



Universidade Federal do Ceará – UFC  
Departamento de Engenharia Elétrica  
Programa de Pós-Graduação em Engenharia Elétrica

Lucas Silveira Melo

Desenvolvimento de uma Plataforma para Implementação de Sistemas  
Multiagentes com aplicação para Recomposição Automática de Sistemas de  
Distribuição de Energia Elétrica

Fortaleza  
2015

**Lucas Silveira Melo**

**Desenvolvimento de uma Plataforma para Implementação de Sistemas Multiagentes  
com aplicação para Recomposição Automática de Sistemas de Distribuição de Energia  
Elétrica**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Orientadora: Prof<sup>a</sup>. Ph.D. Ruth Pastôra Saraiva Leão

Co-Orientador: Prof. Dr. Giovanni Cordeiro Barroso

Fortaleza  
2015

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca de Pós-Graduação em Engenharia - BPGE

- 
- M485d      Melo, Lucas Silveira.  
Desenvolvimento de uma plataforma para implementação de sistemas multiagentes com aplicação para recomposição automática de sistemas de distribuição de energia elétrica / Lucas Silveira Melo. – 2015.  
120 f. : il. color., enc. ; 30 cm.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2015.  
Área de Concentração: Sistemas de Energia Elétrica.  
Orientação: Profa. Dra. Ruth Pastôra Saraiva Leão.  
Coorientação: Prof. Dr. Giovanni Cordeiro Barroso.
1. Engenharia elétrica. 2. Redes elétricas inteligentes. 3. Simulação por computador. I. Título.

LUCAS SILVEIRA MELO

DESENVOLVIMENTO DE UMA PLATAFORMA PARA IMPLEMENTAÇÃO  
DE SISTEMAS MULTIAGENTES COM APLICAÇÃO PARA  
RECOMPOSIÇÃO AUTOMÁTICA DE SISTEMAS DE  
DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito à obtenção do título de Mestre em Engenharia Elétrica. Área de Concentração: Sistema de Energia Elétrica.

Aprovada em: 14/08/2015

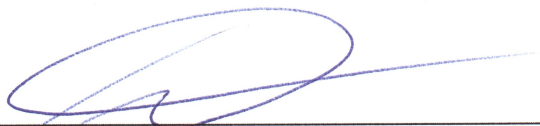
BANCA EXAMINADORA



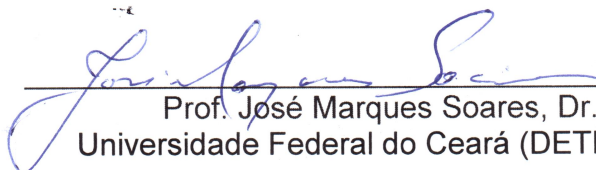
Prof<sup>a</sup>. Ruth Pastora Saraiva Leão, Dr<sup>a</sup>. (Orientador)  
Universidade Federal do Ceará (DEE-UFC)



Prof. Giovanni Cordeiro Barroso, Dr. (Coorientador)  
Universidade Federal do Ceará (DEE-UFC)



Prof. Diego Issicaba, Dr.  
Universidade Tecnológica Federal do Paraná (UTFPR)



Prof. José Marques Soares, Dr.  
Universidade Federal do Ceará (DETI-UFC)

---

# Agradecimentos

---

A Deus em primeiro lugar, por ser a fonte de tudo o que existe, por sua misericórdia e seu amor inestimável por cada um de seus filhos e por nos conceder o dom da vida.

À Nossa Senhora, santuário e repouso da Santíssima Trindade, onde Deus está mais magnífica e divinamente que em qualquer outro lugar do universo.

À minha mãe, Helena, por sua força, sua determinação e por tudo que me ensinou e passou para me dar uma educação sólida e cheia de amor.

À minha noiva, Allana, por ser a pessoa em quem me apoio sempre que tenho dificuldades, por seu amor, sua amizade e por ser a pessoa mais especial que já encontrei.

A meus professores, professora Ruth Leão, minha orientadora, por ser uma pessoa iluminada que nasceu para ensinar, por sua força e liderança, por suas palavras sempre sábias e encorajadoras e por sua atenção maternal com todos seus orientandos. Professor Raimundo Furtado, por seus conselhos, ensinamentos, dedicação e por ser mais que um professor, um amigo. Professor Giovanni Cordeiro, por seus conselhos e orientações por sua humildade, por estar sempre à disposição e por seu carisma e alegria.

Aos professores que ministraram disciplinas durante o Mestrado: Guilherme Barreto, Marques Soares, Arthur Plínio e José Carlos.

Aos meus companheiros de apartamento, Fernando e Amaury, por todos os momentos que passamos juntos, pelas conversas que vararam a madrugada, por compartilhar comigo e deixar que eu compartilhasse as incertezas e esperanças típicas dos recém formados.

Aos meus companheiros de pós-graduação Aluisio, Kathiane, Mário, Janaína, Caio, Silas, Domenico, Lucélia e Roberto pessoas admiráveis que tive o prazer de conviver e trabalhar. A todos os membros do Grupo de Redes Elétricas Inteligentes (GREI) da Universidade Federal do Ceará, em especial, Mateus, Mapse, Tales e Edmundo.

A todos os meus amigos e colegas, em especial aos colegas do GPEC. A todos os funcionários do Departamento de Engenharia Elétrica pela atenção, profissionalismo e dedicação. Ao CNPq pelo apoio financeiro.

*“Estou convencido de que nem a morte, nem a vida,,  
nem os anjos, nem os principados, nem o presente,  
nem o futuro, nem os poderes, nem as forças das alturas  
ou das profundidades, nem qualquer outra criatura,  
nada nos poderá separar do amor de Deus,  
manifestado em Jesus Cristo, Nosso Senhor”.*  
*(Bíblia Sagrada, Romanos 8, 38-39)*

# Resumo

MELO, L. S. “Desenvolvimento de uma Plataforma para Implementação de Sistemas Multiagentes: uma aplicação para Recomposição Automática de Sistemas de Distribuição de Energia Elétrica”. Universidade Federal do Ceará - UFC, 2015, 99p.

É comum a ocorrência de faltas permanentes no sistema de distribuição de energia elétrica. Por tratar-se de um sistema radial, a atuação da proteção para estas faltas causa a desenergização não só do setor em falta, mas de todos os consumidores a jusante do setor onde ocorreu a falta. Fazendo uso dos dispositivos de disjunção normalmente fechados ao longo do alimentador e normalmente abertos nas suas bordas é possível isolar o setor sob falta e re-energizar os setores sãos, reduzindo o número de consumidores afetados por um defeito. Este procedimento é normalmente feito pelos operadores da rede, e além de demandar um tempo considerável, está sujeito a erros por parte do operador. No sentido de tornar automática as análises de restauração da rede e prover o sistema da capacidade de auto-cura, têm sido propostas diversas metodologias para o problema. A maioria destas abordagens adota uma estratégia centralizada e não abordam o aspecto de auto-cura da rede elétrica. Neste trabalho, é proposta uma abordagem utilizando sistemas multiagentes para recomposição de setores de alimentadores de distribuição de energia elétrica. A técnica de sistemas multiagentes vem se mostrando bastante promissora no desenvolvimento de sistemas distribuídos em um contexto de redes elétricas inteligentes. Para que a recomposição ocorra sem a violação das restrições operacionais e de forma coerente, são feitas análises pelos agentes alimentadores antes que qualquer comando seja enviado para as chaves do sistema por meio de agentes dispositivo. O sistema multiagente proposto é implementado em uma plataforma de desenvolvimento de agentes proposta neste trabalho e que utiliza a linguagem de programação Python. A plataforma tem o nome de PADE, *Python Agent DEvelopment framework*. A representação computacional sem simplificações da rede é proporcionada por uma codificação de dados apoiada na teoria de grafos e denominada Representação Nó-Profundidade, que serve de base para o desenvolvimento de uma API de representação da rede-elétrica que modela cada um dos componentes necessários nas análises de recomposição. Ao agente dispositivo é dada a possibilidade de comunicação com IED, que controlam as chaves do sistema, por meio dos protocolos da norma IEC 61850: GOOSE e MMS. Para validar a metodologia proposta são realizadas simulações computacionais utilizando uma rede de distribuição simples como caso base e uma plataforma de testes com: mala de testes de relés, IED de proteção e controle de mercado, switch gerenciável e placas de desenvolvimento de sistemas embarcados.

**Palavras-chaves:** Multiagentes. Recomposição automática. Redes elétricas inteligentes.

# Abstract

MELO, L. S. “Development of a Platform for Implementing Multi-agents Systems : a application to Automatic Restoration of Electric Power Distribution Systems”. Universidade Federal do Ceará - UFC, 2015, 99p.

It is common the occurrence of permanent faults in power distribution systems. In a typical radial power distribution system when the fault protection system operates, may cause power-off not only in the fault section, but also to all customers downstream the fault. Through disjunction devices normally closed along the feeder, and normally open on its edges, is possible to isolate the faulty sector and reenergize the healthy ones, reducing the number of customers affected by a fault. Network operators normally do this procedure manually and in addition to demand a considerable amount of time, is subject to errors on the part of the operator. In order to automate the analysis of the network and provided it of self-healing capacity, various methods have been proposed to solve this matter. Most of these approaches adopts a centralized strategy and do not address the aspect of electric power grid self-healing. In this work is proposed an approach that uses multi-agent systems for self-healing purposes of power distribution systems. Multi-agent are highly suitable for modelling distributed systems in the smart grid domain. For a safe recovery and without violation of operational restrictions the feeder agents perform an evaluation before device agents send any command to the network switches. The proposed multi-agent system is implemented in a agent's development platform proposed in this work that uses the Python programming language. The platform is called PADE, Python Agent DEvelopment framework. The computer representation of the network, without simplifications, is accomplished by a data encoding based on the theory of graphs and named node-depth representation that serves as a basis for the development of an API of network representation that models each of the required components in the restoration analysis. The device agents communicate with IED that in turn control the switches in the network, by means of IEC 61850 protocols: GOOSE and MMS. To validate the proposed approach, computer simulations are performed using a simplified distribution power grid as a case study and a test platform with relay test case, protection and control IED, managed switch and embedded systems.

**Key-words:** Multi-agents. Self-healing. Smart grids.



---

# Lista de ilustrações

---

|   |    |
|---|----|
| Figura 1.1 – Rede elétrica com setor A sob ocorrência de falta. . . . .   | 3  |
| Figura 1.2 – Rede elétrica com atuação da proteção no religador 1. . . . .  | 3  |
| Figura 1.3 – Rede elétrica com setor faltoso isolado do sistema. . . . .  | 4  |
| Figura 1.4 – Rede elétrica com setores B, C e F recompostos. . . . .  | 4  |
| Figura 1.5 – Tecnologias empregadas em uma Rede Elétrica Inteligente em diferentes níveis do SEP. . . . .   | 6  |
| Figura 1.6 – Placas de hardware embarcado Raspberry Pi B e Beaglebone Black. . .  | 10 |
| Figura 1.7 – Bancada experimental de testes de validação. . . . .   | 11 |
| Figura 1.8 – Comunicação entre os agentes BAG e FAG. . . . .  | 12 |
| Figura 1.9 – Comunicação entre Feeder agents e Loads. . . . .   | 13 |
| Figura 1.10 – Representação esquemática dos quatro passos da restauração definida por Solanki, Khushalani e Schulz (2007). . . . .                      | 14 |
| Figura 1.11 – Agentes definidos em Melo (2013). . . . .   | 15 |
| Figura 1.12 – Agentes definidos em Barros (2013). . . . .   | 15 |
| Figura 2.1 – Representação esquemática de um agente interagindo com seu ambiente.   | 20 |
| Figura 2.2 – Diagrama de classes representando associações entre diferentes classes do padrão CIM. . . . .  | 22 |
| Figura 2.3 – Representação em XML da classe Breaker definida no padrão CIM, como exemplo de estrutura ontológica. . . . .                               | 23 |
| Figura 2.4 – Exemplo de mensagem no padrão FIPA-ACL. . . . .  | 25 |
| Figura 2.5 – Exemplo de mensagem no padrão FIPA-ACL em formato XML . . . .  | 26 |
| Figura 2.6 – Elementos constituintes de uma plataforma multiagente. . . . .   | 27 |
| Figura 2.7 – Troca de mensagens entre agentes que implementam o protocolo FIPA ContractNet. . . . .   | 28 |
| Figura 3.1 – Gráficos de quantidade de buscas por materiais introdutórios às linguagens de programação mais conhecidas no Google entre 2005 e 2015. . . | 31 |

|  |    |
|--|----|
| Figura 3.2 – Paradigmas de programação . . . . .   | 34 |
| Figura 3.3 – <i>Loop</i> de eventos do Twisted e chamada de uma <i>callback</i> . . . . .  | 34 |
| Figura 3.4 – Estrutura de arquivos do PADE. . . . .  | 38 |
| Figura 3.5 – Estrutura de classes do framework PADE no padrão UML. . . . .   | 39 |
| Figura 3.6 – Diagramas atividades do processo de identificação no PADE. . . . .  | 40 |
| Figura 3.7 – Registro e atualização de tabela de agentes. . . . .  | 41 |
| Figura 3.8 – Interface gráfica do agente Sniffer do PADE. . . . .  | 42 |
| Figura 3.9 – Diagrama de classes UML de um agente que tem um comportamento<br>FIPA ContractNet associado. . . . .                          | 43 |
| Figura 3.10–Mensagens exibidas pelos agentes chave e Operador. . . . .   | 46 |
| Figura 4.1 – Estados de operação do sistema . . . . .  | 49 |
| Figura 4.2 – Grafo simples e sua representação matricial em RNP. . . . .   | 52 |
| Figura 4.3 – Operação de poda em um grafo. . . . .   | 53 |
| Figura 4.4 – Operação de poda em um grafo. . . . .   | 54 |
| Figura 4.5 – Exemplo de representação por grafo dos elementos que compõem uma<br>rede elétrica teste. . . . .                              | 54 |
| Figura 4.6 – Três possíveis configurações de RNP de setor. . . . .   | 55 |
| Figura 4.7 – Camadas da API de representação da rede elétrica. . . . .   | 56 |
| Figura 4.8 – Tela do terminal de comandos IPython com utilização de objetos árvore<br>e alguns de seus atributos . . . . .                 | 58 |
| Figura 4.9 – Diagrama UML das classes desenvolvidas para a API. . . . .  | 59 |
| Figura 4.10–Tela do terminal de comandos IPython com utilização dos comandos<br>poda e inserção de ramos nos alimentadores. . . . .        | 60 |
| Figura 4.11–Processo de conversão XML para objetos dinâmicos da API em Python. . . . .   | 61 |
| Figura 4.12–Diagrama de atividades do SMRA. . . . .  | 62 |
| Figura 4.13–Análise de propostas pelo AA. . . . .  | 65 |
| Figura 4.14–Análise de inserção pelo AA. . . . .   | 66 |
| Figura 4.15–Diagrama de atividades desenvolvidas pelo AD. . . . .  | 68 |
| Figura 4.16–Representação gráfica dos processos de comunicação entre IED e<br>agentes que compõem o SMRA. . . . .                          | 69 |
| Figura 5.1 – Interface de virtualização proposta pela norma IEC 61850. . . . .   | 73 |
| Figura 5.2 – Mapeamento da ACSI nas camadas dos protocolos de rede de computa-<br>dores. . . . .   | 74 |
| Figura 5.3 – Modelo de dados da ACSI descritos no módulo 7 da norma IEC 61850. . . . .   | 76 |
| Figura 5.4 – Modelo de referência de uma informação no padrão IEC 61850. . . . .   | 76 |
| Figura 5.5 – Possíveis formas de envio e recebimento de informação com IEC 61850. . . . .  | 77 |
| Figura 5.6 – Gráfico de tempo por mensagens GOOSE transmitidas após a modifica-<br>ção de um dos parâmetros do Data Set associado. . . . . | 80 |

|  |     |
|--|-----|
| Figura 5.7 – Modelo de comunicação da norma IEC 61850. . . . .   | 81  |
| Figura 5.8 – Processo de conversão da biblioteca libiec61850 em C para módulo Python. . . . .            | 83  |
| Figura 5.9 – Pilha de implementação da biblioteca libiec61850 . . . . .                                  | 83  |
| Figura 6.1 – Rede elétrica de distribuição considerada para realização dos testes. . . . .               | 86  |
| Figura 6.2 – Topologia física de testes do SMRA. . . . .   | 88  |
| Figura 6.3 – Atuação da proteção sobre a chave 1. . . . .  | 92  |
| Figura 6.4 – Mensagens de proposta recebidas pelo agente S1_AL1. . . . .                                 | 92  |
| Figura 6.5 – Topologia da rede após a inserção do setor B no alimentador S1-AL2. . . . .                 | 93  |
| Figura 6.6 – Proposta enviada pelo agente S2_AL1. . . . .  | 94  |
| Figura 6.7 – Topologia da rede após a inserção do ramo C-F no alimentador S2-AL1. . . . .                | 94  |
| Figura 6.8 – Topologia da rede após a poda do setor F do alimentador S2-AL1. . . . .                     | 95  |
| Figura 6.9 – Diagrama de comunicação entre agentes. . . . .  | 97  |
| Figura B.1 – Diagrama de sequência do cálculo do fluxo de carga de varredura direta-<br>inversa. . . . . | 114 |
| Figura B.2 – . . . . .   | 114 |
| Figura D.1 – Rede elétrica utilizada na realização de casos teste. . . . .                               | 118 |
| Figura D.2 – Análise de carregamento dos trechos do Alimentador S1_AL2. . . . .                          | 119 |
| Figura D.3 – Análise de carregamento dos trechos do alimentador S1_AL1. . . . .                          | 119 |
| Figura D.4 – Análise de carregamento dos trechos do alimentador S2_AL1. . . . .                          | 119 |
| Figura D.5 – Análise de queda de tensão nos nós de carga do alimentador S1_AL2 . . . . .                 | 120 |
| Figura D.6 – Análise de queda de tensão nos nós de carga do alimentador S1_AL1 . . . . .                 | 120 |
| Figura D.7 – Análise de queda de tensão nos nós de carga do alimentador S2_AL1 . . . . .                 | 120 |
| Figura D.8 – Análise de carregamento dos transformadores da subestação S1. . . . .                       | 121 |
| Figura D.9 – Análise de carregamento dos transformadores da subestação S2. . . . .                       | 121 |

