar} = \begin{cases} Z_{ajuste} & \text{Impedância} \\ Z_{ajuste} \times \cos(r-\theta) & \text{Admitância} \\ \dfrac{X_{ajuste}}{\sin(\theta)} & \text{Reatância} \end{cases} \quad (3)$$

A lógica de operação implementada no vIED para executar está função é indicada na Figura 3.21.

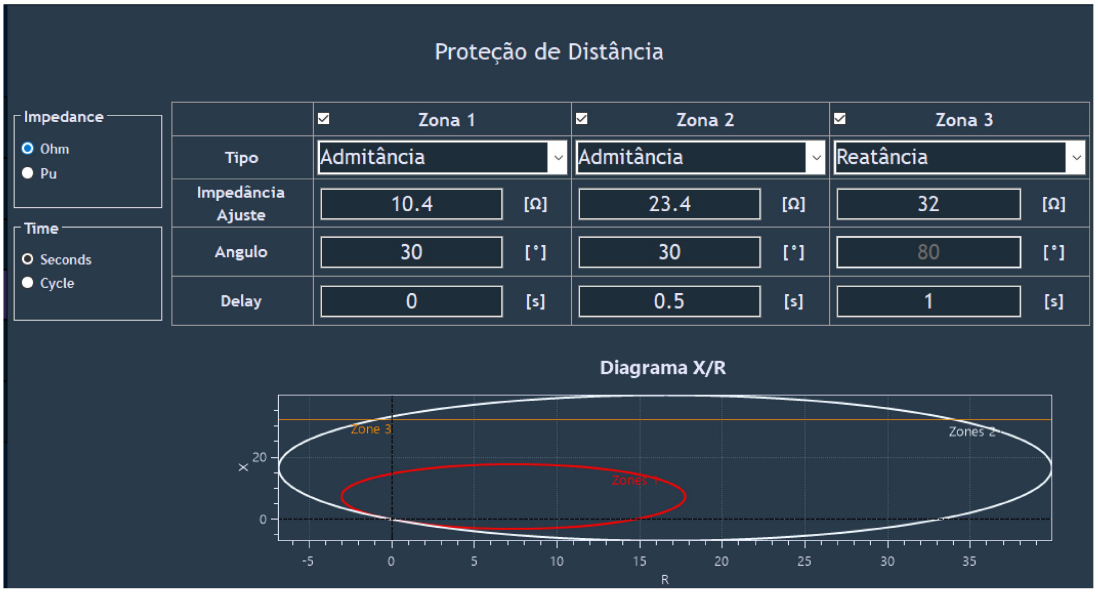
**Figura 3.21** – Lógica utilizada para implementar a função de proteção PDIS




**Fonte:** autoria própria

A parametrização da função permite a utilização de 3 zonas diferentes, recebendo como parâmetro a impedância de ajuste, o ângulo de torque máximo e o tempo de atraso para a sua atuação, conforme mostrado na Figura 3.22.

**Figura 3.22** – Tela de parametrização da função de proteção PDIS no Software Desktop

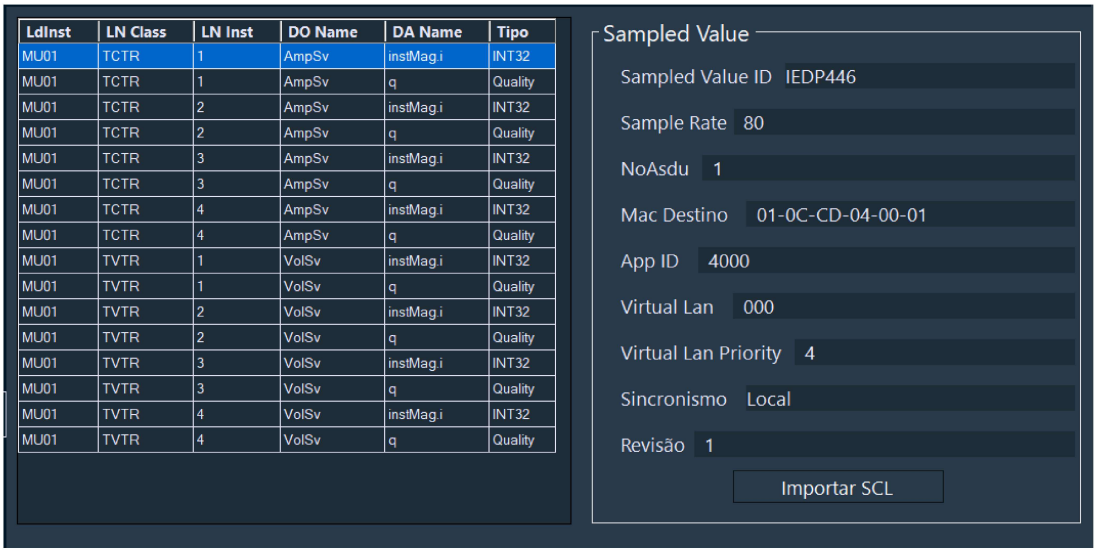



**Fonte:** autoria própria

### 3.3.7 Configuração do Protocolo *Sampled value*

No IED proposto foi considerado a entrada de dados de apenas uma Merging Unit, utilizando o pacote *Sampled value*. A configuração deste pacote, no qual o vIED espera receber, é mostrado pela Figura 3.23. Estes dados podem ser preenchidos manualmente ou importados utilizando o arquivo de configuração SCL da MU.

**Figura 3.23** – Tela de configuração do protocolo *Sampled value*



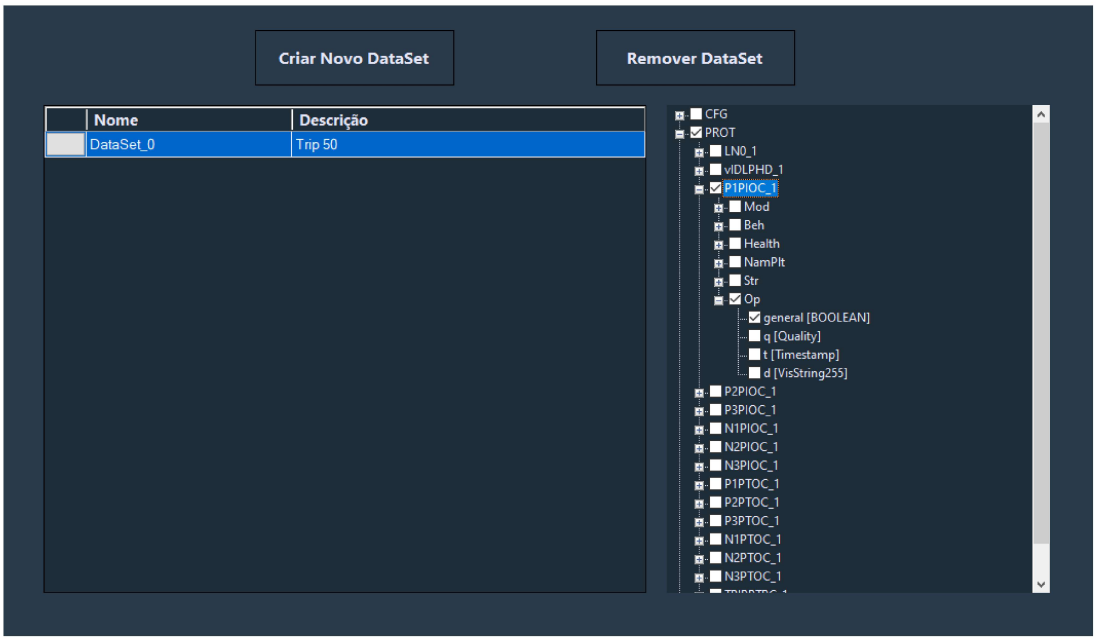

**Fonte:** autoria própria

### 3.3.8 Configuração do Protocolo GOOSE

Nesta configuração dos pacotes GOOSE para envio, o software permite a criação de múltiplos *dataSets*, no qual são escolhidos os dados que serão transmitidos nas mensagens

do protocolo. Essa parametrização é mostrada pela Figura 3.24.