---

## Lista de tabelas

---

|  |    |
|--|----|
| Tabela 1.1 – Trabalhos correlatos de desenvolvimento de sistemas de restauração automática. . . . .                              | 16 |
| Tabela 2.1 – <i>Performativas</i> definidas pela FIPA. . . . .   | 24 |
| Tabela 2.2 – Campos de Mensagem Padronizados. . . . .  | 25 |
| Tabela 4.1 – Comportamentos do agente Alimentador. . . . .   | 64 |
| Tabela 4.2 – Comportamentos do Agente Dispositivo . . . . .  | 67 |
| Tabela 5.1 – Estrutura da norma IEC 61850 . . . . .  | 72 |
| Tabela 6.1 – Distribuição de consumidores por setor na rede teste. . . . .   | 87 |
| Tabela 6.2 – Dados de carregamento e de queda de tensão da rede teste em condições normais. . . . .                              | 87 |
| Tabela 6.3 – Dispositivos conectados em rede Ethernet para execução dos testes. . . . .  | 89 |
| Tabela 6.4 – Agentes presentes na plataforma de testes do SMRA. . . . .  | 89 |
| Tabela 6.5 – Dados de carregamento e de queda de tensão da rede teste após a inserção do setor B no alimentador S1-AL2. . . . .  | 93 |
| Tabela 6.6 – Dados de carregamento e de queda de tensão da rede teste após a inserção do ramo C-F no alimentador S2-AL1. . . . . | 95 |
| Tabela 6.7 – Dados de carregamento e de queda de tensão da rede após a poda do setor F do alimentador S2-AL1. . . . .            | 95 |

---

# Lista de Algoritmos

---

|     |   |    |
|-----|---|----|
| 3.1 | Código de servidor de eco em Python utilizando Twisted . . . . .  | 36 |
| 3.2 | Código de cliente em Python utilizando Twisted . . . . .  | 36 |
| 3.3 | Código para execução em terminal linux dos algoritmos de servidor e cliente.  | 37 |
| 3.4 | Definição de dois agentes: Agente Chave e Agente Operador com seus respectivos comportamentos e código de inicialização . . . . . | 44 |
| 4.1 | Código com a instanciação dos objetos que representam a RNP das árvores apresentadas na Figura 4.5 . . . . .                      | 57 |

---

## Lista de Siglas e Abreviaturas

---

|      |  |
|------|--|
| AA   | Agente Alimentador                       |
| ACL  | Agent Communication Language             |
| ACSI | Abstract Communication Service Interface |
| AD   | Agente Dispositivo                       |
| AI   | Agente Inteligente                       |
| AID  | Agent IDentifier                         |
| AMS  | Agent Management System                  |
| API  | Interface de Programação para Aplicações |
| AS   | Agente Subestação                        |
| BAG  | Bus Agent                                |
| CAO  | Change Ancestor Operator                 |
| CDC  | Common Data Class                        |
| CFP  | Call For Proposes                        |
| CID  | Configured IED Description               |
| CIM  | Common Information Model                 |
| COS  | Centro de Operação do Sistema            |
| DA   | Data Attribute                           |
| DF   | Directory Facilitator                    |

|       |   |
|-------|---|
| DNP3  | Distributed Network Protocol                      |
| DS    | DataSet   |
| EPRI  | Electric Power Research Institute                 |
| FA    | Feeder Agent                                      |
| FAG   | Feeder Agent                                      |
| FIPA  | Foundation for Intelligent Physical Agents        |
| GOOSE | Generic Object Oriented Substation Event          |
| GREI  | Grupo de Redes Eléctricas Inteligentes            |
| GS    | Group Settings                                    |
| IEC   | International Electrotechnical Commission         |
| IED   | Dispositivo Eletrônico Inteligente                |
| IEEE  | Institute of Electrical and Electronics Engineers |
| JADE  | Java Agent DEvelopment framework                  |
| LA    | Load Agent  |
| LD    | Logical Device                                    |
| LN    | Logical Node                                      |
| MMS   | Manufacturing Message Specification               |
| MTS   | Message Transport System                          |
| NA    | Normalmente Aberto                                |
| NF    | Normalmente Fechado                               |
| PADE  | Python Agent DEvelopment framework                |
| PAO   | Preserve Ancestor Operator                        |
| PD    | Physical Device                                   |
| PIP   | Python Package Installer                          |
| REI   | Redes Eléctricas Inteligentes                     |
| RNP   | Representação Nó-Profundidade                     |

|       |  |
|-------|--|
| SCADA | Supervisory Control and Data Acquisition       |
| SCL   | Substation Communication Language              |
| SED   | Sistema Elétrico de Distribuição               |
| SEL   | Schweitzer Engineering Laboratories            |
| SEP   | Sistemas Elétricos de Potência                 |
| SMA   | Sistemas Multiagentes                          |
| SMRA  | Sistema Multiagente de Recomposição Automática |
| SV    | Sampled Values                                 |
| TC    | Transformador de Corrente                      |
| UCA   | Utility Communication Architecture             |
| UML   | Unified Modeling Language                      |
| XML   | Extensible Markup Language                     |



# Sumário

---

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>1</b>  |
| 1.1      | Motivação  | 1         |
| 1.2      | Justificativa e Objetivos                                      | 7         |
| 1.3      | Metodologia  | 9         |
| 1.4      | Revisão Bibliográfica  | 11        |
| 1.5      | Estrutura do Trabalho  | 17        |
| <b>2</b> | <b>Sistemas Multiagentes</b>                                   | <b>19</b> |
| 2.1      | Introdução   | 19        |
| 2.2      | Agentes Inteligentes   | 19        |
| 2.2.1    | Definições e Características de AI                             | 19        |
| 2.2.2    | Comunicação entre AI   | 21        |
| 2.2.3    | Linguagem de Ontologia   | 21        |
| 2.3      | Comunicação entre Agentes                                      | 22        |
| 2.3.1    | FIPA-ACL   | 23        |
| 2.4      | Plataforma Multiagente   | 25        |
| 2.5      | Protocolos de Comunicação Definidos pela FIPA                  | 27        |
| 2.6      | Conclusão  | 29        |
| <b>3</b> | <b>PADE: Plataforma para Desenvolvimento e Execução de SMA</b> | <b>30</b> |
| 3.1      | Introdução   | 30        |
| 3.2      | A escolha da Linguagem de Programação Python                   | 30        |
| 3.3      | Desenvolvimento da Plataforma                                  | 32        |
| 3.3.1    | O Framework Twisted  | 32        |
| 3.3.1.1  | Programação Assíncrona Orientada a Eventos                     | 33        |
| 3.3.1.2  | Componentes Básicos do Twisted                                 | 35        |
| 3.3.1.3  | Exemplos de Execução de Aplicações com Twisted                 | 35        |
| 3.3.2    | O Framework PADE   | 35        |
| 3.3.2.1  | Estrutura de Arquivos do PADE                                  | 36        |
| 3.3.2.2  | Filosofia de Execução dos Agentes no Framework PADE            | 37        |
| 3.3.2.3  | O Agente AMS   | 38        |
| 3.3.2.4  | O Agente Sniffer   | 41        |
| 3.3.2.5  | Estrutura de um Agente Modelado no PADE                        | 42        |
| 3.4      | Exemplo de Implementação de Agentes no PADE                    | 43        |
| 3.5      | Conclusão  | 45        |
| <b>4</b> | <b>Sistema de Recomposição Automático</b>                      | <b>47</b> |

|          |   |            |
|----------|---|------------|
| 4.1      | O problema da Restauração de SED . . . . .  | 48         |
| 4.1.1    | Descrição do Problema . . . . .   | 48         |
| 4.1.2    | Classificação do Problema . . . . .   | 49         |
| 4.1.3    | Descrição Matemática de Recomposição de um SED . . . . .  | 50         |
| 4.1.4    | Estrutura de Dados RNP . . . . .  | 52         |
| 4.2      | Interface de Programação de Aplicações para Representação da Rede Elétrica                              | 55         |
| 4.2.1    | XML para Entrada de Dados . . . . .   | 59         |
| 4.3      | Estrutura de SMA Proposta . . . . .   | 60         |
| 4.3.1    | Agente Alimentador . . . . .  | 62         |
| 4.3.1.1  | Descrição dos Comportamentos do AA . . . . .  | 63         |
| 4.3.2    | Agente Dispositivo . . . . .  | 67         |
| 4.4      | Comunicação entre Agentes e IED . . . . .   | 68         |
| 4.5      | Conclusão . . . . .   | 68         |
| <b>5</b> | <b>A Norma IEC 61850: Redes de Comunicação e Sistemas em Subestações</b>                                | <b>70</b>  |
| 5.1      | O surgimento da Norma IEC 61850 . . . . .   | 70         |
| 5.2      | Estrutura da Norma IEC 61850 . . . . .  | 72         |
| 5.3      | Modelo de Dados da IEC 61850 . . . . .  | 74         |
| 5.3.1    | Modelo de Representação da Informação . . . . .   | 74         |
| 5.3.2    | Modelo de Troca de Informações . . . . .  | 77         |
| 5.4      | Desenvolvimento de uma Biblioteca para Comunicação Via Protocolos de<br>Comunicação IEC 61850 . . . . . | 80         |
| 5.4.1    | Biblioteca libiec61850 . . . . .  | 81         |
| 5.4.2    | API IEC 61850 em Python . . . . .   | 82         |
| 5.5      | Conclusão . . . . .   | 83         |
| <b>6</b> | <b>Testes e Simulações . . . . .</b>  | <b>85</b>  |
| 6.1      | Descrição da Rede Elétrica Utilizada nos casos de Teste . . . . .                                       | 85         |
| 6.2      | Montagem de uma Plataforma para Realização de Testes . . . . .  | 87         |
| 6.3      | Descrição do Processo de Recomposição Automática . . . . .  | 90         |
| 6.4      | Caso Teste . . . . .  | 91         |
| 6.5      | Resultados da Recomposição . . . . .  | 96         |
| <b>7</b> | <b>Conclusões, Contribuições e Trabalhos Futuros . . . . .</b>  | <b>98</b>  |
| 7.1      | Conclusões . . . . .  | 98         |
| 7.2      | Contribuições . . . . .   | 99         |
| 7.3      | Trabalhos Futuros . . . . .   | 100        |
| 7.4      | Trabalho Publicado . . . . .  | 101        |
|          | <b>Referências . . . . .</b>  | <b>102</b> |
|          | <b>Apêndices . . . . .</b>  | <b>107</b> |
|          | <b>APÊNDICE A XML de representação da rede elétrica . . . . .</b>                                       | <b>108</b> |

|                   |  |            |
|-------------------|--|------------|
| <b>APÊNDICE B</b> | <b>Algoritmo de fluxo de carga . . . . .</b>                                   | <b>113</b> |
| <b>APÊNDICE C</b> | <b>Algoritmo de descoberta de dados de um servidor IEC<br/>61850 . . . . .</b> | <b>116</b> |
| <b>APÊNDICE D</b> | <b>Análises da rede teste . . . . .</b>  | <b>118</b> |
|                   | D.1 Análise de carregamento dos trechos . . . . .                              | 118        |
|                   | D.2 Análise de queda de tensão dos nós de carga . . . . .                      | 120        |
|                   | D.3 Análise de carregamento dos transformadores das subestações . . . . .      | 121        |

---

# Capítulo 1

## Introdução

---

O grande avanço, nas últimas décadas, de processadores digitais de dados, de tecnologias de informação e comunicação, bem como de instrumentação digital, imputaram significativas melhorias aos Sistemas Elétricos de Potência (SEP). Muitos dos recursos a exemplo dos relés digitais e dos sistemas supervisórios SCADA (*supervisory control and data acquisition*) tipificam a evolução no processo de medição, supervisão, controle, proteção, automação e comunicação do SEP.

Os sistemas SCADA aquisitionam dados como medições operacionais e estado de equipamentos, além de serem utilizados para enviar comandos de controle, comunicando com os dispositivos físicos via rede de dados (MOMOH, 2012).

Os relés de proteção digitais são Dispositivos Eletrônicos Inteligentes (IED) que utilizam técnicas avançadas de processamento de sinais e algoritmos inteligentes. Além disso, possuem a capacidade de se comunicar por meio de protocolos de comunicação em rede, aumentando sua eficiência e aplicabilidade (KIMURA et al., 2008).

### 1.1 Motivação

De acordo com Solanki, Khushalani e Schulz (2007), apesar da modernização dos IED e de outros dispositivos presentes no SEP, visando o aumento da confiabilidade e eficiência do sistema, a ocorrência de defeitos de natureza diversa na rede elétrica é inevitável.

A rede elétrica de distribuição é parte vulnerável do SEP, sendo constantemente afetada por defeitos (XU; CHOW; TAYLOR, 2006). Isso decorre do fato de que grande parcela da rede está exposta às intempéries assim como a característica da rede de ser muito ramificada, cobrindo grandes extensões.

De acordo com Xu, Chow e Taylor (2006), a grande maioria das faltas de energia são temporárias, sendo restabelecido o fornecimento de energia normalmente por meio de religadores automáticos que realizam tentativas de restabelecimento da rede dentro de tempos programados. No entanto, a ocorrência de faltas permanentes na rede elétrica também é recorrente e causa inúmeros transtornos, tanto aos consumidores, quanto às concessionárias de energia.

Para tentar minimizar o problema de desligamento de clientes devido a ocorrência de falta permanente em apenas um segmento da rede elétrica é comum que os alimentadores de distribuição de energia elétrica sejam radiais e seccionados em blocos de carga, neste trabalho denominados setores, de acordo com a codificação apresentada em Marques (2013), por meio de dispositivos com capacidade de seccionamento normalmente fechados (NF), que podem ser chaves (manuais ou automáticas), disjuntores ou religadores. Neste trabalho, com o intuito de simplificar a representação da rede, todos os dispositivos de seccionamento serão considerados como religadores.

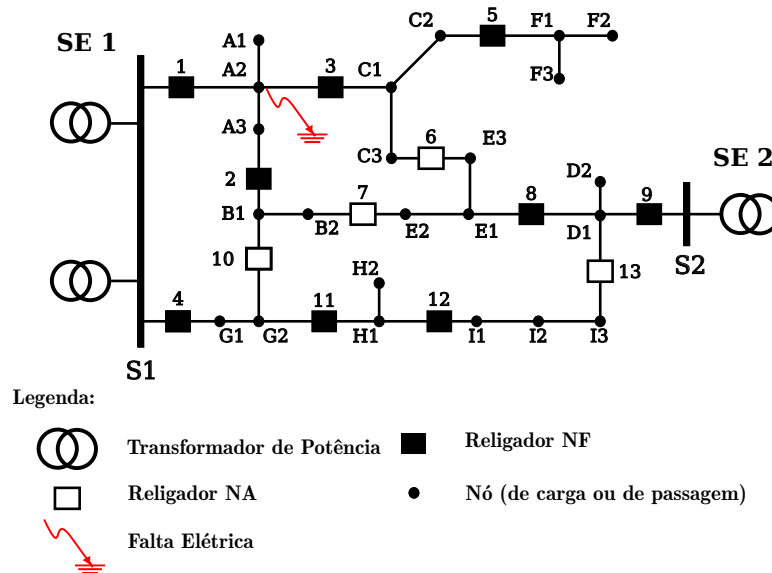
Utilizando esta prática, é possível isolar do sistema porções defeituosas da rede através da abertura dos religadores NF. Embora seja uma solução muito utilizada, verifica-se que, por exemplo, quando um religador atua pela proteção, ele isola não apenas o setor defeituoso, mas todos os setores jusantes à falta. A solução para esse problema é a instalação de religadores normalmente abertos (NA) nas bordas do alimentador, de forma que os setores são jusantes ao setor faltoso possam ter seu fornecimento restabelecido por outros alimentadores através do fechamento de religadores NA, dando surgimento aos sistemas radiais com recurso.

Na Figura 1.1 é mostrada uma rede elétrica que servirá como exemplo para demonstrar como a recomposição pode ocorrer. Na rede é possível observar a existência de duas subestações identificadas como SE 1 e SE 2. A subestação SE 1 possui dois alimentadores radiais com recurso. Um dos alimentadores de SE 1 contém os setores A, B, C e F, enquanto o outro é constituído pelos setores G, H e I. Já a subestação SE 2 possui apenas um alimentador que por sua vez é composto pelos setores D e E. Também é possível observar a existência de pontos em cada um dos setores, identificados pela letra que determina o setor e por um número complementar, estes pontos são chamados nós de carga ou de passagem. Quando os pontos representam nós de carga, estes nós caracterizam cargas presentes na rede, no caso de uma rede de média tensão de 13,8 kV, os pontos representam cada um dos transformadores 13,8 kV:380 V, dessa forma é possível representar a rede elétrica de maneira fiel, sem recorrer a concentração de cargas em um único ponto. Quando os pontos representam nós de passagem significam apenas uma bifurcação da rede e podem ser vistos como nós de carga com carga zero associada (MARQUES, 2013).

Ainda na Figura 1.1 é mostrada a ocorrência de uma falta no setor A. Já na Figura 1.2 é mostrada a retirada da falta do sistema pela atuação da proteção do religador 1, desenergizando os setores A, B, C e F. Já na Figura 1.3 é mostrado o isolamento do setor

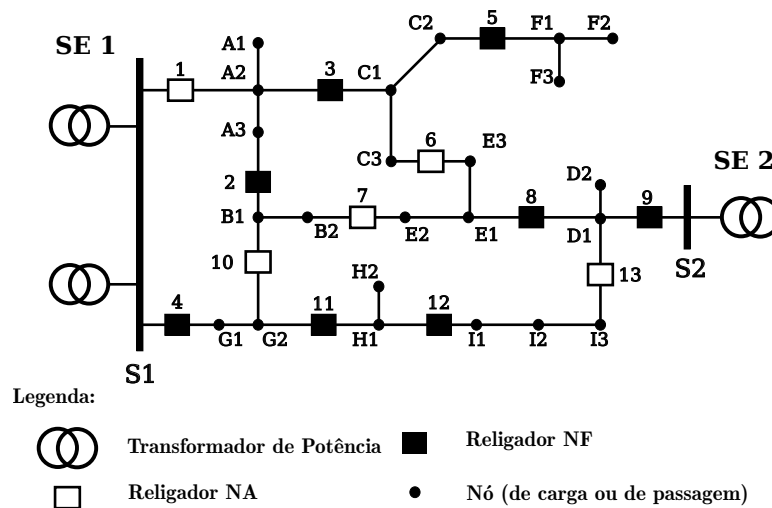
em falta A, por meio da abertura dos religadores 2 e 3. Finalmente na Figura 1.4 os setores desenergizados que não possuem defeito, B, C e F, são reenergizados por meio do fechamento dos religadores 10 e 6 respectivamente.

Figura 1.1 – Rede elétrica com setor A sob ocorrência de falta.



Fonte: Própria do autor.

Figura 1.2 – Rede elétrica com atuação da proteção no religador 1.

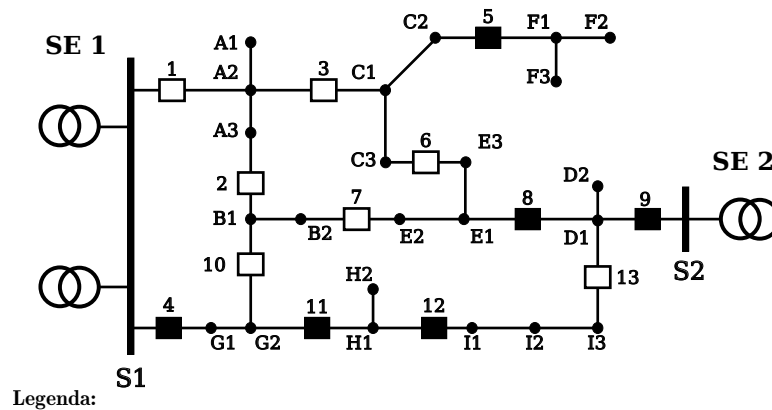


Fonte: Própria do autor.

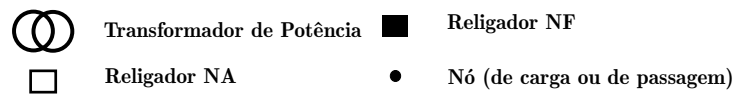
No processo de reenergização dos setores desenergizados, diversos aspectos devem ser observados para que a recomposição seja efetuada de maneira satisfatória e sem prejudicar nenhum equipamento do SEP, causando prejuízos ainda maiores. Segundo Zidan, El-Saadany e Chaar (2011), as seguintes restrições devem ser observadas durante o processo de recomposição da rede elétrica:

- Ausência de sobrecarga nos transformadores da subestação recompositora;

Figura 1.3 – Rede elétrica com setor faltoso isolado do sistema.

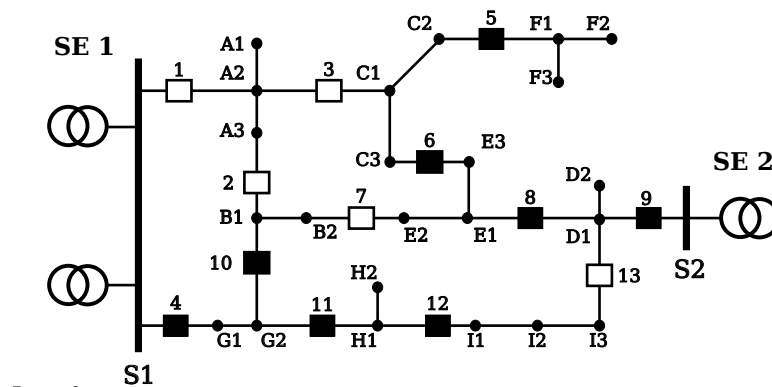


Legenda:

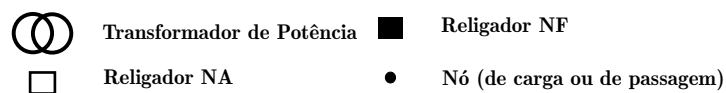


Fonte: Própria do autor.

Figura 1.4 – Rede elétrica com setores B, C e F recompostos.



Legenda:



Fonte: Própria do autor.

- Ausência de sobrecarga nos condutores da rede elétrica;
- Manutenção de níveis adequados de tensão de acordo com o exigido pela legislação;
- Manutenção da radialidade dos alimentadores;

Na tomada de decisão, a recomposição do suprimento pode ser baseada na potência ou no número de consumidores. Neste trabalho adotou-se a segunda alternativa, a de *maximizar* o número de consumidores com fornecimento de energia restabelecido e que esse objetivo seja alcançado com o menor impacto possível na rede, ou seja com um número *mínimo* de manobras nos dispositivos de chaveamento.

Outra variável em um processo de recomposição é a presença de cargas prioritárias, como por exemplo, grandes indústrias, hospitais ou até mesmo dependentes de equipamentos elétricos para sobrevivência.

Dessa forma o processo de recomposição da rede elétrica, mediante condição de contingência, deve ser tratado como um problema de otimização, com múltiplos objetivos, por vezes conflitantes entre si e com restrições (MARQUES et al., 2014).

Hoje a grande maioria das concessionárias conta com ferramentas computacionais que auxiliam no processo de tomada de decisão para a reconfiguração da rede e sua consequente reenergização após a ocorrência de uma falta, mas que não têm como objetivo final a restauração automática do sistema e por isso atuam apenas como ferramentas que facilitam a análise de restauração da rede por parte do operador. Por vezes o problema na rede elétrica é identificado através de ligações telefônicas de consumidores desenergizados e a recomposição é realizada por uma equipe em campo, realizando manobras em chaves manuais, seguindo orientações do centro de operações do sistema (COS) (MELO, 2013).

Esse processo de recomposição manual da rede elétrica está sujeito à ocorrência de falhas, principalmente por se tratar de um processo manual, de grande estresse para o operador e extremamente dependente de sua experiência e de seu conhecimento da rede, além de demandar uma grande quantidade de tempo, o que por sua vez afeta negativamente os indicadores de qualidade de fornecimento de energia da concessionária e degrada sua imagem perante os consumidores.

A solução de restauração manual da rede elétrica não está em harmonia com o conceito de redes elétricas inteligentes (REI), uma tendência crescente para as redes elétricas de todo o mundo.

As REI são definidas em CGEE (2012) como:

‘Aquela que utiliza tecnologia digital avançada para monitorar e gerenciar o transporte de eletricidade em tempo real com fluxo de energia e de informações bidirecionais entre o sistema de fornecimento de energia e o cliente final, integrando e possibilitando ações por todos os usuários a ela conectados, de modo a fornecer eficientemente uma energia sustentável, econômica e segura. Além de proporcionar ao consumidor informações sobre o seu consumo, tarifa e qualidade do serviço e do produto recebido em tempo real’.

Ainda para (LOPES et al., 2012) “Smart grid é o termo geralmente usado para descrever a integração dos elementos da rede elétrica com uma infraestrutura de informação, oferecendo inúmeros benefícios tanto para os geradores e distribuidores como para os consumidores de eletricidade”.

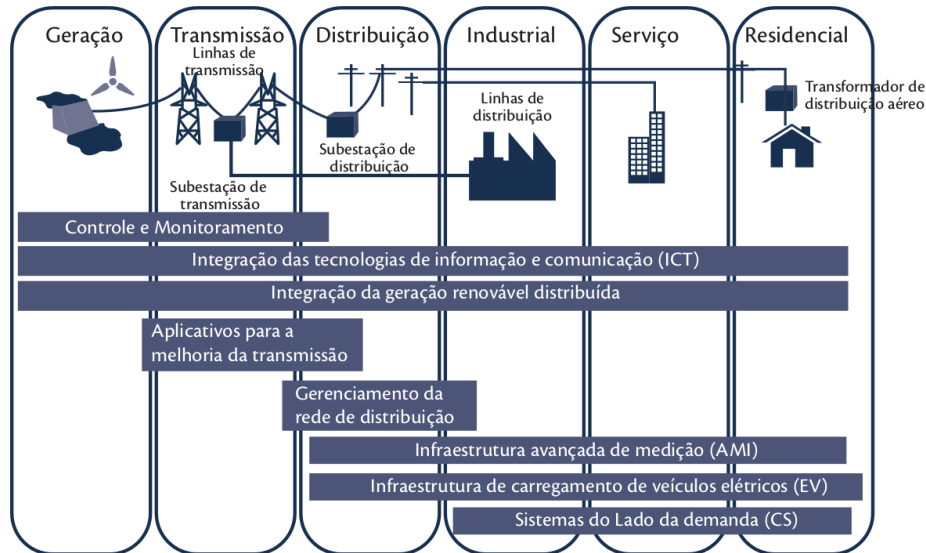
Assim, os dispositivos presentes na rede elétrica são dotados da capacidade de recepção e transmissão de dados via infraestrutura de comunicação avançada.

Na Figura 1.5 são mostradas as tecnologias utilizadas na composição de uma REI e sua presença de acordo com cada um dos níveis do SEP. A infraestrutura tecnologia da informação e comunicação é uma constante para qualquer que seja a tecnologia aplicada,



proporcionando controle e monitoramento das variáveis do sistema e proporcionando o desenvolvimento de aplicações inovadoras como infraestrutura de medição avançada, veículos elétricos e automação da distribuição.

Figura 1.5 – Tecnologias empregadas em uma Rede Elétrica Inteligente em diferentes níveis do SEP.



Fonte: (CGEE, 2012)

Com base nesse cenário, uma tendência atual é a implantação de esquemas de re-composição da rede elétrica de distribuição que permitam sua reconfiguração de forma automática (ZIDAN; EL-SAADANY; CHAAR, 2011) e a adoção de IED que proporcionem automação em todos os níveis do SEP.

De acordo com Sudhakar e Srinivas (2011) até o ano de 1993 apenas as técnicas de sistemas especialistas e base de conhecimento eram aplicadas ao problema da recomposição de sistemas de distribuição de energia elétrica, a partir deste ano muitas técnicas diferentes foram e vêm sendo propostas com o objetivo de buscar soluções mais eficientes para o problema, Sudhakar e Srinivas (2011) enumera as seguintes:

- Busca heurística;
- Sistemas Especialistas;
- Lógica Fuzzy;
- Otimização matemática;
- Redes de Petri;
- Algoritmos genéticos;
- Busca Tabu;

- Redes neurais artificiais;
- Algoritmos de colônias de formigas;

Estas técnicas, em sua grande maioria, utilizam uma abordagem de aquisição de dados e execução de seus algoritmos de forma centralizada.

Os sistemas de automação do SEP foram concebidos seguindo uma abordagem centralizada. Um exemplo claro disso são os sistemas SCADA que coletam dados dos dispositivos em campo e concentram a análise e processamento destes dados.

O setor elétrico tem passado por uma transformação em seu modelo de negócios, planejamento e operação que impacta diretamente na sua segurança. Ao invés de sua concepção original somente com grandes usinas injetando energia na rede, que é transportada por longas linhas de transmissão, chegando ao sistema de distribuição e por fim aos consumidores, o novo modelo da rede elétrica também prevê a presença de outros atores distribuídos ao longo da rede que podem interferir no fluxo de energia do SEP, injetando ou consumindo potência, podendo se conectar ou desconectar livremente no sistema.

Esse tipo de concepção vai de encontro a uma estrutura de controle centralizada, uma vez que para manter a rede segura, os atores, que podem ser por exemplo sistemas de geração distribuída, precisam ter acesso às informações operacionais do SEP. Ao centralizar as informações do sistema em uma unidade, será necessário grande poder de processamento pois a quantidade de dados gerada pelos atores deverá ser enviada à unidade central e disponibilizada para outras aplicações. Adicionalmente, centralizar o controle pode não ser uma abordagem eficiente de projeto neste tipo de situação, já que uma falha de comunicação na unidade central pode pôr em risco o funcionamento de todo o sistema.

## 1.2 Justificativa e Objetivos

Em um cenário de REI, em que é requerida inserção de tecnologia na rede elétrica por meio de funções avançadas de automação, a utilização de sistemas distribuídos é conveniente e até mesmo necessária para o bom desempenho destas funções.

Neste novo cenário baseado na utilização de sistemas distribuídos, os Sistemas Multiagente (SMA) surgem como uma técnica bastante promissora para aplicação em sistemas de potência, tendo em vista sua capacidade de modelagem, simulação e controle distribuído para desenvolvimento de funções de automação (MCARTHUR et al., 2007b; MCARTHUR et al., 2007a).

De acordo com McArthur et al. (2007a) as características e propriedades requeridas para desenvolvimento de SMA aplicados em sistemas elétricos são:

- Utilização de uma arquitetura aberta para sistemas distribuídos;

- Flexibilidade que denota a capacidade de um agente responder corretamente às situações dinâmicas e suportar replicação em diversas topologias da rede;
- Autonomia que implica na capacidade do agente programar suas próprias ações e construir um novo plano, se uma ação de controle em particular, falhar ou mudar em função das alterações na topologia da rede, por exemplo;
- Extensibilidade (capacidade de expansão) que consiste na capacidade do sistema permitir adição de novas funcionalidades ou alteração das funcionalidade existente;
- Tolerância a falha consiste na capacidade do agente ou sistema multiagente possuir recursos para solução de problemas quando uma parte do sistema falhar, de forma a atingir as metas para o qual foi projetado, sem afetar o desempenho do sistema

Visando um sistema com processamento distribuído e com características de robustez, flexibilidade, tolerância a falhas, extensibilidade e escalabilidade, este trabalho tem como objetivos gerais o desenvolvimento de uma plataforma para desenvolvimento e execução de SMA em linguagem Python e baseado nos padrões estabelecidos pela *Foundation for Intelligent Physical Agents* (FIPA), assim como a implementação de um sistema de localização, isolamento e recomposição automática pós-falha, de alimentadores de redes de distribuição de energia elétrica, atendendo os critérios de prioridade e respeitando as restrições dos equipamentos, utilizando a técnica de SMA.

Os objetivos específicos são:

- Propor uma estrutura composta por agentes Alimentador e agentes Dispositivo para a realização da recomposição;
- Embarcar os agentes em diferentes plataformas de hardware distribuídos e com possibilidade de comunicação via interface de rede;
- Desenvolver componentes de software que permitam o estabelecimento de comunicação entre agentes e IED utilizando os protocolos GOOSE e MMS, estabelecidos na norma IEC 61850 e assim implementar a função de autorecuperação (*self-healing*) aos alimentadores de redes de distribuição de energia elétrica;
- Utilizar teoria dos grafos, por meio de uma estrutura de dados chamada representação nó-profundidade (RNP), para representar a topologia e as características de alimentadores de distribuição;
- Desenvolver uma interface de programação de aplicação (API) para representação dos componentes do sistema elétrico necessários à análise de recomposição e que seja baseada em RNP.

- Realizar testes com o SMRA utilizando uma plataforma distribuída que compreende softwares embarcados em hardwares conectados em rede com IED reais e virtuais, recebendo sinal de corrente gerado a partir de um dispositivo de testes de relés.