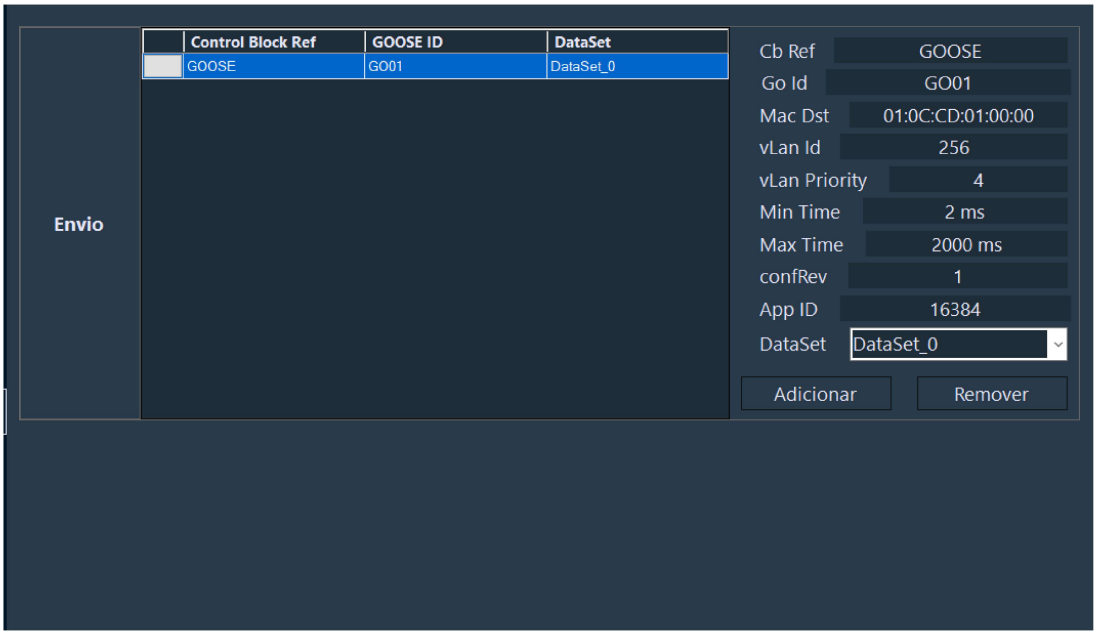
**Figura 3.24** – Tela de criação dos conjuntos de dados enviados pelo GOOSE




**Fonte:** autoria própria

O IED permite o envio de até 10 mensagens GOOSE distintas, estas são parametrizadas conforme indicado na Figura 3.25.

**Figura 3.25** – Tela de configuração do protocolo GOOSE




**Fonte:** autoria própria

Ao final, o software possibilita a emissão do arquivo SCL do vIED contendo todas as informações parametrizadas anteriormente.

## 3.4 Virtual Merging Unit

Para permitir o teste e validação da arquitetura de proteção e controle de forma prática no ambiente virtual, foi criado um testador virtual chamado Virtual Merging Unit (vMU). Esse testador é capaz de enviar continuamente pacotes SV e replicar uma sequência de estados predefinidos pelo usuário, a fim de testar o funcionamento dos relés em condições nominais, distúrbios ou faltas.

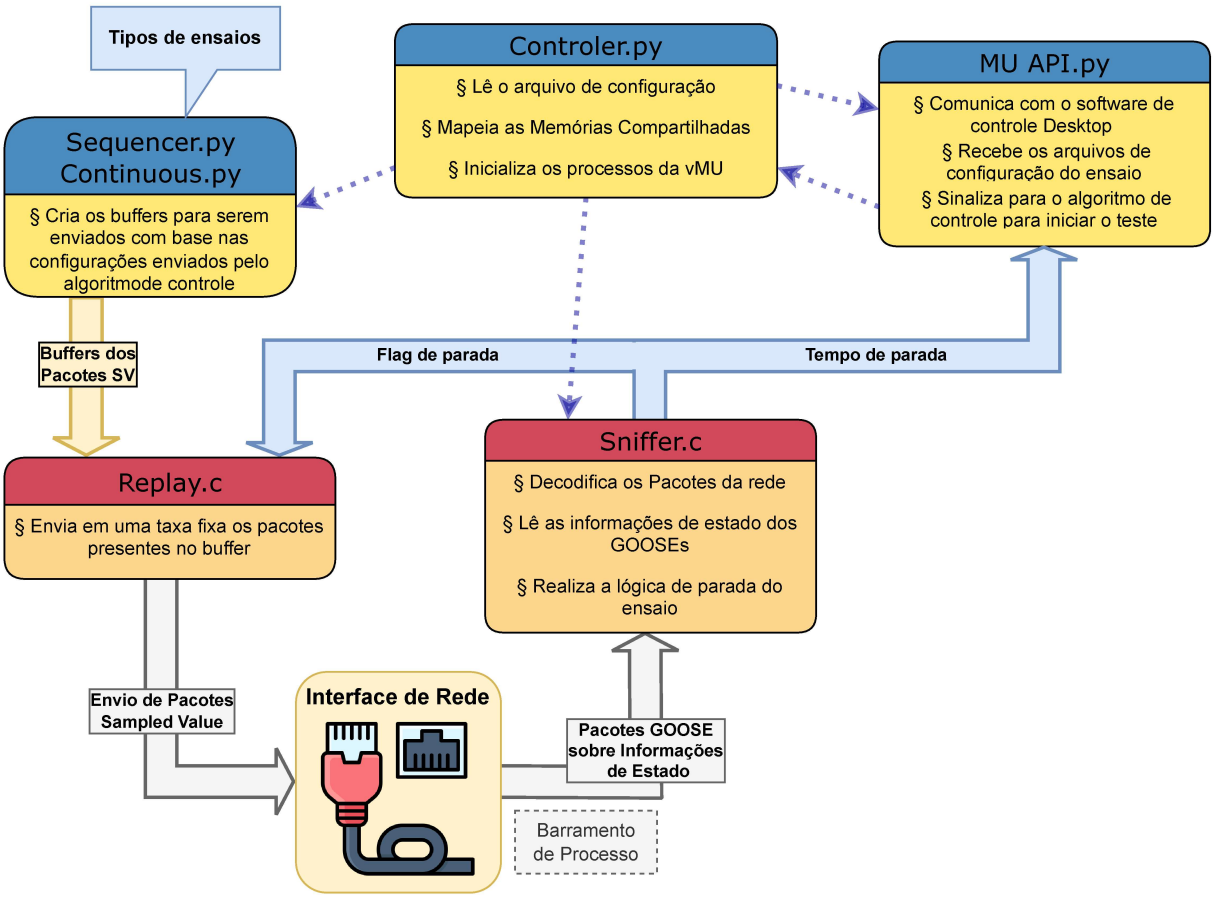
Semelhante ao vIED, o vMU utiliza um algoritmo de baixo nível em execução dentro de uma máquina virtual no servidor. Este foi implementado nas linguagens C e Python. Além disso, também possui uma interface gráfica desenvolvida em C# para o sistema operacional Windows, que permite ao operador configurar os testes desejados. A tela inicial desse software é exibida na Figura 3.26 abaixo.