## 1.3 Metodologia

A rede elétrica foi representada por meio de uma API orientada a objetos, que representa os componentes do Sistema Elétrico de Distribuição (SED) por meio de classes do tipo *Subestacao*, *Alimentador*, *Setor*, *Trecho*, *NoDeCarga* e *Chave*. Essas classes possuem associações entre si, que podem ser representadas na forma de diagramas de classe da *Unified Modeling Language* (UML). Para que a rede elétrica possa ser manipulada e analisada de forma eficiente, buscou-se sua representação por meio da teoria dos grafos, com a representação computacional de nó-profundidade (RNP), proposta por (DELBEM et al., 2004).

Para que um sistema de recomposição automática seja capaz de implementar a função de *self-healing* é necessário que exista uma comunicação bidirecional entre aplicação, que realiza as análises de restauração da rede, e os dispositivos de proteção e controle (IED). Desta forma, uma infraestrutura de comunicação que conecte estas duas partes, e um conjunto de protocolos com os requisitos necessários para uma ação rápida, eficiente e segura são indispensáveis. Estes requisitos são proporcionados pelos protocolos da norma IEC 61850 que integra os dispositivos por meio de uma rede de comunicação TCP/IP.

Para a comunicação dos agentes com os IEDs foi utilizada a biblioteca *libiec61850*, software livre, e implementada na linguagem de programação C. *libiec61850* implementa aplicações cliente/servidor com capacidade de enviar e receber mensagens do protocolo MMS, assim como aplicações editor/assinante com protocolo GOOSE. Uma vez que os agentes foram desenvolvidos em Python, algumas das funcionalidades cliente da biblioteca foram traduzidas para Python fazendo um link com o código em C. Para simular os servidores IEC 61850, o código em C, sem link com Python, foi utilizado.

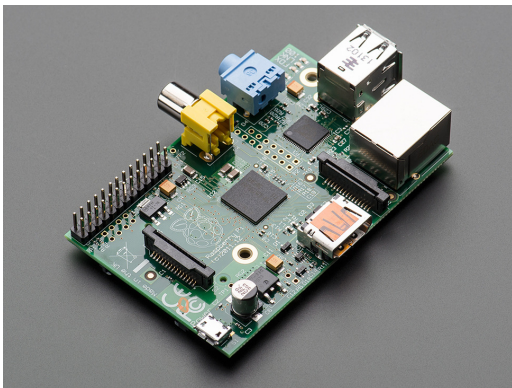
Neste trabalho, a implementação do SMA foi realizada utilizando a linguagem de programação Python. No entanto, não existem projetos funcionais de código aberto para a implementação de SMA, com base nos padrões FIPA (Foudation for Intelligent Physical Agents), em Python, ao contrário da plataforma JADE (Java Agent DEvelopment framework) em Java.

Python possui poderosas bibliotecas para desenvolvimento de aplicações em sistemas distribuídos e vem sendo utilizado com sucesso para construção de aplicações clientes e servidores utilizando diversos protocolos existentes. Dessa forma optou-se pela construção de um framework em Python para o desenvolvimento de SMA distribuído e que siga as principais especificações dos padrões elaborados pela FIPA.

Para a construção do framework Python foi utilizada a biblioteca Twisted, que implementa protocolos já existentes e possui ferramentas para implementação de protocolos personalizados. O Twisted possui um núcleo de processamento assíncrono, sendo assim, todas as conexões são processadas em uma única thread assíncronamente, economizando recursos do sistema operacional.

Os agentes foram embarcados nas placas de desenvolvimento RaspberryPi e BeagleBone Black. Estes dispositivos de hardware possuem elevado poder de processamento, com processadores ARM de 700 MHz (RaspberryPi) e 1,0 GHz (BeagleBone Black), ambos com 512 MB de memória RAM e 8 GB para armazenamento permanente, sendo capazes de executar um sistema operacional Linux com interface de rede, sistema de arquivos, pinos de IO e barramento USB para conexão de dispositivos periféricos. Na Figura 1.6 são mostradas as duas placas de desenvolvimento, na Figura 1.6a o Raspberry Pi B e na Figura 1.6b o Beaglebone Black. Sua portabilidade foi fator decisivo na escolha destes dispositivos, uma vez que podem ser facilmente instalados ocupando espaço físico bem reduzido.

Figura 1.6 – Placas de hardware embarcado Raspberry Pi B e Beaglebone Black.



(a) Raspberry Pi B.



(b) Beaglebone Black recebida.

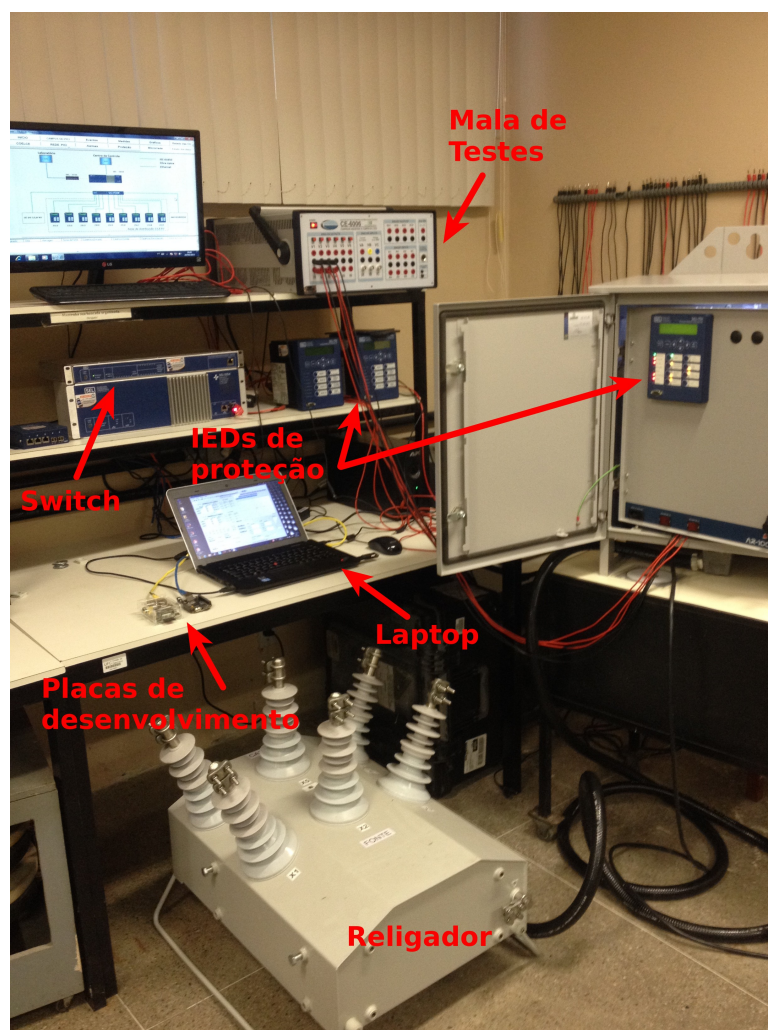
Fonte: (ADAFRUIT, 2015b; ADAFRUIT, 2015a)

Para realização de testes de validação práticos, foram utilizados três relés SEL-751, que são relés de proteção digitais, multifunção de uso comercial, utilizados para proteção de alimentadores, transformadores de pequeno porte, bancos de capacitores, reatores e geradores.

Para simulação de ocorrências na rede elétrica, a mala de testes hexafásica Conprove CE-6006 foi utilizada para gerar sinais de corrente às entradas analógicas do relé de proteção SEL-751.

A estrutura de testes pode ser vista na Figura 1.7.

Figura 1.7 – Bancada experimental de testes de validação.



Fonte: Própria do autor.

## 1.4 Revisão Bibliográfica

Conforme mencionado anteriormente, o problema da reconfiguração de redes elétricas vem sendo estudado há algum tempo pela comunidade acadêmica. Estes estudos podem ser divididos em duas categorias: estudos que visam a reconfiguração da rede com vistas a otimizar a operação da rede elétrica por meio da minimização das perdas e os estudos que visam a reconfiguração da rede mediante a ocorrência de contingências no SED.

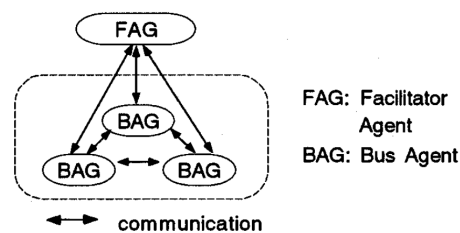
Este trabalho pertence à segunda categoria, dos trabalhos que estudam a reconfiguração da rede mediante situação de contingência. Os principais objetivos abordados são a minimização dos consumidores desenergizados e a minimização do número de manobras realizadas para o restabelecimento da rede, mediante a não violação das restrições operacionais dos dispositivos presentes no SED. Outros critérios podem ainda ser levados em consideração como, por exemplo, a presença de cargas prioritárias e até mesmo a possibilidade de ilhamento de parte da rede que possui geração distribuída.

Neste aspecto, existem diversos trabalhos que empregam diferentes técnicas de inteligência computacional. Existem os trabalhos que empregam abordagens centralizadas e aqueles que utilizam abordagens distribuídas. O presente trabalho utiliza a técnica de SMA para o desenvolvimento de um sistema de recomposição automática com filosofia distribuída, uma vez que SMA surge como uma técnica bastante promissora para aplicação em sistemas de potência, tendo em vista sua capacidade de modelagem, simulação e controle distribuído (MCARTHUR et al., 2007b; MCARTHUR et al., 2007a).

SMA vem sendo bastante utilizado para o desenvolvimento de sistemas distribuídos e será discutido com mais detalhes no Capítulo 2. Muitos trabalhos utilizam SMA para construção de sistemas de recomposição automática distribuídos, alguns deles serão discutidos a seguir.

Nagata e Sasaki (2002) propõem um esquema de recomposição para uma rede composta de oito subestações e quatorze barras. Dois tipos de agentes são utilizados para realizar a recomposição, agente barra (BAG) e um agente facilitador (FAG). O agente barra, que pode ser mais de um, é desenvolvido para encontrar uma configuração sub-ótima da rede após a ocorrência de um problema através da negociação de potência com outros agentes barra. O agente facilitador, apenas um para toda a rede, tem como objetivo gerenciar o processo de decisão, organizando os BAG por nível de tensão e decidindo qual BAG deve iniciar a negociação. Por centralizar ações num único FAG, esta abordagem é considerada centralizada. Um simples esquema da comunicação entre os agentes BAG e FAG é mostrado na Figura 1.8.

Figura 1.8 – Comunicação entre os agentes BAG e FAG.



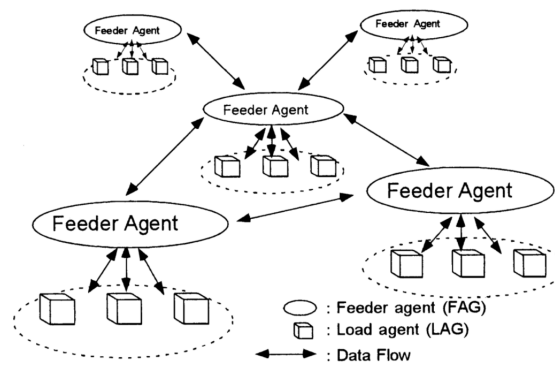
Fonte: (NAGATA; SASAKI, 2002).

Uma proposta de SMA descentralizado é apresentada em Nagata et al. (2003). Trata-se de um sistema de recomposição definido por agentes em dois níveis hierárquicos. No nível mais baixo foram definidos agentes carga (LA), que possuem informações locais dos trechos a que estão associados, como por exemplo capacidade dos condutores, corrente demandada pelo trecho e estado das chaves que definem o trecho. Em um nível superior estão os agentes alimentadores (FA) que são responsáveis por negociar potência com outros agentes alimentadores e também definem por qual agente carga deve ser iniciada a recomposição. Esta estrutura está descrita na Figura 1.9. Para realizar a recomposição dos trechos, os agentes alimentadores consideram três condições e priorizam a recomposição

na seguinte ordem: primeiro é verificada a possibilidade de recompor os trechos pelo mesmo transformador que alimentava as cargas desenergizadas. Caso não seja possível, a possibilidade de recomposição pela mesma subestação é verificada. Não sendo possível a recomposição por nenhum destes casos, então busca-se a recomposição por meio de alimentadores conectados a outra subestação. Nesta abordagem a recomposição é realizada trecho a trecho, sendo necessário antes enviar comando de abertura para todas as chaves do alimentador afetado.

Este SMA possui características distribuídas, o que de acordo com Barros (2013) torna a proposta bastante atrativa para o sistema de distribuição onde há uma grande quantidade de equipamentos e dispositivos.

Figura 1.9 – Comunicação entre Feeder agents e Loads.



Fonte: (NAGATA et al., 2003).

Em Solanki, Khushalani e Schulz (2007), um SMA distribuído para recomposição de SED e sem a presença de um agente centralizador é apresentado. O autor propõe três tipos de agentes: agente carga (LA), agente chave (CA) e agente gerador (GA) que interagem em um processo de quatro passos:

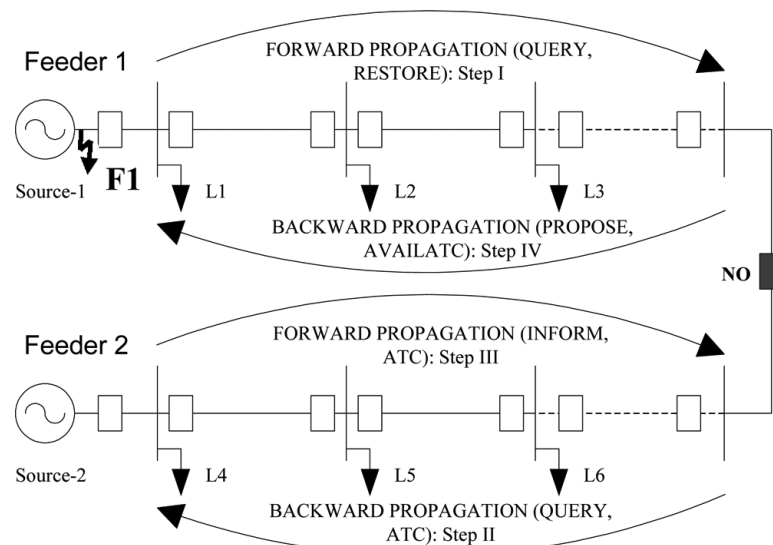
- O agente carga percebe a desenergização de seu barramento e repassa informações de pré-falta ao agente chave, que por sua vez se conecta aos demais agentes chave que também receberam mensagem de seus agentes carga que estão desenergizados;
- O agente chave que está associado à uma chave de encontro recebe o montante de potência que precisa ser restabelecida e repassa essa informação em cascata, de agente chave em agente chave até o agente geração;
- O agente gerador envia a capacidade disponível da fonte para o agente chave associado à chave de encontro do alimentador desenergizado;
- O agente chave envia a mensagem do agente gerador para o agente carga desenergizado, que verifica a possibilidade de recomposição do trecho associado a ele. Caso seja possível a recomposição, o agente carga retorna mensagem de confirmação para



que o agente chave atue e restabeleça o trecho. Os dados do agente gerador são então atualizados e passados para o próximo agente chave que realiza o mesmo comportamento. Isso é feito até que não seja mais possível a recomposição pela fonte escolhida.

Na Figura 1.10 é mostrada uma representação dos quatro passos desta abordagem da recomposição.

Figura 1.10 – Representação esquemática dos quatro passos da restauração definida por Solanki, Khushalani e Schulz (2007).

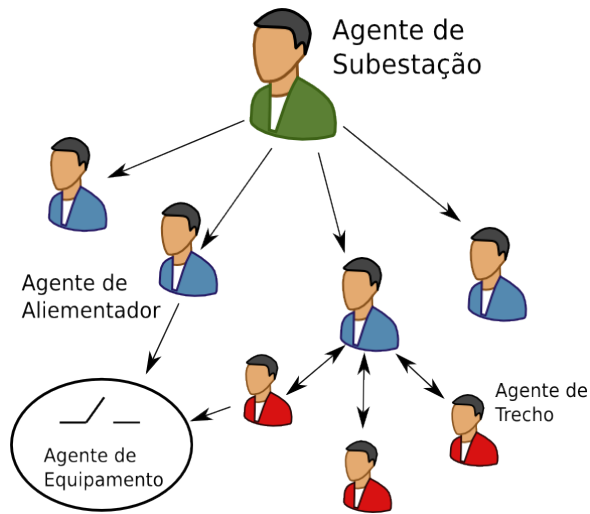


Fonte: (SOLANKI; KHUSHALANI; SCHULZ, 2007)

Em Melo (2013) é proposto um SMRA baseado na metodologia apresentada em Nagata et al. (2003), com a presença de agentes alimentador e agentes trecho, mas com a adição de dois novos tipos de agentes, o agente subestação e o agente equipamento. Na Figura 1.11 é mostrado um esquema representativo destes agentes e dos canais de comunicação existentes entre eles. A recomposição ocorre nas seguintes etapas:

- A falta é identificada pelo agente equipamento que repassa a informação ao agente alimentador associado, juntamente com os dados de pré-falta;
- O agente alimentador envia comando de abertura para todas as chaves a jusante do trecho faltoso, faz os cálculos da potência necessária para recompor os trechos desenergizados e se comunica com os demais agentes alimentadores a fim de encontrar a solução mais adequada para a recomposição;
- Após escolher aquela que julga ser a melhor opção de recomposição, o agente alimentador se comunica com o agente trecho que está conectada à chave de encontro do alimentador escolhido para recompor os trechos desenergizados;

Figura 1.11 – Agentes definidos em Melo (2013).

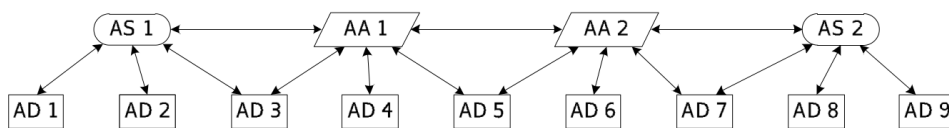


Fonte: (MELO, 2013)

- O agente trecho realiza análises de restrições e caso seja possível envia comando de fechamento para o agente equipamento fechar a chave e realizar a recomposição. Esse comportamento se repete até que todos os trechos sejam recompostos ou que não seja mais possível a recomposição.

Barros (2013) propõe um sistema de recomposição automática com três tipos de agentes: dispositivo (AD), alimentador (AA) e subestação (AS). Os AS estão associados a cada uma das subestações do SED e são responsáveis por analisar o carregamento dos transformadores da subestação e compartilhar essa informação com os AA. Além disso, são responsáveis por gerenciar a recomposição do sistema quando acontece uma falta em algum dos transformadores sob sua responsabilidade. Os AA, por sua vez, gerenciam o carregamento dos trechos e gerenciam também a recomposição do sistema, trecho a trecho. Os AD estão integrados aos IED associados aos dispositivos de disjunção presentes na rede (Chaves automáticas, religadores ou disjuntores); os AD interpretam os dados do sistema elétrico disponibilizados pelos IED e filtram informações relevantes aos AA e AS. Na Figura 1.12 é possível visualizar um diagrama que representa as associações entre os agentes propostos em (BARROS, 2013).

Figura 1.12 – Agentes definidos em Barros (2013).



Fonte: (BARROS, 2013)

Na Tabela 1.1 são apresentados os trabalhos analisados até aqui, exibindo o tipo de aplicação do SMA e quais seus principais componentes, além de classificar as abordagens

como centralizada ou descentralizada. Como pode ser visto, a grande maioria dos trabalhos explora uma abordagem descentralizada, por ser uma importante característica dos SMA. Já a quantidade de agentes depende exclusivamente da modelagem do sistema adotada pelo desenvolvedor.

Tabela 1.1 – Trabalhos correlatos de desenvolvimento de sistemas de restauração automática.

| Referência                          | Aplicação   | Tipos de agentes   | Controle        | Plataforma |
|-------------------------------------|---|--|-----------------|------------|
| (NAGATA; SASAKI, 2002)              | Recomposição de sistema de transmissão de energia               | agente barra<br>agente facilitador   | Centralizado    | -          |
| (NAGATA et al., 2003)               | Recomposição de sistema de distribuição de energia              | agente alimentador<br>agente carga   | Descentralizado | -          |
| (SOLANKI; KHUSHALANI; SCHULZ, 2007) | Recomposição de sistema de distribuição com geração distribuída | agente chave<br>agente carga<br>agente gerador                                 | Descentralizado | JADE       |
| (MELO, 2013)                        | Recomposição de sistemas de distribuição de energia             | agente subestação<br>agente alimentador<br>agente trecho<br>agente equipamento | Descentralizado | JADE       |
| (BARROS, 2013)                      | Recomposição de sistemas de distribuição de energia             | agente subestação<br>agente dispositivo<br>agente alimentador                  | Descentralizado | JADE       |

Fonte: Adaptado de (BARROS, 2013)

Entre os diferenciais deste trabalho em relação aos trabalhos apresentados na Tabela 1.1 é possível citar:

- Implementação dos agentes em Python, utilizando uma nova plataforma de desenvolvimento, o PADE;
- Modelagem de uma nova abordagem para recomposição da rede elétrica de distribuição utilizando sistemas multiagentes e representando a rede por meio da estrutura de dados RNP;
- Desenvolvimento do conceito de módulos de hardware de baixo custo embarcados, onde os agentes serão executados;
- Implementação da comunicação entre os agentes e os IED por meio dos protocolos estabelecidos pela norma IEC 61850.

Para que estes objetivos fossem alcançados, este trabalho foi necessário o desenvolvimento dos seguintes componentes de software:

- API para representação da rede elétrica de distribuição de energia, que permite uma visão topológica da rede, por meio de teoria dos grafos, que são representados através da estrutura RNP e ainda a realização de estudos de fluxo de carga e de curto-circuito da rede representada.
- biblioteca que permite fácil integração de aplicações Python com dispositivos que se comunicam nos padrões descritos na norma IEC 61850, permitindo funcionalidades como, estabelecimento de conexão com dispositivos, requisição de seu modelo de dados, leitura de valores digitais e analógicos dos dados do dispositivo e envio de comandos para dispositivo.

## 1.5 Estrutura do Trabalho

O trabalho está dividido em sete capítulos. No Capítulo 1, o trabalho é contextualizado com a apresentação de motivação, justificativa, objetivos, metodologia e revisão bibliográfica para desenvolvimento de sistema multiagente para recomposição automática de redes elétricas.

No Capítulo 2 uma revisão bibliográfica sobre sistemas multiagentes é apresentada, algumas definições sobre SMA são discutidas e os conceitos sobre plataforma multiagente baseados nos padrões FIPA, apresentados.

No Capítulo 3 é apresentada a implementação de uma plataforma, baseada nos padrões definidos pela FIPA, para desenvolvimento, execução e monitoramento de agentes distribuídos utilizando ferramentas disponíveis na linguagem Python.

No Capítulo 4 o problema da recomposição de sistemas elétricos de distribuição é definido formalmente, juntamente com alguns conceitos de grafos, da representação nó-profundidade, da sua relação com os componentes do SED e a modelagem das classes que serão utilizadas pelo sistema de recomposição automática para analisar a rede e tomar as decisões que resultarão na recomposição do sistema.

No Capítulo 5, são discutidos aspectos da implementação real do sistema de recomposição automática, entre eles o mais importante são os conceitos chave da norma IEC 61850, como, sua estrutura, seu modelo de dados, seus protocolos e seus serviços. Em seguida, os conceitos da norma são exemplificados através de testes realizados por meio da biblioteca `libiec61850`, que fornece a base para o desenvolvimento de um módulo IEC 61850 em Python.

No Capítulo 6, são discutidos os resultados das simulações computacionais do SMRA realizadas seguindo uma ordem crescente de proximidade às condições reais de operação do sistema. Primeiro, os agentes são executados em uma única máquina, em seguida são

---

embarcados nas placas de desenvolvimento Raspberry Pi e Beagle Bone Black, e por fim são incorporados aos testes dois IEDs reais, do fabricante SEL, comunicando via protocolos de comunicação da norma IEC 61850 e simulação de ocorrência na rede por meio da mala de testes Conprove CE-6006.

No Capítulo 7 são apresentadas as conclusões do presente trabalho, bem como propostas de melhorias e trabalho futuros, além de contribuições e trabalhos publicados em eventos.

---

# Capítulo 2

## Sistemas Multiagentes

---

### 2.1 Introdução

Neste Capítulo, o conceito de sistemas multiagentes é apresentado e discutido, juntamente com suas tecnologias e ferramentas. No início é definido o conceito de agente inteligente (AI). Uma vez que um AI possua habilidade social, seu conceito é estendido para compor o escopo de sistemas multiagentes. Na sequência, o termo ontologia é definido e alguns exemplos são discutidos, para melhor compreensão do termo. Os padrões FIPA são descritos e evidenciados logo em seguida, uma vez que desempenham importante papel no desenvolvimento de um SMA e fornecem a base do Python Agent DEvelopment framework (PADE).

### 2.2 Agentes Inteligentes

Muitas aplicações computacionais utilizam agentes inteligentes (AI) no desenvolvimento de suas tarefas. Atualmente AI é empregado em campos como inteligência artificial, banco de dados, sistemas operacionais, redes de computadores, sistemas elétricos, entre outras aplicações (BELLIFEMINE; CAIRE; GREENWOOD, 2007). Muitas áreas da ciência e da engenharia desenvolvem pesquisas que utilizam AI na resolução de problemas complexos. Para alguns destes problemas uma abordagem distribuída é mais conveniente, tornando o problema mais fácil de ser modelado e/ou solucionado.

#### 2.2.1 Definições e Características de AI

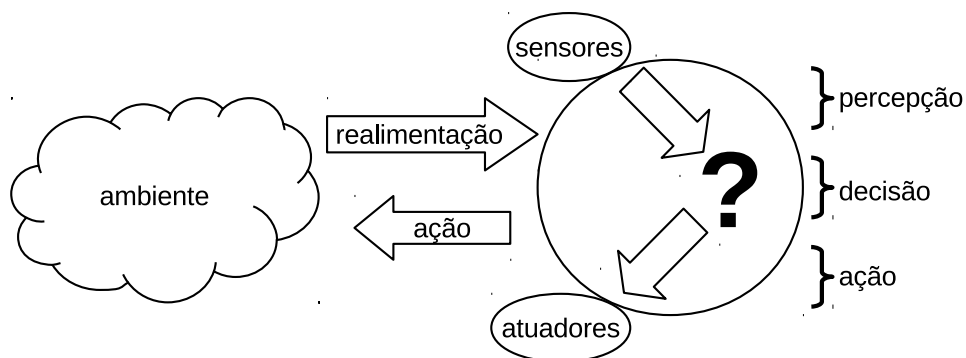
De acordo com Wooldridge (2009):

‘Sistemas multiagentes são sistemas compostos por múltiplos elementos computacionais chamados de agentes, que interagem entre si. Agentes são sistemas computacionais com duas importantes capacidades. Primeiro, são capazes de realizar ações autonomamente, decidindo por si só, o que precisam fazer para atingir os objetivos para que foram especificados. Segundo, agentes são capazes de interagir com outros agentes, não apenas trocando informações, mas interagindo entre si, apresentando comportamentos semelhantes às interações sociais cotidianas dos seres-humanos: cooperação, coordenação, negociação, entre outras’. (Tradução própria)

Wooldridge (2009) afirma ainda que: “Um agente é um sistema de computador que está situado em um determinado ambiente e que é capaz de agir de modo autônomo neste ambiente a fim de atingir os objetivos que lhe foram especificados” (Tradução própria).

Na Figura 2.1 é mostrado o esquema representativo de um agente atuando sobre um determinado ambiente. De acordo com Wooldridge (2009), em diversos casos o agente não tem total domínio sobre o ambiente sobre o qual interage, ou seja, o agente só tem controle parcial sobre o ambiente, podendo apenas influenciá-lo. Este tipo de comportamento não determinístico, implica que uma mesma ação, pode gerar resultados diferentes no ambiente, em circunstâncias aparentemente iguais, gerando inclusive resultados não desejados. Dessa forma, agentes que interagem nestas circunstâncias devem estar preparados para reações adversas do ambiente.

Figura 2.1 – Representação esquemática de um agente interagindo com seu ambiente.



Fonte: (WOOLDRIDGE, 2009)

Diante destas definições, uma pergunta é recorrente. Quando é possível afirmar que um agente possui comportamento inteligente? Wooldridge e Jennings (1995) tentam responder a esta pergunta listando algumas características esperadas em um agente inteligente. São elas:

- Reatividade: AI devem ser capazes de perceber alterações em seu ambiente e num curto intervalo de tempo realizar as ações necessárias para que seus objetivos especificados possam ser alcançados;

- Proatividade: AI devem ser capazes de tomar decisões de forma autônoma, ou seja, sem a necessidade de serem solicitados para isso;
- Habilidade Social: AI devem ser capazes de interagir com outros agentes, sempre que for necessário para realizar seus objetivos.

## 2.2.2 Comunicação entre AI

Para que a habilidade de interação social seja atingida entre agentes, primeiro é necessário que os agentes utilizem uma terminologia comum e que seja entendida por todos aqueles que participam do ato comunicativo. Por exemplo, ao solicitar uma certa quantidade de potência elétrica de um agente, aqueles que participam da conversa devem saber, por exemplo, o que significa o termo potência aparente, ou a unidade MVA. Assim os agentes precisam conhecer uma ontologia comum ao conteúdo de suas mensagens. Em Wooldridge (2009) o termo ontologia é definido como:

‘Uma ontologia é uma definição formal de um corpo de conhecimento. O tipo mais comum de ontologia utilizada na construção de agentes envolvem componentes estruturais. Essencialmente uma taxonomia de classes e relações de subclasses agrupadas com definições de relacionamentos entre si’. (Tradução própria)

De acordo com esta definição, ontologia define um corpo de conhecimento que envolve relações entre componentes estruturais. Estes componentes podem ser representados, por exemplo, por classes e associações entre subclasses.

Essas associações podem ser expressas por meio de diagramas de classe UML, que relacionam classes e definições de dependência, como herança, associação, agregação e composição (MCMORRAN, 2007).

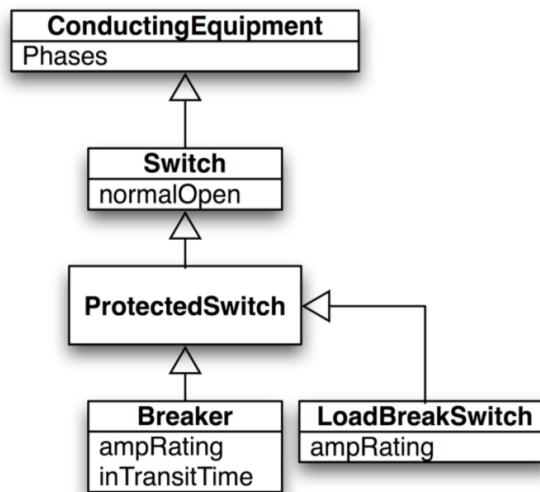
O padrão definido nas normas IEC61970-301 e IEC61968-11 e conhecido como *Common Information Model* (CIM) é um exemplo, mais específico, de associações de classes, que tem como propósito definir um modelo de dados padronizado para o SEP, e conseqüentemente uma ontologia para os termos associados ao SEP. Na Figura 2.2 é mostrado um diagrama de classes UML, as quais estão no padrão CIM. No diagrama é possível identificar, por exemplo, as associações de herança entre as classes que podem ser lidas da seguinte forma: *Disjuntor é uma chave de proteção. Chave com abertura em carga é uma chave de proteção. Chave de proteção é uma chave. Chave é um equipamento condutor.*

## 2.2.3 Linguagem de Ontologia

Existem diversas formas de representar ontologias, que são chamadas de linguagens de ontologia. Uma das mais populares linguagens de ontologia é a *extensible markup language* - XML.



Figura 2.2 – Diagrama de classes representando associações entre diferentes classes do padrão CIM.



Fonte: (MCMORRAN, 2007)

A XML não é uma linguagem de ontologia por si mesma, mas pode ser utilizada para definir ontologias simples (WOOLDRIDGE, 2009). Na Figura 2.3 é mostrada uma representação em XML da classe *Breaker* definida pelo padrão CIM. Por meio das tags XML é possível obter informações úteis a respeito de um componente do SEP. As tags são representações de um vocabulário controlado que provêm uma ontologia simples, mas de grande utilidade.

## 2.3 Comunicação entre Agentes

A comunicação ocupa um papel central em qualquer SMA, além disso, para que dois agentes possam se comunicar é necessário que estes agentes entendam as mensagens transmitidas, tanto no nível de semântica (ontologia), quanto no nível de sintaxe (protocolo).

Com o intuito de padronizar alguns aspectos dos SMA, entre eles a comunicação, o *Institute of Electrical and Electronics Engineers* (IEEE) decidiu criar uma organização para cuidar de todos os detalhes relacionados à padronização de SMA, e decidiu chamá-la de *Foundation for Intelligent Physical Agents* (FIPA), que está vinculada à Sociedade de Computação do IEEE.

A FIPA tem por objetivo central definir padrões para tecnologias baseadas em sistemas multiagentes, proporcionando interoperabilidade entre os SMA e outras tecnologias (FIPA, 2015). A FIPA foi originalmente fundada em 1996 por um grupo de organizações acadêmicas e industriais que tinham como objetivo definir uma série de padrões e especificações que permitissem a usabilidade desses sistemas em uma ampla gama de aplicações (BELLIFEMINE; CAIRE; GREENWOOD, 2007).

Figura 2.3 – Representação em XML da classe Breaker definida no padrão CIM, como exemplo de estrutura ontológica.

```
<?xml version="1.0" ?>
<Breaker>
  <Phases>
    <Phase>A</Phase>
    <Phase>B</Phase>
    <Phase>C</Phase>
  </Phases>
  <ampRating>
    <multiplier>1</multiplier>
    <unit>A</unit>
    <value>800</value>
  </ampRating>
  <inTransitingTime>
    <multiplier>m</multiplier>
    <unit>s</unit>
    <value>300</value>
  </inTransitingTime>
  <normalOpen>False</normalOpen>
</Breaker>
```

Fonte: Própria do autor.

A FIPA define padrões relacionados a três questões centrais em SMA: comunicação entre agentes, gerenciamento de agentes, e arquitetura dos agentes. Nas subseções que seguem são apresentados os principais conceitos referentes à estrutura de comunicação e gerenciamento de agentes definidos pela FIPA.