**Figura 3.26** – Tela inicial do software de configuração do vMU


**Fonte:** autoria própria

Para a máquina virtual do servidor, foram desenvolvidos os seguintes algoritmos, ilustrados na Figura 3.27:

**Figura 3.27** – Algoritmos implementados no software do MU virtual




**Fonte:** autoria própria

A descrição funcional de cada unidade é resumidamente explicada na sequência.

**Controle.py:** Este é o código principal do vMU, sendo responsável por ler os arquivos de configuração e iniciar os outros serviços. Além disso, ele faz o mapeamento das memórias compartilhadas na RAM para que os outros serviços possam utilizá-las e inicia os testes.

**MU API.py:** Este algoritmo é responsável pela comunicação entre a máquina virtual e o software desktop de configuração. Ele recebe os arquivos de configuração do usuário e envia sinais para o **Controle.py** a fim de iniciar os testes. Além disso, envia os resultados dos ensaios de volta para o desktop.

**Sequencer.py:** Este algoritmo executa os ensaios de sequências, nos quais cria um conjunto de buffers contendo os pacotes SV conforme especificado no arquivo de configuração do operador. Em seguida, ele inicia o processo **Replay.c** para replicá-los.

**Continuous.py:** Este algoritmo realiza ensaios de envio contínuo de pacotes SV, criando apenas um buffer que será replicado em loop pelo processo **Replay.c**.

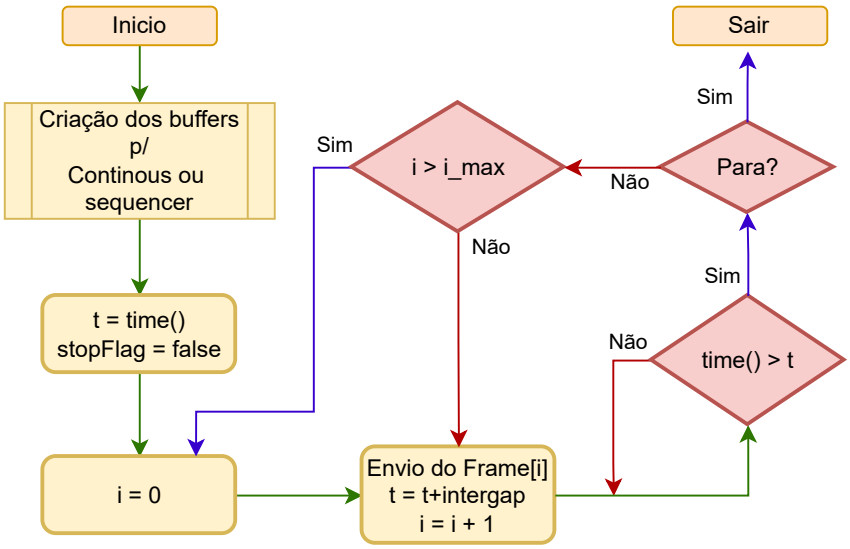
**Replay.c:** Este componente envia os pacotes *Sampled value* pela rede Ethernet em uma taxa fixa.

**Sniffer.c:** Funciona como um algoritmo de monitoramento dos testes, capturando os pacotes GOOSE na rede e verificando seu estado.

### 3.4.1 Envio dos pacotes

No envio dos pacotes *Sampled value* foi utilizado o seguinte algoritmo.

**Figura 3.28** – Algoritmo de reprodução e envio dos Pacotes *Sampled value*



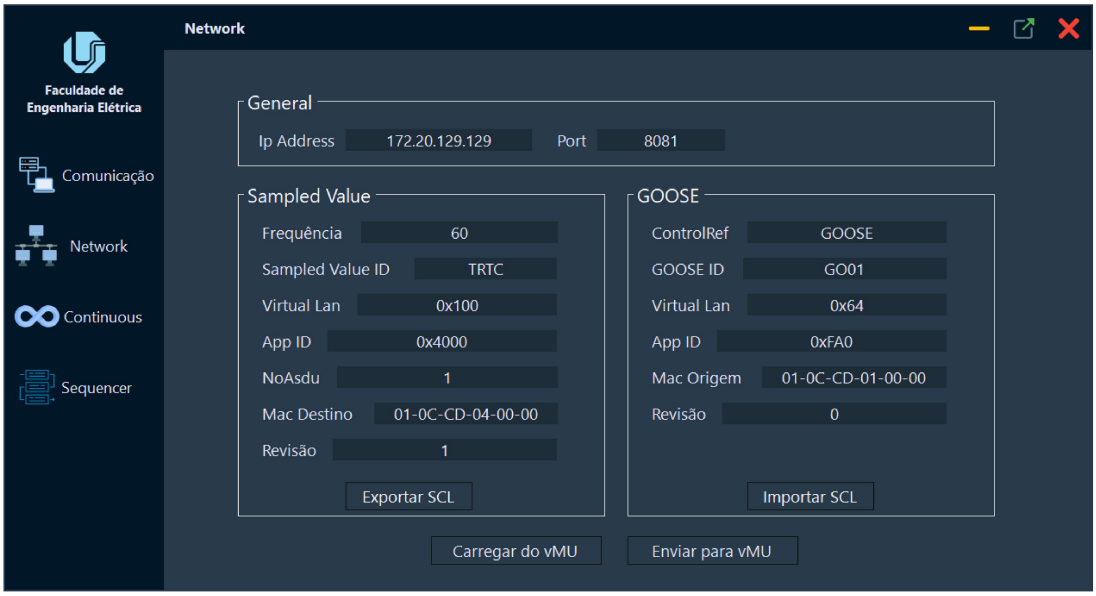

**Fonte:** autoria própria

Neste modelo, o tempo de envio entre cada pacote é sempre o tempo inicial de envio mais uma constante (*intergap*). Por esse motivo, caso um pacote seja atrasado no meio da execução dos testes, os outros pacotes subsequentes não serão influenciados.

### 3.4.2 Configurações dos protocolos

Na configuração dos protocolos da vMU, o operador consegue definir a frequência de envio, a quantidade de amostras por pacote (NoAsdu) e outros parâmetros da rede, conforme mostrado na Figura 3.29. Além do protocolo SV, nesta tela também é configura o pacote GOOSE no qual contém as informações de estado que a vMU espera receber.

**Figura 3.29** – Tela de configuração dos protocolos do vMU




**Fonte:** autoria própria

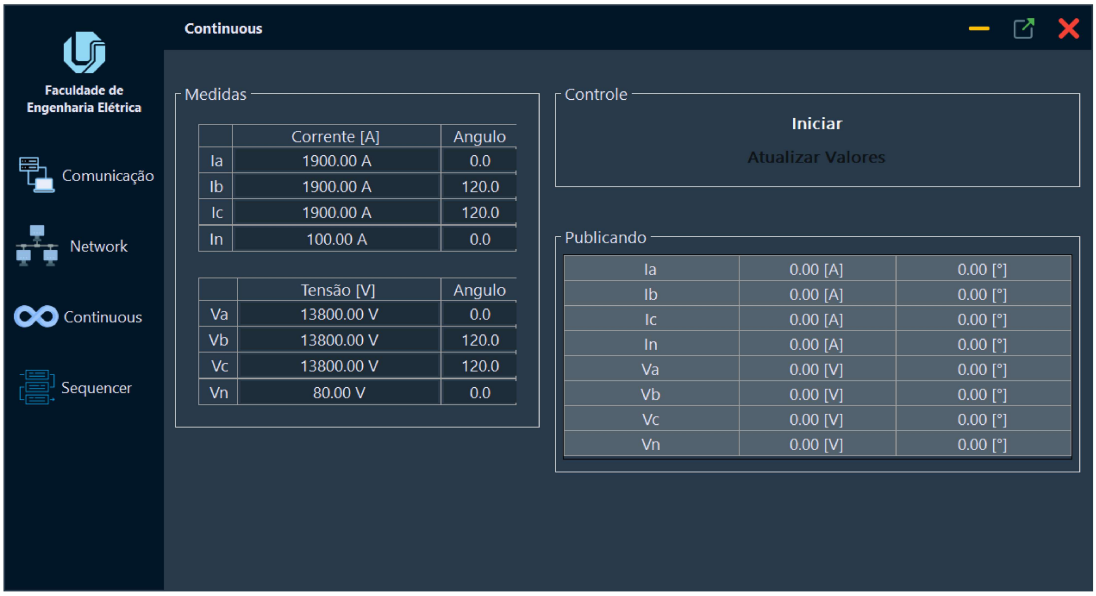
O GOOSE pode ser configurado de duas maneiras: manualmente, utilizando os parâmetros exibidos na figura, ou importando um dos arquivos de configuração SCL que contenham o bloco de controle do GOOSE desejado. Neste trabalho, foi considerado apenas um bloco de controle GOOSE de entrada, com todos os seus dados sendo do tipo booleano.

A parametrização do protocolo SV é realizada manualmente, e é possível exportar as configurações definidas para um arquivo SCL, que pode ser posteriormente utilizado para configurar o IED. Foi considerada uma configuração fixa com 4 canais de corrente e 4 de tensão.

### 3.4.3 Ensaio Continuous

O ensaio contínuo envolve a transmissão contínua dos pacotes *Sampled value*, simulando uma condição normal de operação. Esse modo foi empregado para reproduzir o tráfego nominal na rede Ethernet da subestação. Na Figura 3.30, é apresentado a configuração dessa função.

**Figura 3.30** – Tela de configuração do ensaio Continuous do vMU



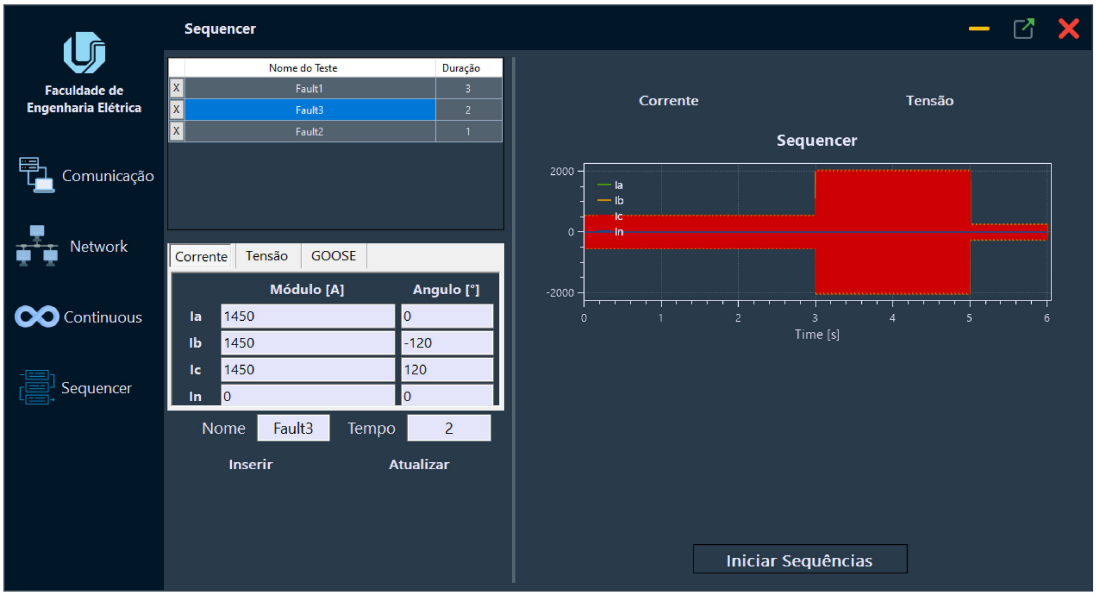

**Fonte:** autoria própria

Neste ensaio, o operador define as gradezas fasoriais de corrente e tensão que serão enviadas, e uma vez iniciado o teste, é possível atualizar esses dados em tempo real (*hot-line*), sem a necessidade de parar o envio.

### 3.4.4 Ensaio Sequencer

O ensaio de Sequências tem como finalidade replicar um conjunto de sequências definidas pelo usuário para simular uma condição de falta. A configuração desta função está indicada na Figura 3.31.