### 2.3.1 FIPA-ACL

A *agent communication language* (ACL) definida pela FIPA é bastante similar à *knowledge query and manipulation language* (KQML), ou seja, as mensagens ACL podem ser interpretadas como objetos (em termos de linguagens orientadas a objetos). Mensagens ACL possuem uma performativa, que pode ser pensada como uma classe de mensagem, e uma série de parâmetros, que são os atributos do objeto mensagem (WOOLDRIDGE, 2009).

As performativas são representadas por rótulos identificadores padronizados de cada mensagem que é trocada entre os agentes, e fornecem uma pré-descrição acerca do objetivo do ato comunicativo. Por exemplo, quando um agente quer informar o resultado de alguma tarefa para outro agente, este pode utilizar uma mensagem com performativa *INFORM*, mas quando precisa solicitar uma proposta, é mais conveniente que envie uma

mensagem com performativa *CALL FOR PROPOSES* (CFP). O padrão ACL possui 22 atos comunicativos, alguns dos mais utilizados, juntamente com sua descrição, estão expostos na Tabela 2.1.

Tabela 2.1 – *Performativas* definidas pela FIPA.

| <b>Atos de comunicação definidos pela FIPA</b> | <b>Descrição</b>  |
|--|---|
| Accept Proposal                                | Ato de aceitar uma proposta submetida anteriormente.  |
| Agree  | Ato de aceitar a realização de uma ação.  |
| Call for Proposal                              | Ato de chamar por propostas para a realização de uma determinada ação.  |
| Failure  | Ato de informar a outro agente que uma ação foi atendida, mas que houve falha em sua execução.  |
| Inform   | O remetente informa ao receptor que uma dada proposição é verdadeira.   |
| Not Understood                                 | O remetente informa ao receptor que percebeu a realização de uma ação por parte do receptor, mas que não compreendeu a realização daquela ação. |
| Propose  | Ato de submeter uma proposta, para a realização de uma ação.  |
| Refuse   | Ato de recusar a execução de uma ação dada, explicando os motivos da recusa.  |
| Reject Proposal                                | Ato de rejeitar uma proposta para a realização de uma ação durante uma negociação.  |
| Request  | O remetente requisita ao receptor a realização de uma ação.   |
| Subscribe                                      | O ato de requisitar a notificação do valor de uma referência, para o remetente, sempre que este valor for alterado.                             |

Fonte: (FIPA00037, 2004)

Na Tabela 2.2 são mostrados os parâmetros das mensagens ACL e seus respectivos significados. Por meio dos parâmetros de uma mensagem é possível implementar filtros que selecionam determinadas mensagens para um comportamento específico de um agente. Por exemplo, para tratar mensagens enviadas exclusivamente de um agente, basta que se crie uma filtro que irá acessar o parâmetro *sender* da mensagem e verificar se este parâmetro, condiz com os valores predefinidos no filtro criado para esta mensagem. No caso do filtro retornar valor verdadeiro, então o comportamento associado ao filtro é executado, caso contrário, nada é feito.

Na Figura 2.4 é mostrado um exemplo de uma mensagem ACL no formato de string, com seus parâmetros e valores, conforme especificado em (FIPA00070, 2004).

Nem sempre o formato mostrado na Figura 2.4 é apropriado, ou conveniente, para a troca de mensagens entre agentes, ou entre agentes e aplicações. Nesse caso as mensagens no padrão ACL, podem ser representadas por meio de uma estrutura xml, um padrão universalmente aceito e adotado, sem perda nenhuma de significado. A mesma mensagem

Tabela 2.2 – Campos de Mensagem Padronizados.

| Parâmetro       | Descrição   |
|-----------------|---|
| performative    | Tipo do ato comunicativo da mensagem.                                 |
| sender          | Identidade do remetente da mensagem.                                  |
| receiver        | Identidade dos destinatários da mensagem.                             |
| reply-to        | Agente para dirigir mensagens subsequentes dentro de uma conversa.    |
| content         | Conteúdo da mensagem.   |
| language        | Linguagem no qual o conteúdo da mensagem é expresso.                  |
| encoding        | Codificação específica do conteúdo da mensagem.                       |
| ontology        | Ontologia que dá sentido aos símbolos no conteúdo da mensagem.        |
| protocol        | Protocolo de interação utilizado para estruturar uma conversa.        |
| conversation-id | Identidade única de uma conversa.                                     |
| reply-with      | Expressão utilizada pelo agente de resposta a identificar a mensagem. |
| in-reply-to     | A referência a uma ação anterior em que a mensagem é uma resposta.    |
| reply-by        | Data/hora indicando quando uma resposta deve ser recebida.            |

Fonte: (FIPA00061, 2004)

Figura 2.4 – Exemplo de mensagem no padrão FIPA-ACL.

```
(request
  :sender (agent-identifier :name raspberrypi@192.168.0.101:4001)
  :receiver (agent-identifier :name beaglebone@192.168.0.102:4001)
  :ontology CIM
  :language json
  :protocol fipa-request
  :content
    """"{'ref' : 'R_01',
        'data' : {'chaves' : [1, 2], 'estados' : [0, 1]}}
    """"
)
```

Fonte: Própria do autor.

da Figura 2.4, está exibida na Figura 2.5, mas no padrão xml, que também pode ser utilizado na troca de mensagens entre agentes e está especificado em (FIPA00071, 2004).

## 2.4 Plataforma Multiagente

A FIPA define um modelo padronizado de plataforma de agentes que é composto de quatro elementos principais (FIPA00023, 2004):

Figura 2.5 – Exemplo de mensagem no padrão FIPA-ACL em formato XML

```

<?xml version="1.0" ?>
<ACLMessage date="25/03/2015 as 21:00:47:427816">
  <performative>request</performative>
  <sender>raspberrypi@192.168.0.101:4001</sender>
  <receivers>
    <receiver>beaglebone@192.168.0.102:4001</receiver>
  </receivers>
  <content>{'ref': R_01,
            'data': {'chaves': [1, 2], 'estados': [0, 1]
                    }}
  </content>
  <language>json</language>
  <ontology>CIM</ontology>
  <protocol>fipa-request protocol</protocol>
  <conversationID>
    27f5344a-d34b-11e4-b6af-e8b1fc5c3cdf
  </conversationID>
</ACLMessage>

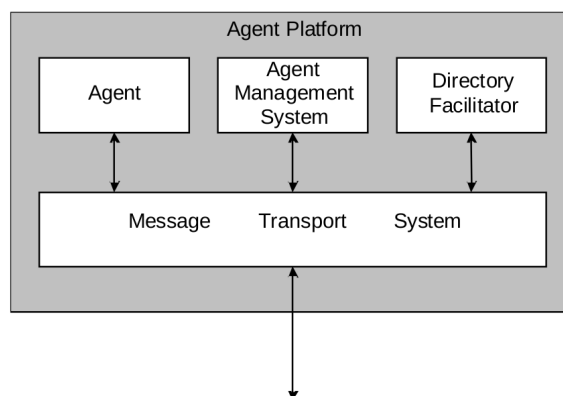
```

Fonte: Própria do autor.

- **Agentes:** São os agentes propriamente ditos, que executam suas atividades na plataforma e que constituem o corpo do SMA.
- **Agent Management System (AMS):** É o agente que supervisiona o acesso e a utilização da plataforma de agentes. Só um AMS é permitido em uma plataforma multiagente. Cada agente precisa da permissão do AMS para poder entrar na plataforma e ficar visível para os demais agentes;
- **Directory Facilitator (DF):** Este agente provê o serviço de páginas amarelas à plataforma, ou seja, é por meio do DF que os agentes que possuem serviços para disponibilizar aos demais cadastram seus serviços e é também por meio do DF que os agentes que precisam de serviços procuram por eles.
- **Message Transport System (MTS):** É a camada de software que controla todas as trocas de mensagens no SMA, incluindo mensagens entre agentes numa mesma plataforma e também entre agentes em diferentes plataformas.

O relacionamento destes componentes pode ser visualizado na Figura 2.6.

Figura 2.6 – Elementos constituintes de uma plataforma multiagente.



Fonte: Adaptado de (FIPA00023, 2004)

## 2.5 Protocolos de Comunicação Definidos pela FIPA

De acordo com Kurose e Ross (2006), “um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento”.

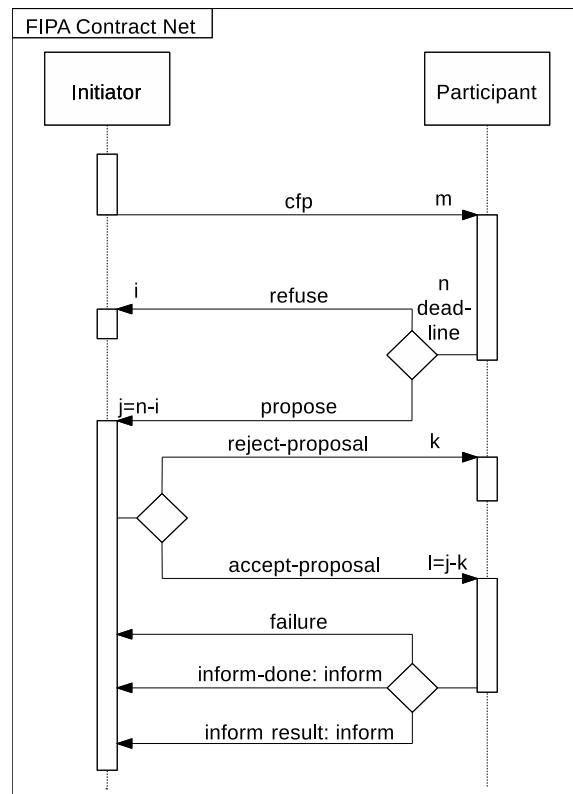
Entre os padrões FIPA estão as definições de protocolos de comunicação entre agentes. Os protocolos de comunicação FIPA são uma sequência predeterminada de mensagens para estabelecer comunicação entre dois ou mais agentes que têm um determinado objetivo. Este objetivo pode ser o envio de uma requisição, uma negociação ou uma maneira de monitorar o comportamento de um agente. Os protocolos definidos pela FIPA são:

- FIPA Request Interaction Protocol Specification (FIPA00026, 2004);
- FIPA Query Interaction Protocol Specification (FIPA00027, 2004);
- FIPA Request When Interaction Protocol Specification (FIPA00028, 2004);
- FIPA Contract Net Interaction Protocol Specification (FIPA00029, 2004);
- FIPA Iterated Contract Net Interaction Protocol Specification (FIPA00030, 2004);
- FIPA Brokering Interaction Protocol Specification (FIPA00033, 2004);
- FIPA Recruiting Interaction Protocol Specification (FIPA00034, 2004);
- FIPA Subscribe Interaction Protocol Specification (FIPA00035, 2004);
- FIPA Propose Interaction Protocol Specification (FIPA00036, 2004).

Os protocolos são inicializados quando algum dos agentes que possui um protocolo implementado recebe uma mensagem com as restrições especificadas nos filtros de mensagens passados como parâmetros nas funções que implementam os protocolos. Para exemplificar

a implementação de um protocolo FIPA será utilizado o protocolo FIPA ContractNet que tem sua estrutura exibida na Figura 2.7.

Figura 2.7 – Troca de mensagens entre agentes que implementam o protocolo FIPA ContractNet.



(FIPA00029, 2004)

No diagrama esquemático da Figura 2.7 é mostrada a troca de mensagens entre dois tipos de agentes, agente iniciante e agente participante.

Uma sequência de mensagens é trocada em ordem pré-estabelecida, ou seja, após o início da troca de mensagens, para cada mensagem recebida uma outra mensagem é preparada e reenviada para o outro agente, até que se chegue ao fim do protocolo.

De modo bem simplificado, no protocolo FIPA ContractNet, as seguintes ações podem ser identificadas:

- o agente iniciante inicia o protocolo, enviando mensagens do tipo *call for proposes* (CFP) para outros agentes que podem vir a atender sua solicitação;
- os agentes participantes recebem a mensagem CFP e podem, ou enviar uma mensagem do tipo REFUSE se recusando a ajudá-lo, ou enviar uma mensagem PROPOSE com uma proposta para o agente iniciante;
- o agente iniciante recebe as propostas enviadas pelos agentes participantes e analisa cada uma delas, enviando, mensagens REJECT-PROPOSAL, recusando a proposta, ou mensagens ACCEPT-PROPOSAL, aceitando a proposta.

- os agentes participantes que tiveram suas propostas aceitas, executam as ações que foram prometidas ao agente iniciante e em seguida enviam mensagens INFORM caso tenham sido bem sucedidos em suas tarefas ou FAILURE caso não tenham conseguido cumprir o que prometeram.

## 2.6 Conclusão

De acordo com (WOOLDRIDGE, 2009) a utilização de sistemas multiagentes é adequada nas seguintes situações:

- ambientes muito dinâmicos e complexos ou com incerteza associada, neste tipo de situação SMA podem ser uma boa solução;
- ambientes que podem ser modelados como sociedades de agentes, que cooperam entre si para atingir determinado objetivo, ou ainda, que podem competir para obter melhores resultados;
- ambientes que apresentam recursos distribuídos, como: dados sendo aquisitados por elementos sensores remotos ou dispositivos de controle distribuídos. Para estes casos uma abordagem centralizada dificulta o gerenciamento dos dados ou torna inviável sua implementação;
- ambientes que apresentam sistemas legados que são importantes e de difícil substituição. Para estes casos uma camada multiagentes pode ser implementada, permitindo a interação desta parte do sistema com os demais componentes de software.

Por se tratar o sistema de automação do SED de um conjunto de dispositivos que realizam aquisição de dados e que têm a possibilidade de receber e enviar comandos, a abordagem de sistemas multiagentes se torna uma opção viável e conveniente, que se enquadra nas características do terceiro item da lista de situações citada por (WOOLDRIDGE, 2009).



---

## Capítulo 3

# PADE: Plataforma para Desenvolvimento e Execução de SMA

---

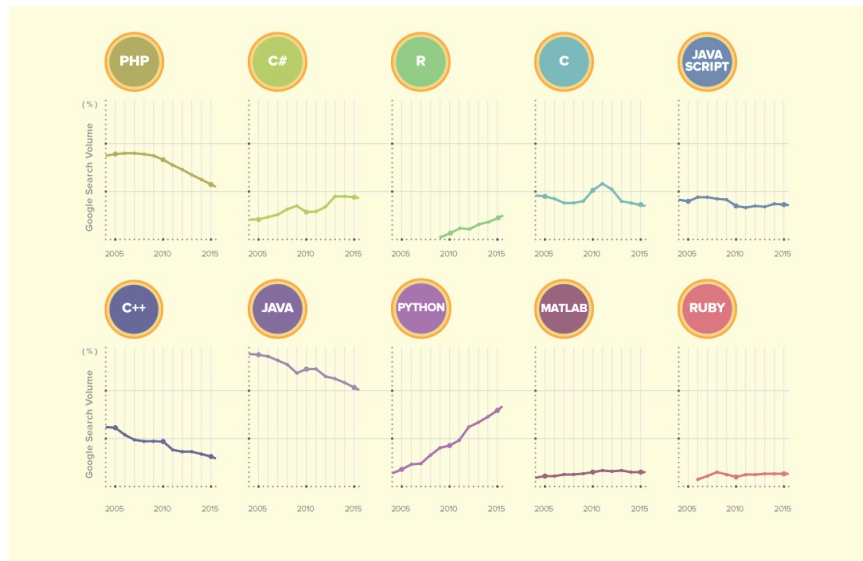
### 3.1 Introdução

Neste capítulo é mostrado o desenvolvimento de um dos objetivos específicos deste trabalho, definidos no Capítulo 1, um framework para desenvolvimento de sistemas multiagentes: o Python Agent DEvelopment framework (PADE), que permite a implementação, execução e monitoramento de um SMA em ambiente distribuído. O PADE foi desenvolvido inteiramente em linguagem de programação Python, utilizando uma biblioteca para construção de sistemas distribuídos, o Twisted. Entre muitas vantagens PADE possui uma abordagem clara e fácil de tornar difíceis implementações em realidade com apenas algumas linhas de código, fazendo uso da principal funcionalidade do Python: simplicidade, além de prover acesso facilitado entre agentes sendo executados em diferentes dispositivos.

### 3.2 A escolha da Linguagem de Programação Python

A linguagem de programação Python vem despertando crescente interesse, tanto de programadores experientes como iniciantes. De acordo com a Figura 3.1, onde são mostrados gráficos a respeito da quantidade de buscas por materiais introdutórios às linguagens de programação mais conhecidas, realizadas no Google entre 2005 e 2015, existe uma forte tendência de crescimento da quantidade de programadores que utilizam Python,

Figura 3.1 – Gráficos de quantidade de buscas por materiais introdutórios às linguagens de programação mais conhecidas no Google entre 2005 e 2015.



Fonte: Adaptado de (KERANEN, 2015).

em relação as demais linguagens.

Python é uma linguagem de programação essencialmente orientada a objetos, mas pode ser facilmente utilizada para programação estruturada ou para programação funcional. Python possui algumas características importantes, que estimularam sua adoção.