**Figura 3.31** – Tela de configuração do ensaio Sequencer do vMU




**Fonte:** autoria própria

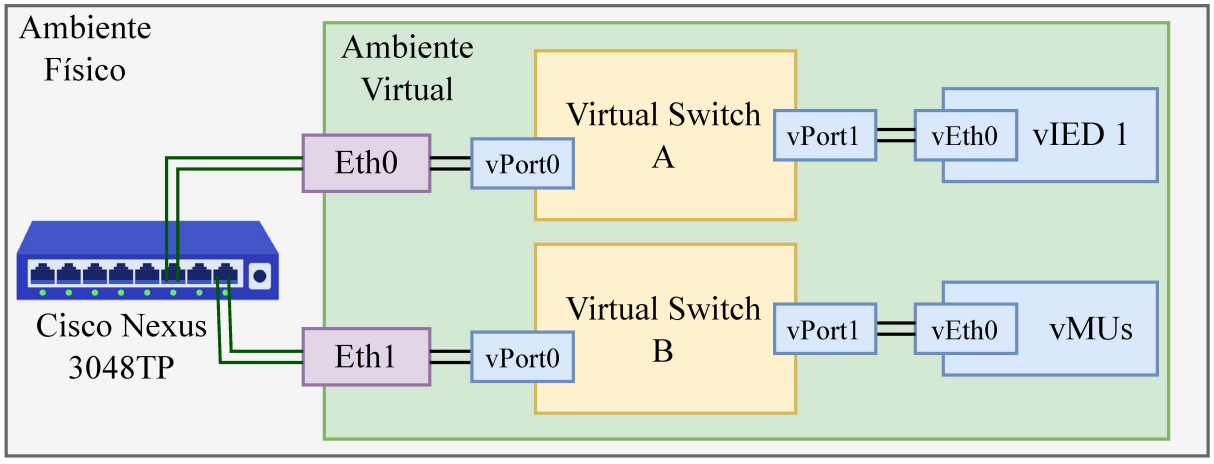
Neste ensaio, o operador define uma sequência de estados, especificando os valores fasoriais de corrente e tensão, bem como a duração em que cada estado será simulado.

## 3.5 Validação e testes realizados

A principal preocupação na transição de hardware dedicado para o ambiente virtual é se os equipamentos manterão a mesma eficiência e precisão. Portanto, nesta seção, serão apresentados os resultados dos testes realizados para verificar e validar o desempenho dos dispositivos de proteção virtualizados desenvolvidos neste trabalho.

Os testes incluem a análise do tempo de resposta do IED criado e a precisão do envio dos pacotes SV na Merging Unit virtual. Nesse cenário, foram utilizados dois Switches virtuais para separar o tráfego do vIED e da vMU. Cada um dos Switches possuía uma porta de conexão externa, que era conectada a outro switch físico. Dessa forma, esperava-se simular o tráfego gerado por uma MU no pátio da subestação até chegar à máquina virtual do IED. Esse arranjo é ilustrado na Figura 3.32.

**Figura 3.32** – Topologia de rede utilizada durantes os ensaios de validação do vIED e VMU




**Fonte:** autoria própria

A seguir são apresentados os testes realizados no trabalho que avaliam a velocidade de resposta do IED Virtual e a precisão no envio das grandezas elétricas amostradas pela Merging Unit Virtual.

### 3.5.1 Teste 1 - Validação do tempo de resposta do vIED

Este primeiro ensaio busca verificar o tempo de atraso das funções de proteção em relação a quantidade de Merging Units conectadas na rede Ethernet. Cada vMU enviou um pacote *Sampled value* com as seguintes configurações:

- Taxa de Envio (SmpRate): 4800Hz (80/cycle)
- Virtual Lan ID: 100

- Virtual Lan Prioridade: 4
- N° de canais: 8 - 4 de corrente e 4 de tensão
- Tamanho do Pacote: 117 bytes

Para cada teste das funções de proteção foram colocadas na rede as seguintes quantidade de MU: 1, 10, 20, 25, 30. Desse modo, busca-se verificar o aumento do atraso da atuação das funções com o aumento do tráfego da rede gerada pelas MU.

O tempo de resposta do vIED levou em consideração o tempo em que ele enviou a mensagem GOOSE (identificando o *trip*), esta mensagem foi transmitida pelo Switch físico da Figura 3.32, até o instante em que a vMU recebeu o GOOSE enviado pelo IED. Ou seja, utilizando a função "clock_gettime" e o *clock* monotonic do linux com precisão de nanosegundos. A vMU marcou o tempo desde o início do ensaio até receber a mensagem de *trip* do vIED.

### 3.5.2 Teste 2 - Avaliação da variação entre os tempos de envio da vMU

Neste segundo ensaio é avaliado a precisão do envio dos pacotes *Sampled value* pela vMU. Para isto, foram capturado os pacotes na entrada do Switch A da Figura 3.32 utilizando o software de captura do Linux chamado *Tcpdump* com a opção de *hardware-timestamp* do adaptador de rede.

Neste teste foi utilizado o modo Continuous da vMU com os seguintes dados:

- Taxa de Envio (SmpRate): 4800Hz (80/cycle)
- Tempo entre Frames (interGap): 208,3333 $\mu s$
- Virtual Lan ID: 100
- Virtual Lan Prioridade: 7
- N° de canais: 8 - 4 de corrente e 4 de tensão
- Tamanho do Pacote: 117 bytes

# 4 RESULTADOS E DISCUSSÕES

## 4.1 Resultados do Teste 1 - Validação do tempo de resposta do vIED

A seguir é apresentado o tempo de resposta do vIED para as proteções: PIOC, PTOC e PDIS. Nelas, foram realizadas 30 repetições da atuação da proteção para cada quantia de Merging Units na rede, os resultados obtidos são mostrados no gráfico abaixo.

No eixo das ordenadas dos gráficos a seguir, à esquerda está mostrando o tempo de atraso medido em relação ao teórico, e no lado direito, o tempo medido da atuação da proteção.

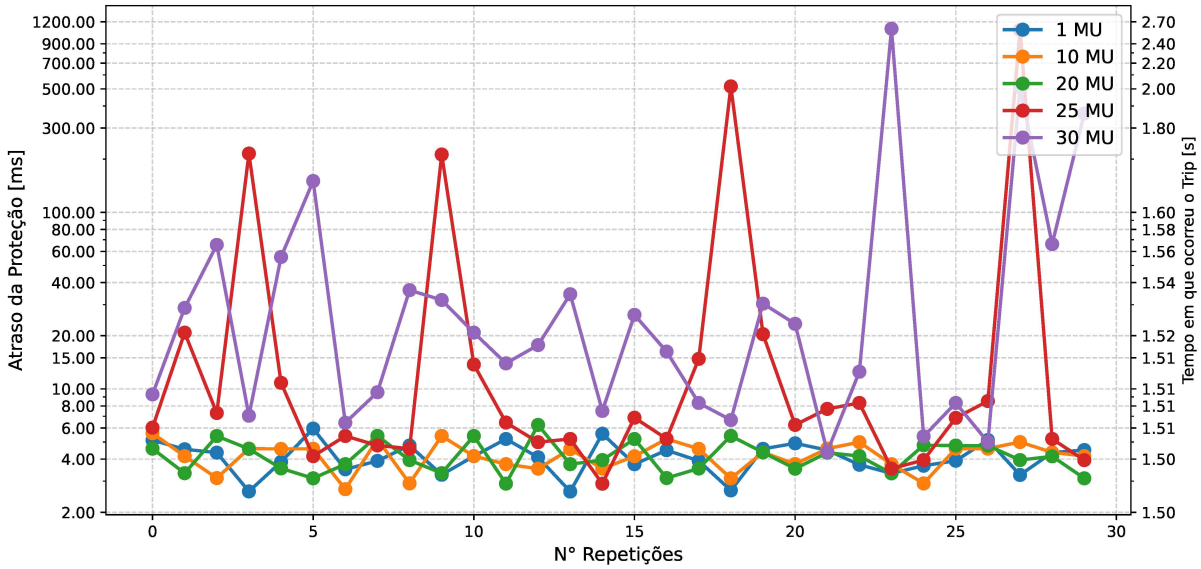
**PIOC** – A primeira função de proteção a ser testada foi a Proteção de SobreCorrente Instantânea (ANSI 50), as parametrizações do IED e da MU são mostradas abaixo.

**Tabela 4.1** – Parametrização do teste da função PIOC

| vIED | | vMU | | |
|---|---|---|---|---|
| Parâmetro | Valor | Sequência | Valor | Tempo |
| Pickup | 2400 A | Pre-Falta | 1200 A | 1 s |
| Tempo de Atraso | 0.5 s | Falta | 4800 A | 3 s |
| | | Pos-Falta | 400 A | 2 s |

Com base na sua parametrização, o tempo de resposta teórico desta proteção é 1,50 s, considerando o tempo de 1,00 s da Pre-Falta e 0,50 s da atuação da função de proteção. Os resultados obtidos são mostrados no gráfico da Figura 4.1

**Figura 4.1** – Tempo de atraso na atuação da proteção PIOC em relação ao número de MUs na rede




**Fonte:** autoria própria

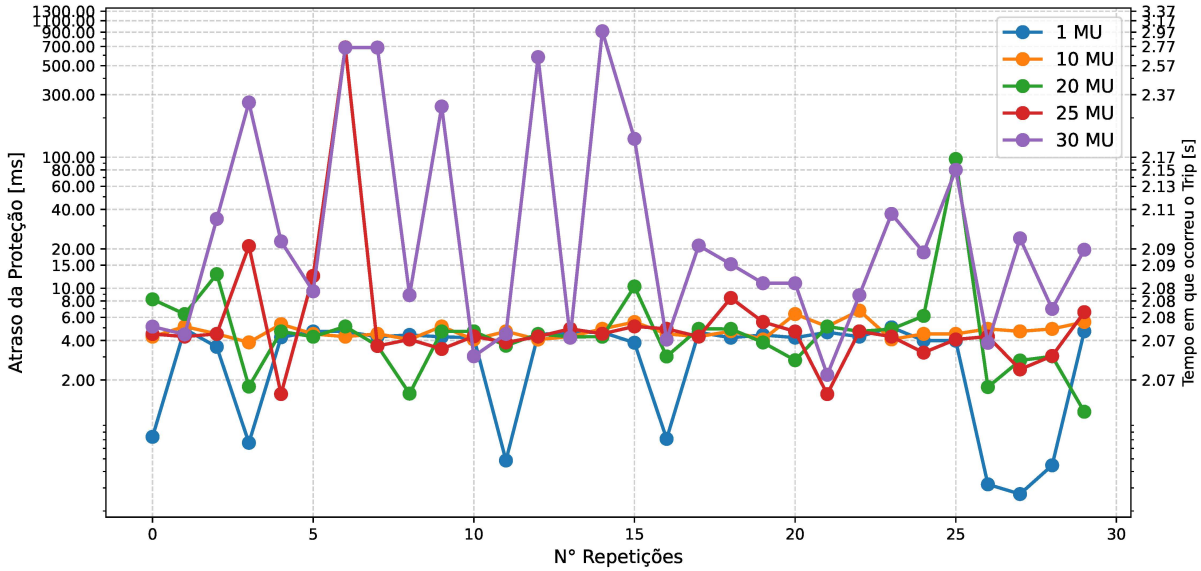
**PTOC** – A segunda função de proteção foi a Proteção de SobreCorrente de Tempo Inverso, a sua parametrização é indicada na Tabela 4.2.

**Tabela 4.2** – Parametrização do teste da função PTOC

| vIED | | vMU | | |
|---|---|---|---|---|
| Parâmetro | Valor | Sequência | Valor | Tempo |
| Pickup | 1320 A | Pre-Falta | 1200 A | 1 s |
| Curve | C1 | Falta | 4800 A | 3 s |
| Time Dial | 0,2 s | Pos-Falta | 400 A | 2 s |

Com base na sua parametrização, o tempo de resposta teórico desta proteção é 2,07 s, considerando o tempo de 1,00 s da Pre-Falta e 1,07 s da atuação da função de proteção. Os resultados obtidos são mostrados no gráfico da Figura 4.2

**Figura 4.2** – Tempo de atraso na atuação da proteção PTOC em relação ao número de MUs na rede




**Fonte:** autoria própria

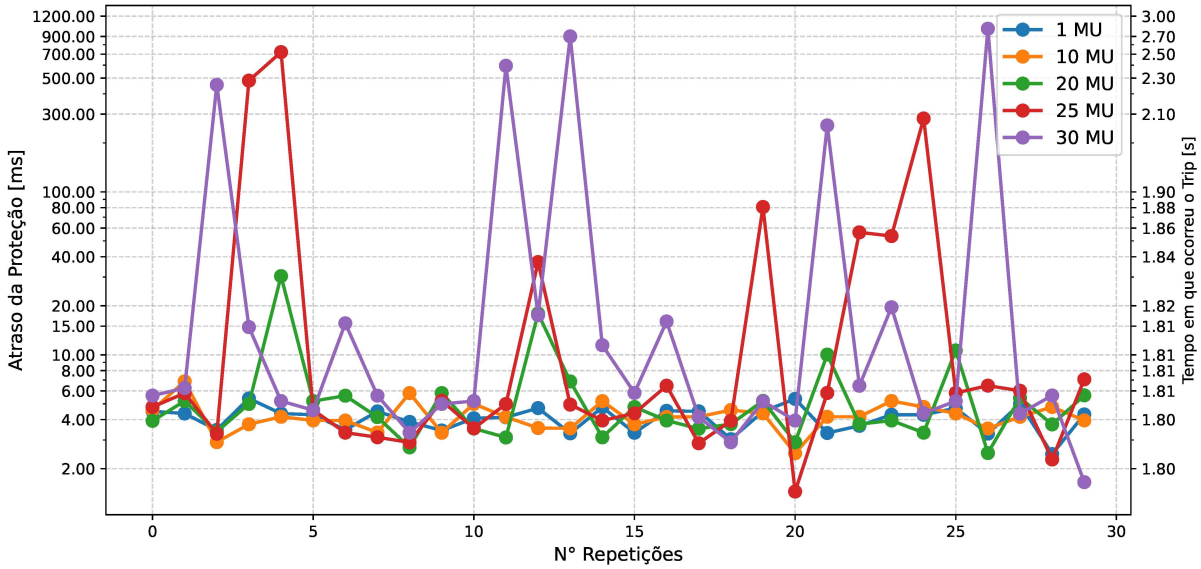
**PDIS** – Por último, foi verificado o tempo de atraso da proteção de distância, a sua parametrização é indicada na Tabela 4.3 e os resultados obtidos na Figura 4.3.