- Python é software livre;
- Python é uma linguagem de programação madura, criada em 1992;
- Python é multiplataforma;
- Python já é componente padrão de muitas das distribuições Linux;
- Python possui diversas ferramentas livres e de qualidade que proporcionam maior agilidade no desenvolvimento de software, tal como, IDE com auto-completação de código, destaque de sintaxe, correção de erros automática e debugadores;
- Python vem sendo adotada massivamente pela comunidade científica, principalmente por conta de suas facilidades de manipulação de matrizes e link direto com scripts executados em C;
- Python é uma das linguagens mais utilizadas para o aprendizado de programação, pois possui uma sintaxe clara, concisa e organizada.
- Python possui estruturas de dados padrões da linguagem que oferecem fácil organização dos elementos a serem manipulados, como por exemplo: listas, dicionários, tuplas e conjuntos;

- Python já vem com uma grande variedade de módulos que permitem o desenvolvimento fácil e rápido de qualquer tipo de aplicação seja ela o *backend* de um sistema web até um algoritmo de *machine learn* para identificação de padrões em imagens;
- Existem módulos Python, que não são padrões da linguagem, disponíveis para quase tudo que se pretenda fazer. Esses módulos são implementados e mantidos pela comunidade de desenvolvedores, e facilmente instaláveis por meio do *Python Package Installer* (PIP), bastando apenas uma conexão com a internet para isso;
- Python possui uma comunidade ativa de desenvolvedores e de usuários que compartilham informação na internet e respondem dúvidas quanto à utilização da linguagem e das bibliotecas desenvolvidas para a linguagem.

### 3.3 Desenvolvimento da Plataforma

De acordo com Bellifemine, Caire e Greenwood (2007), essencialmente, um framework para o desenvolvimento de SMA deve oferecer três funcionalidades:

- Uma biblioteca para a construção de agentes que implemente os padrões de interoperabilidade definidos pela FIPA;
- Um ambiente de execução distribuído para que os agentes possam se comunicar e realizar suas atividades, mesmo estando em diferentes máquinas ou plataformas;
- Um ambiente gráfico ou em linha de código que permita o monitoramento e controle das atividades dos agentes em execução;

Essas funcionalidades foram criadas para compor um framework de desenvolvimento de SMA em Python. O framework chama-se PADE e foi contruído utilizando as ferramentas disponíveis no framework Twisted, uma biblioteca escrita em Python utilizada para implementação de sistemas distribuídos. Twisted é software livre licenciado sob os termos da licença MIT.

#### 3.3.1 O Framework Twisted

Twisted é um mecanismo de rede, assíncrono, baseado em eventos e escrito em Python. Twisted tem suporte aos protocolos mais comuns das camadas de transporte e aplicação da internet como: TCP, UDP, HTTP, SSH, IMAP, FTP, entre outros. Twisted implementa versões de cliente e servidor destes protocolos, assim como oferece ferramentas que facilitam sua manipulação e configuração (MCKELLAR; FETTIG, 2013).

Twisted também oferece ferramentas de baixo nível para que protocolos genéricos sejam implementados na camada de aplicação de maneira fácil e rápida. Foram estas as ferramentas utilizadas no desenvolvimento do framework PADE.

### 3.3.1.1 Programação Assíncrona Orientada a Eventos

Quando sistemas distribuídos são desenvolvidos baseados no paradigma de programação síncrona, existem duas escolhas para o desenvolvedor: não tratar requisições externas enquanto um evento é processado ou desenvolver a aplicação fazendo uso de *threads*, que por vezes pode introduzir complexidade extra ao software.

No paradigma de programação assíncrona, o software desenvolvido continua podendo receber e tratar múltiplas conexões, mesmo enquanto está processando eventos anteriores, sem a necessidade de ter que controlar *threads*. Dessa forma, o paradigma de programação assíncrona contrasta com os outros dois possíveis tipos de programação, o *single-thread* síncrono e o *multi-thread* (MCKELLAR; FETTIG, 2013):

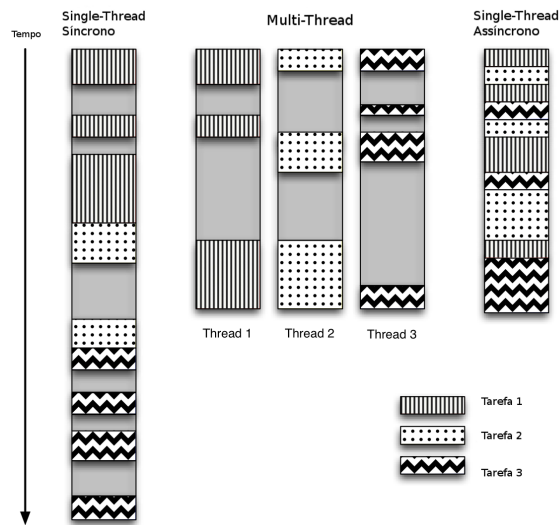
- Single-thread síncrono: Neste paradigma de programação as ações realizadas pelo programa são executadas de forma serial. Se alguma tarefa for bloqueada, todas as outras tarefas do programa precisarão esperar o encerramento desta tarefa para que possam ser executadas. Este tipo de programação é o mais simples de ser implementado, mas para alguns casos pode sofrer com problemas de lentidão, que tornaria este tipo de implementação impraticável.
- Multi-thread: Neste paradigma de programação as tarefas de um determinado programa são executadas paralelamente por threads de controle diferentes. Isto permite que o software continue tratando eventos, mesmo com o bloqueio de uma determinada tarefa. No entanto, construir blocos de código que consumam recursos paralelamente de hardware pode não ser uma tarefa muito simples, o que aumenta bastante a complexidade do código a ser escrito.

No paradigma single-thread assíncrono as tarefas de um determinado programa são executadas intercaladamente em um único thread de controle. Esse paradigma combina as vantagens de execução de programas com múltiplas *threads*, com a facilidade de implementação de um programa de *thread* única.

Na Figura 3.2 é ilustrada a execução de um programa que precisa executar três tarefas, utilizando cada um dos paradigmas de programação citados.

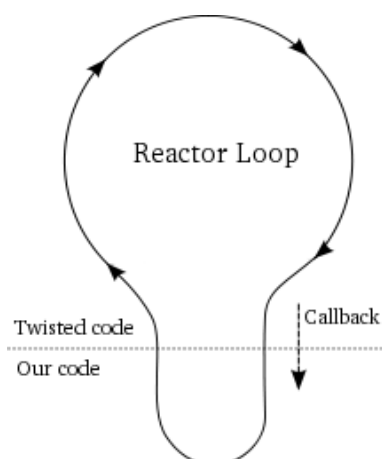
Já o conceito de programação orientada a eventos (assíncrono), refere-se à característica do fluxo do programa ser determinado por eventos externos. Esse tipo de paradigma utiliza sempre um loop de eventos (*reactor loop*) assim como métodos que são chamados quando um evento acontece (*callbacks*).

Figura 3.2 – Paradigmas de programação



Fonte: Adaptado de (MCKELLAR; FETTIG, 2013)

Na Figura 3.3 é mostrada uma representação do loop de eventos Twisted e da chamada de uma *callback* quando um evento é processado. Fica claro nesta figura que o desenvolvedor da aplicação que faz uso do Twisted não precisa se preocupar em gerenciar o loop de eventos, o gerenciamento fica a cargo da própria biblioteca. Assim, a única preocupação de quem desenvolve uma aplicação com Twisted é escrever *callbacks* para processar diferentes tipos de eventos.

Figura 3.3 – Loop de eventos do Twisted e chamada de uma *callback*.

Fonte: (MCKELLAR; FETTIG, 2013).

### 3.3.1.2 Componentes Básicos do Twisted

O Twisted possui alguns componentes básicos que fazem parte de sua arquitetura de funcionamento. Para desenvolver qualquer aplicação com Twisted é necessário conhecer e compreender as funcionalidades desses componentes. Os principais componentes do Twisted são (MCKELLAR; FETTIG, 2013):

- **Reactor:** É o núcleo do Twisted, chamado também de *loop* principal. Este loop supervisiona os eventos que decorrem de chamadas de protocolos de rede, de temporização ou de alteração na estrutura do sistema de arquivos. Na ocorrência de qualquer um desses eventos, o reactor os direciona para o método responsável por tratá-los;
- **Transport:** Este componente fornece as interfaces necessárias para se estabelecer conexão entre dois pontos comunicantes em rede;
- **Protocols:** Twisted fornece interfaces que implementam como eventos relacionados a diversos protocolos da camada de aplicação devem ser tratados, como por exemplo, chamadas HTTP, Telnet, DNP, FTP, IMAP, entre outros. Mas, também, Twisted oferece as ferramentas necessárias para que protocolos customizados sejam construídos;
- **Protocol Factories:** Twisted instancia um objeto *Protocol* sempre que, por exemplo, uma nova conexão é realizada com um servidor. No entanto, esta instancia é perdida assim que a conexão é encerrada, dessa forma todas as informações desta conexão são perdidas. Com o intuito de armazenar informações permanentes no protocolo, Twisted implementa o componente *Protocol Factories* de modo que informações pertinentes, como, por exemplo, quantos clientes estão conectados no servidor, sejam mantidas, mesmo depois que estas conexões sejam encerradas.

### 3.3.1.3 Exemplos de Execução de Aplicações com Twisted

Por meio dos Algoritmo 3.1, Algoritmo 3.2 e Algoritmo 3.3 é possível mostrar a implementação, respectivamente, de um servidor Twisted que retorna o dado recebido; um cliente que envia uma mensagem para o servidor, exibe a resposta e em seguida encerra a conexão; e as linhas de código necessárias para executar estes dois comportamentos (servidor e cliente) em um terminal Linux.

## 3.3.2 O Framework PADE

Utilizando como base de implementação o framework Twisted, foi desenvolvido o PADE, um framework para desenvolvimento, gerenciamento e execução de SMA em Python. O PADE segue alguns dos padrões estabelecidos pela FIPA, que visam interoperabilidade

**Algoritmo 3.1:** Código de servidor de eco em Python utilizando Twisted

```
1 from twisted.internet import protocol, reactor
2
3 class Echo(protocol.Protocol):
4     def dataReceived(self, data):
5         self.transport.write(data)
6
7 class EchoFactory(protocol.Factory):
8     def buildProtocol(self, addr):
9         return Echo()
10
11 reactor.listenTCP(8000, EchoFactory())
12 reactor.run()
```

**Algoritmo 3.2:** Código de cliente em Python utilizando Twisted

```
1 from twisted.internet import reactor, protocol
2
3 class EchoClient(protocol.Protocol):
4     def connectionMade(self):
5         self.transport.write('Hello, world!')
6     def dataReceived(self, data):
7         print 'Server said:', data
8         self.transportloseConnection()
9
10 class EchoFactory(protocol.ClientFactory):
11     def buildProtocol(self, addr):
12         return EchoClient()
13
14     def clientConnectionFailed(self, connector, reason):
15         print 'Connection failed.'
16         reactor.stop()
17
18     def clientConnectionLost(self, connector, reason):
19         print 'Connection lost.'
20         reactor.stop()
21
22 reactor.connectTCP("localhost", 8000, EchoFactory())
23 reactor.run()
```

entre SMA desenvolvidos em diferentes plataformas, mas não se detém aos seus detalhes mais específicos, buscando uma abordagem de comunicação mais direta e objetiva.

**3.3.2.1 Estrutura de Arquivos do PADE**

O framework PADE é composto por sete módulos principais que agrupam suas funcionalidades. A seguir a descrição de cada um dos módulos PADE:

- **acl:** implementa alguns dos padrões FIPA de comunicação entre agentes, como por

**Algoritmo 3.3:** Código para execução em terminal linux dos algoritmos de servidor e cliente.

```
1      $ python echoserver.py # No Terminal 1
2      $ python echoclient.py # No Terminal 2
3      Server said: Hello , world!
4      Connection lost .
```

exemplo a possibilidade de composição de mensagens no padrão FIPA-ACL, definidos em (FIPA00061, 2004; FIPA00070, 2004; FIPA00071, 2004).

- **behaviours:** implementa alguns dos comportamentos e protocolos definidos e padronizados pela FIPA para execução pelos agentes, (FIPA00026, 2004; FIPA00029, 2004; FIPA00035, 2004);
- **core:** principal módulo do PADE, onde são implementados os algoritmos de cliente e servidor dos agentes por meio do framework Twisted. É nesse módulo que estão implementadas as funções de execução de agentes definidos pelo usuário e dos agentes padrões: AMS e Sniffer, definidos em (FIPA00023, 2004);
- **db:** módulo que implementa a comunicação do núcleo do framework PADE com um banco de dados. Por meio deste módulo o framework pode, por exemplo, armazenar todas as mensagens trocadas pelos agentes;
- **examples:** apresenta alguns exemplos simples de construção e execução de agentes;
- **gui:** onde estão implementadas as classes que definem a interface gráfica do AMS;
- **misc:** módulo onde estão implementadas algumas funcionalidades de uso geral, como, por exemplo, exibir mensagens de forma padronizada na tela;
- **tests:** módulo de testes, a ser implementado futuramente.

Na Figura 3.4 pode ser observada a estrutura de arquivos que compõem o código fonte do framework PADE.

### 3.3.2.2 Filosofia de Execução dos Agentes no Framework PADE

O núcleo do framework PADE consiste na funcionalidade de execução dos agentes e está descrito Figura 3.5 em forma de diagrama de classes do padrão UML. Na Figura é possível visualizar a classe **Agent** e suas relações com as demais classes no módulo **core**.

Todo agente desenvolvido com os módulos disponibilizados pelo framework deve herdar da classe **Agent**. Esta classe implementa um protocolo definido pelas bibliotecas do Twisted, e representados na Figura 3.5 por meio das classes **AgentProtocol** e **AgentFactory**. Essa implementação permite que um agente comporte-se horas como cliente, horas como



Figura 3.4 – Estrutura de arquivos do PADE.

```

Pade/
  |---acl/
  |       |---__init__.py
  |       |---aid.py
  |       |---filters.py
  |       |---messages.py
  |---behaviours/
  |           |---__init__.py
  |           |---protocols.py
  |---core/
  |       |---__init__.py
  |       |---agent.py
  |       |---ams.py
  |       |---sniffer.py
  |---db/
  |       |---database_access.py
  |---doc/
  |---examples/
  |---gui/
  |       |---__init__.py
  |       |---gui.py
  |---misc/
  |       |---__init__.py
  |       |---common.py
  |       |---utility.py
  |---tests/

```

Fonte: Própria do autor.

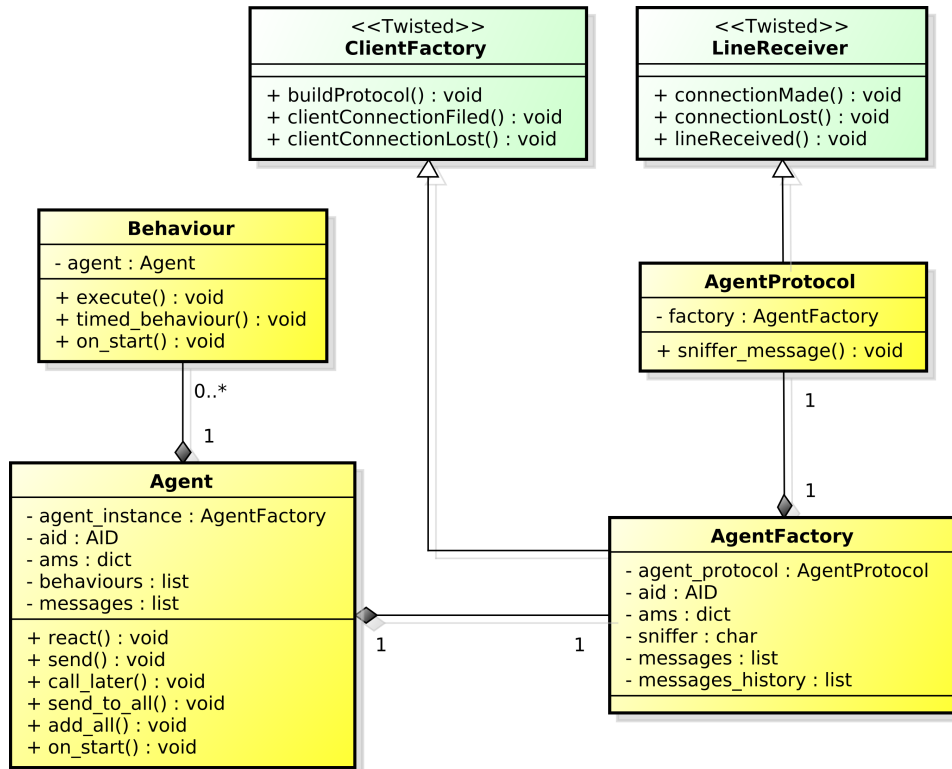
servidor, ou seja, um agente é um nó comunicante na rede, que pode iniciar uma troca de mensagens, mas que também está disponível para responder solicitações de outros nós da rede, bastando que para isso os agentes conheçam os endereços uns dos outros.

Um agente na plataforma é identificado pelo seu AID *agent Identifier* que tem a seguinte composição: [nome local]@[endereço IP do agente] : [porta]. Um exemplo de AID para um agente de nome `alimentador_2115` que é executado na máquina de IP 192.168.1.2 na porta 5002, seria: `alimentador_2115@192.168.1.2:5001`.

### 3.3.2.3 O Agente AMS

O AMS (Agent Management System) desempenha função muito importante para a plataforma de SMA e de acordo com o padrão (FIPA00023, 2004), que define as características e atribuições do AMS, é mandatória a presença de um gerenciador na plataforma.