**Tabela 4.3** – Parametrização do teste da função PDIS

| vIED | | vMU | | |
|---|---|---|---|---|
| Parâmetro | Valor | Sequência | Valor | Tempo |
| Tipo | Adminitância (MHO) | Pre-Falta | 11,50 Ω | 1 s |
| Pickup | 15 Ω | Falta | 2,87 Ω | 3 s |
| Angle | 32° | Pos-Falta | 34,5 Ω | 2 s |
| Tempo de Atraso | 0,8 s | | | |

Com base na sua parametrização, o tempo de resposta teórico desta proteção é 1,80 s, considerando o tempo de 1,00 s da Pre-Falta e 0,80 s da atuação da função de proteção. Os resultados obtidos são mostrados no gráfico da Figura 4.3

**Figura 4.3** – Tempo de atraso na atuação da proteção PDIS em relação ao número de MUs na rede




**Fonte:** autoria própria

Os valores médios, máximo e mínimo dos Gráficos 4.1, 4.2 e 4.3 são apresentadas na Tabela 4.4.

**Tabela 4.4** – Resultados estatísticos a cerca do atraso da atuação das proteções

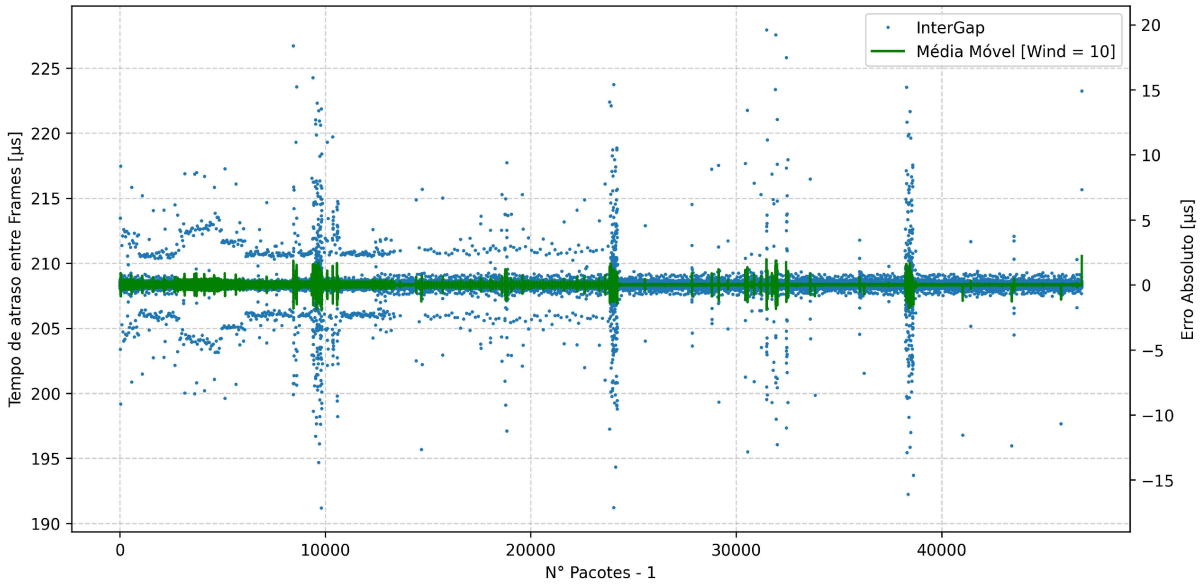
| Estatística | | Quantidade de MUs na rede | | | | |
|---|---|---|---|---|---|---|
| | | 1 MU | 10 MU | 20 MU | 25 MU | 30 MU |
| Médio | PIOC (ANSI 50) | 4,14 ms | 4,22 ms | 4,93 ms | 74,05 ms | 89,24 ms |
| | PTOC (ANSI 51) | 3,48 ms | 4,73 ms | 4,62 ms | 27,94 ms | 125,75 ms |
| | PDIS (ANSI 21) | 4,10 ms | 4,22 ms | 4,71 ms | 60,56 ms | 113,21 ms |
| Máximo | PIOC (ANSI 50) | 5,96 ms | 5,60 ms | 6,25 ms | 1078,33 ms | 1095,83 ms |
| | PTOC (ANSI 51) | 5,04 ms | 6,76 ms | 12,83 ms | 689,69 ms | 915,28 ms |
| | PDIS (ANSI 21) | 5,40 ms | 6,87 ms | 10,63 ms | 721,45 ms | 1004,37 ms |
| Mínimo | PIOC (ANSI 50) | 2,62 ms | 2,70 ms | 2,91 ms | 2,90 ms | 4,39 ms |
| | PTOC (ANSI 51) | 0,27 ms | 3,87 ms | 1,14 ms | 1,56 ms | 2,19 ms |
| | PDIS (ANSI 21) | 2,46 ms | 2,49 ms | 1,45 ms | 1,45 ms | 1,65 ms |

## 4.2 Resultados do Teste 2 - Avaliação da variação entre os tempos de envio da vMU

Após capturar os tráfego da rede por 10 segundos, foi realizada um filtragem obtendo apenas os pacotes *Sampled value*. E, a partir dos *frames* obtidos, foi calculado o

tempo de atraso entre o envio de cada pacote (intergap), indicado na Figura 4.4.

**Figura 4.4** – Variação do tempo entre frames




**Fonte:** autoria própria

Os resultados estatísticos dos dados apresentados no gráfico são mostrados na tabela abaixo.

**Tabela 4.5** – Resultados obtidos no ensaio da variação do envio de pacotes da vMU

| Parâmetro | Valor [$\mu s$] |
|---|---|
| Valor Médio | 208,33743 |
| Máximo | 227,93612 |
| Mínimo | 191,18404 |
| Desvio Padrão | 0,87542 |
| Variância | 0,76635 |

## 4.3 Discussões gerais

Em relação Teste 1, nos resultados apresentados na Tabela 4.4, quando o número de MU conectadas na rede foi igual ou inferior à 20, o vIED teve um tempo de resposta satisfatório, visto que o atraso na proteção não foi superior à 16 ms, ou seja, menor que um ciclo de onda (< 1 cycle). Apresentando resultados similares aos IED reais.

Porém, quando o número de Merging Units conectadas na rede foi maior ou igual à 25, houve um aumento significativo no tempo de atuação do vIED, chegando até cerca de 1100 ms de atraso. Entretanto, esta quantia de Merging Units é relativamente alta, pois a banda de transmissão que as 25 MU gastam é superior à 100 Mbi/s.

Portanto, considerando que o adaptador de rede não perdeu nenhum pacote, o atraso observado pode ser atribuído à necessidade tanto do vIED quanto da vMU de

realizar a decodificação dos pacotes presentes na rede. O vIED realiza esse processo para obter os valores de corrente e tensão do pacote *Sampled value*, enquanto a vMU o faz para detectar a atuação do IED decodificando o pacote GOOSE. Consequentemente, à medida que o fluxo de dados na rede aumenta, os equipamentos virtuais demandam mais tempo para concluir o processo de decodificação dos pacotes.

Além disso, ao examinar os gráficos apresentados nas Figuras 4.1, 4.2 e 4.3, observou-se a presença ocasional de pontos atípicos. Essa ocorrência pode ser atribuída tanto ao possível atraso na decodificação quanto à eventual perda de pacotes GOOSE pelo adaptador de rede. A perda de tais pacotes, que indicam a atuação da função de proteção, pode resultar em atrasos significativos em determinados momentos.

A seguir, no segundo Teste, Os resultados apresentados na Tabela 4.5 indicam uma baixa variação no tempo de envio entre os pacotes, visto que o valor médio foi próximo do teórico e o Desvio Padrão e a Variância foram baixos.

Além disso, o atraso observado pode ter sido gerado pelo próprio *driver* do adaptador de rede. Isto foi verificado ao medir o tempo no qual a função para enviar o pacote era invocada, revelando que esse intervalo era consistentemente de 1/4800 segundos, com precisão em nanossegundos. Esses resultados indicam que eventuais atrasos estavam relacionados à própria execução da função.

Além disso, a função de envio considera apenas o tempo inicial e o incrementa a cada pacote, o que implica que a ocorrência de atrasos ocasionais não se propaga para o envio dos pacotes subsequentes.

# 5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foram desenvolvidos dois equipamentos do sistema Elétrico de forma virtual: (i) um IED para proteção e automação e (ii) uma Merging Unit para envio de valores amostrados. Estes dispositivos tiveram o propósito de verificar e testar a arquitetura PAC centralizada/virtual.

Os equipamentos virtuais foram hospedados em um servidor laboratorial, onde foram realizados testes de desempenho e validação a cerca deles. Com base nos resultados, o IED virtual obteve tempos de atuação satisfatórios, menores que 1 ciclo de onda (i.e., 16ms). Além disso, foi possível verificar os limites de dados que a aplicação suporta. A Merging Unit Virtual também alcançou resultados que estão em conformidade com a norma IEC 61850, conseguindo enviar os pacotes Sampled Values com uma variação máxima de 17,15 µs considerando o ambiente virtual.

Ademais, este trabalho ilustra a implementação desses equipamentos virtualmente e apresenta resultados positivos em relação ao novo modelo virtual do sistema de proteção e controle.

Como perspectivas para trabalhos futuros, no que diz respeito ao IED Virtual, há a intenção de aumentar a quantidade de Merging Units na rede que ele suporta, implementar outras funções de proteção, adicionar o recebimento de mensagens GOOSE de outros IEDs e integrar o algoritmo do Servidor MMS no IED com o barramento de Estação. Em relação à Merging Unit Virtual, pretende-se implementar mais testes específicos para cada tipo de proteção e adicionar o protocolo de redundância PRP no envio dos frames SV.

## REFERÊNCIAS

AFTAB, M. A. et al. Iec 61850 based substation automation system: A survey. *International Journal of Electrical Power & Energy Systems*, v. 120, p. 106008, 9 2020. ISSN 01420615.

CIGRE, Study Committee B5.60. *Protection, Automation and Control Architectures with Functionality Independent of Hardware*. [S.l.], 2022.

DPDK. 2023. . Acessado em: 02/11/2023.

FERREIRA, R. D. F.; OLIVEIRA, R. S. de. Cloud iec 61850: Dds performance in virtualized environment with opendds. In: *2017 IEEE International Conference on Computer and Information Technology (CIT)*. [S.l.: s.n.], 2017. p. 231–236.

FERREIRA, R. D. F.; OLIVEIRA, R. S. de. Cloud iec 61850 a case study of a software defined protection, automation & control system. In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. [S.l.: s.n.], 2018. v. 1, p. 75–82.

FFTW. 2023. www.fftw.org. Acessado em: 02/11/2023.

FILHO, J. M.; MAMEDE, D. R. *Proteção de Sistemas Elétricos de Potência*. 2. ed. Rio de Janeiro: LTC, 2020. E-book.

HIGGINS, N. et al. Concept for intelligent distributed power system automation with iec 61850 and iec 61499. In: *2008 IEEE International Conference on Systems, Man and Cybernetics*. [S.l.: s.n.], 2008. p. 36–41.

HOROWITZ, S. H.; PHADKE, A. G. *Power System Relaying*. 4th. ed. [S.l.]: Wiley, 2014. ISBN 9781118662007.

IEC. *IEC 60255-3: Electrical Relays - Part 3: Single Input Energizing Quantity Measuring Relays with Dependent or Independent Time*. [S.l.], 1989.

IEC 61850-2. *Communication networks and systems for power utility automation - Part 2: Glossary*. [S.l.], 2003. v. 1, n. IEC 61850-2.

IEC 61850-8-1. *Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*. [S.l.], 2011. v. 2, n. IEC 61850-8-1.

IEC 61850-9-2. *Communication networks and systems for power utility automation - Part 9-2: Specific communication service mapping (SCSM) - Sampled values over ISO/IEC 8802-3*. [S.l.], 2011. v. 2, n. IEC 61850-9-2.

IEC 61869-9. *Instrument transformers – Part 9: Digital interface for instrument transformers*. [S.l.], 2016. v. 2, n. IEC 61869-9.

IEEE. *IEEE Std. C37.112: IEEE Standard Inverse Time Characteristics for Overcurrent Relay*. [S.l.], 1996.

IEEE 802.1Q. *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks*. [S.l.], 2018.

ISO/IEC 7498-1:1994. *Information technology Open Systems Interconnection Basic Reference Model: The Basic Model*. [S.l.], 1994.

JANSSEN, M. C.; APOSTOLOV, A. Iec 61850 impact on substation design. In: . [S.l.]: IEEE, 2008. p. 1–7. ISBN 978-1-4244-1903-6.

Linux KVM. 2023. linux-kvm.org. Acessado em: 02/11/2023.

Open vSwitch. 2023. . Acessado em: 02/11/2023.

ROSCH, D. et al. Virtualsubstation: An iec 61850 framework for a containernet based virtual substation. In: . [S.l.]: IEEE, 2022. p. 1–6. ISBN 978-1-6654-5505-3.

ROSCH, D.; NICOLAI, S.; BRETSCHNEIDER, P. Container-based virtualization of an iec 61850 substation co-simulation approach. In: . [S.l.]: IEEE, 2022. p. 1–6. ISBN 978-1-6654-6865-7.

SAMARA-RUBIO, D.; MCKENZIE, G.; KHAJURIA, P. *A Paradigm Shift in Power System Protection*. [S.l.], 2022.

VPAC Alliance. 2023. vpacalliance.com. Acessado em: 02/11/2023.

VYATKIN, V. et al. Towards intelligent smart grid devices with iec 61850 interoperability and iec 61499 open control architecture. In: *IEEE PES T&D 2010*. [S.l.: s.n.], 2010. p. 1–8.

WOJTOWICZ, R.; KOWALIK, R.; RASOLOMAMPIONONA, D. D. Next generation of power system protection automation—virtualization of protection systems. *IEEE Transactions on Power Delivery*, v. 33, p. 2002–2010, 8 2018. ISSN 0885-8977.