Figura 3.5 – Estrutura de classes do framework PADE no padrão UML.



Fonte: Própria do autor.

No PADE, o AMS exerce funções de controle e supervisão e mantém uma tabela que contém os identificadores dos agentes. De acordo ainda com a FIPA, os agentes precisam registrar-se no AMS para adquirirem um identificador válido, sendo assim capazes de se comunicar com outros agentes.

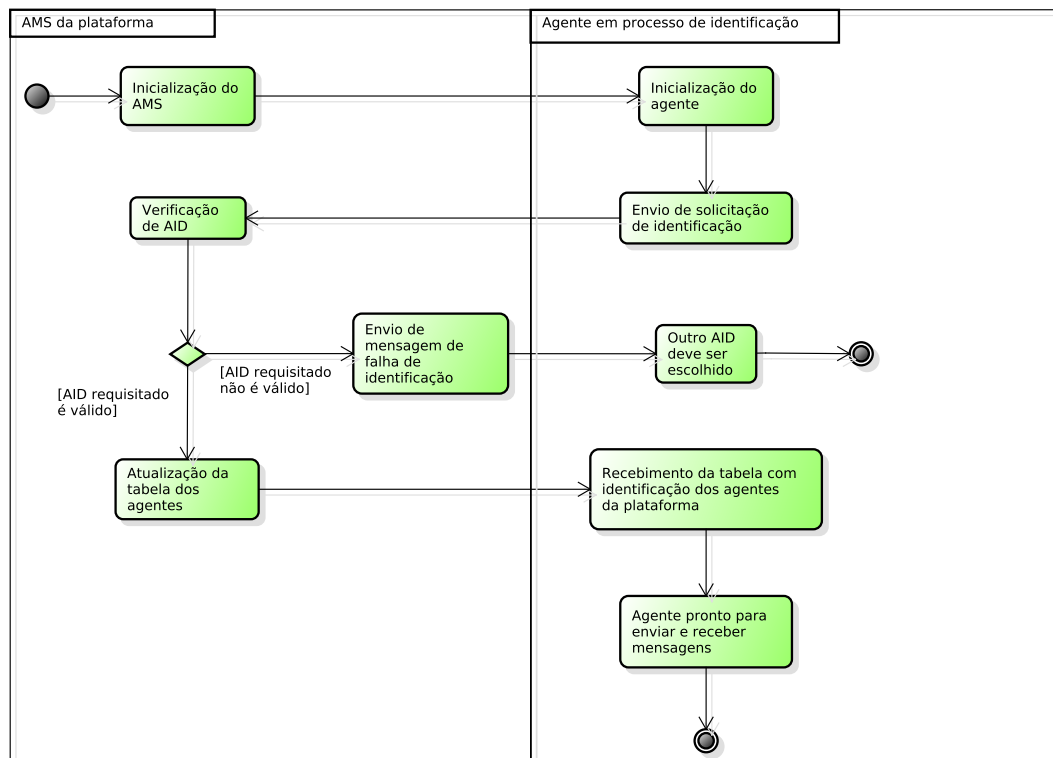
No PADE, o AMS é um agente, e necessariamente deve ser o primeiro agente a ser lançado, já que os outros precisam do AMS para se cadastrarem.

Cada agente lançado na plataforma deve enviar mensagem ao agente AMS identificandose e assim ficando visível aos demais agentes. O processo de um agente que solicita identificação e do AMS que recebe e analisa o pedido pode ser representado por meio de um diagrama de atividades conforme mostrado na Figura 3.6.

O diagrama mostrado na Figura 3.6 descreve os comportamentos de:

- solicitação de identificação: o agente que acabou conectar-se à rede de comunicação envia seu AID ao AMS solicitando identificação e permissão para entrar na plataforma de agentes;
- análise de solicitação: o agente AMS recebe a solicitação do agente e verifica se o AID enviado já existe entre os AID dos agentes ativos na plataforma. Se nenhum dos agentes ativos estiver cadastrado com o AID solicitado, então o AMS envia mensagem de autorização para entrada do agente.

Figura 3.6 – Diagramas atividades do processo de identificação no PADE.



Fonte: Própria do autor.

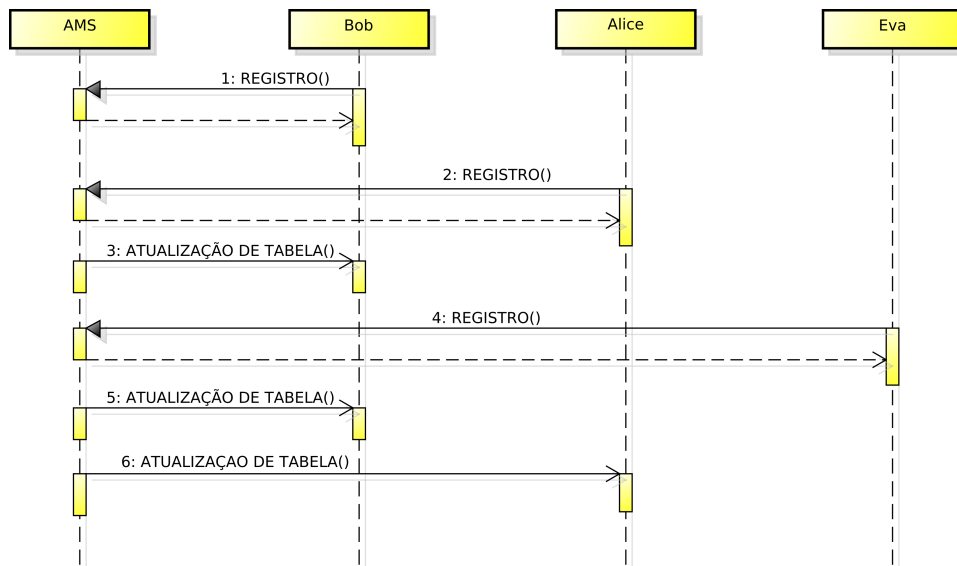
- resposta da solicitação: caso tenha seu pedido aceito, o agente recebe uma tabela com os endereços de todos os agentes presentes na plataforma e pode comunicar-se com cada um deles. Caso tenha seu pedido de identificação negado, o agente deve enviar uma nova proposta com AID diferente para o AMS.

No PADE, assim como o AMS, cada agente possui uma tabela com o nome e o endereço de cada agente em execução na plataforma. Essa tabela é distribuída e atualizada pelo AMS sempre que um agente entra ou sai da plataforma. Essa situação pode ficar bem clara por meio de um exemplo.

Os agentes Bob, Eva e Alice, inicialmente estão fora da plataforma PADE. Primeiro o agente Bob é executado. Bob se registra junto ao AMS que valida seu AID. Como não existe nenhum outro agente em execução na plataforma, nenhuma tabela é criada ou atualizada. Em seguida o agente Alice entra em execução registrando-se junto ao AMS, que valida seu AID, atualiza sua tabela de agentes ativos e distribui esta tabela aos demais agentes da rede, no caso para o agente Bob. Da mesma forma, o agente Eva também se registra junto ao AMS, que atualiza as tabelas de todos os agentes (Bob e Alice) com o novo agente disponível para comunicação. Essa situação é mostrada na Figura 3.7.

Dessa forma quando um agente deseja se comunicar com outro, busca em sua própria tabela o endereço do agente destino, sem precisar perguntar o endereço ao agente AMS, e mesmo que o agente AMS seja desativado, os agentes ainda serão capazes de se comunicar

Figura 3.7 – Registro e atualização de tabela de agentes.



Fonte: Própria do autor.

pois possuem a tabela de endereços dos agentes permanentes na rede.

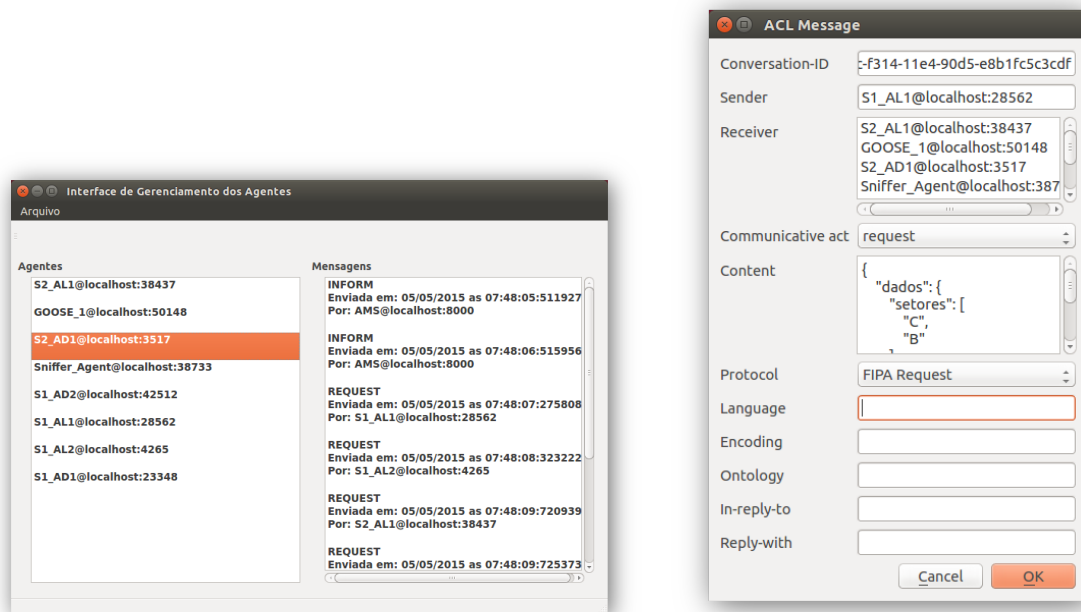
Logicamente sem a presença de um agente AMS na plataforma, as funções de supervisão, registro e controle ficam desprovidas pela plataforma. Mas a comunicação dos agentes que já estavam na plataforma continua possível, o que em muitos casos pode ser útil, no entanto esta deve ser uma situação temporária, uma vez que depois que o agente AMS é desativado, a tabela de agentes não poderá mais ser atualizada.

### 3.3.2.4 O Agente Sniffer

Outro componente muito útil em uma plataforma multiagente é o agente Sniffer. No PADE, o agente Sniffer tem a função de enviar mensagens periódicas aos agentes para realizar testes de conexão. Caso o agente não responda à mensagem, é informado ao AMS que o agente não está mais na plataforma, e assim a tabela dos outros agentes é atualizada.

Outra função que é executada pelo agente Sniffer é o registro das conversas realizadas pelos agentes. Essas mensagens tanto podem ser armazenadas em um arquivo de log como podem ser exibidas em uma interface gráfica. Na Figura 3.8a é mostrada a janela principal da interface gráfica do agente Sniffer, que exibe na coluna da esquerda todos os agentes que estão ativos e que possuem cadastro junto ao AMS e na coluna da direita as mensagens recebidas pelo agente selecionado. Na Figura 3.8b é mostrada uma janela secundária da interface do agente Sniffer, que é lançada ao selecionar alguma das mensagens exibidas na janela principal. Nesta janela são mostrados todos os campos da mensagem no padrão ACLMessage recebida pelo agente selecionado.

Figura 3.8 – Interface gráfica do agente Sniffer do PADE.



(a) Mensagens recebidas pelos agentes.

(b) Detalhamento da mensagem recebida.

Fonte: Própria do autor.

### 3.3.2.5 Estrutura de um Agente Modelado no PADE

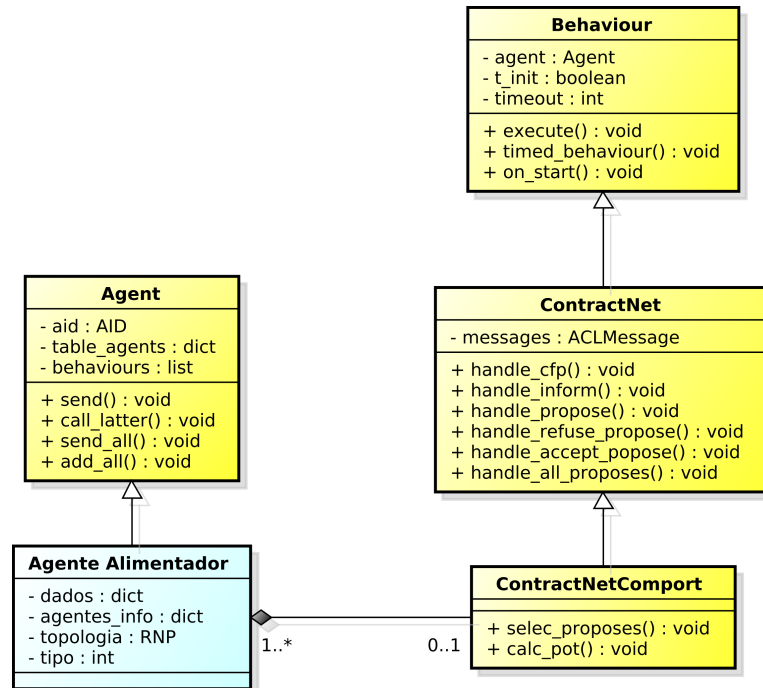
A estrutura de código de um agente modelado por meio do framework PADE é bem simples e tem um padrão definido. Este padrão consiste em:

- classes que herdam de um comportamento/protocolo disponibilizado pelo PADE;
- uma classe que herda da classe **Agent**.

Dessa forma, um arquivo que modele um agente no PADE sempre terá uma única classe que representa a instância do agente, e quantas classes forem necessárias para representar instâncias dos comportamentos definidos para os agentes.

Por exemplo, para implementar um comportamento do tipo Contractnet Iniciante no PADE, o diagrama de classes UML mostrado na Figura 3.9 pode ser utilizada. No diagrama UML é possível observar que a classe **AgenteAlimentador** herda da classe **Agent** todas as características necessárias para que o agente possa ser executado, identificado e em seguida consiga estabelecer comunicação com os demais. A classe **ContractNetComport** herda da classe **ContractNet** todos os métodos necessários para que o protocolo FIPA ContractNet seja implementado, podendo ser do tipo iniciante ou participante. Por fim, existe uma associação do tipo composição entre as classes **AgenteAlimentador** e **ContractNetComport** que indica que um objeto do tipo **AgenteAlimentador** pode conter um ou vários objetos **ContractNetComport**.

Figura 3.9 – Diagrama de classes UML de um agente que tem um comportamento FIPA ContractNet associado.



Fonte: Própria do autor.

### 3.4 Exemplo de Implementação de Agentes no PADE

No Algoritmo 3.4 é mostrado o código de dois agentes implementados utilizando as bibliotecas disponibilizadas pelo PADE. O primeiro agente é chamado de agente Chave e possui um comportamento FIPA Request participante associado enquanto outro agente é chamado de agente Operador e possui um comportamento FIPA Request iniciante.

O código do Algoritmo 3.4 implementa o envio de uma solicitação que o agente Operador faz para que o agente Chave possa abrir a chave que está sob seu comando. A classe **AgenteChave** define o agente Chave, está indicada na linha 38, e é associada à classe **ComportAgenteChave** que implementa o protocolo FIPA Request na linha 11.

A classe **AgenteOperador**, na linha 49, define o agente Operador e o associa ao comportamento definido na classe **ComportAgenteOperador** que implementa o protocolo FIPA Request na linha 26.

A partir da linha 70, está o procedimento de instanciação das classes dos agentes, configuração do endereço do agente AMS da plataforma, no método `set_ams()`, e inicialização da plataforma no método `start_loop()`.

As mensagens de inicialização da plataforma e dos agentes Chave e Operador, conforme programados no Algoritmo 3.4 são exibidas na Figura 3.10.

**Algoritmo 3.4:** Definição de dois agentes: Agente Chave e Agente Operador com seus respectivos comportamentos e código de inicialização

```

1  #!/coding=utf-8
2
3  from pade.behaviours.protocols import FipaRequestProtocol
4  from pade.misc.utility import display_message
5  from pade.misc.common import set_ams, start_loop
6  from pade.core.agent import Agent
7  from pade.acl.messages import ACLMessage
8  from pade.acl.aid import AID
9
10 import loremipsum
11
12
13 class ComportAgenteChave(FipaRequestProtocol):
14     """Comportamento do Agente Chave"""
15
16     def __init__(self, agent):
17         super(ComportAgenteChave, self).__init__(
18             agent=agent, message=None, is_initiator=False)
19
20     def handle_request(self, message):
21         display_message(self.agent.aid.localname, 'Mensagem Request
22             recebida')
23         message = message.create_reply()
24         message.set_performative(ACLMessage.INFORM)
25         message.set_content('Operacao realizada')
26         self.agent.send(message)
27
28 class ComportAgenteOperador(FipaRequestProtocol):
29     """Comportamento do Agente Operador"""
30
31     def __init__(self, agent, message):
32         super(ComportAgenteOperador, self).__init__(
33             agent=agent, message=message, is_initiator=True)
34         self.agent = agent
35
36     def handle_inform(self, message):
37         display_message(self.agent.aid.localname, 'Mensagem Inform recebida
38             ')
39
40 class AgenteChave(Agent):
41     """Definicao do Agente Chave"""
42
43     def __init__(self, aid):
44         super(AgenteChave, self).__init__(aid=aid, debug=False)
45
46         comp = ComportAgenteChave(self)
47
48         self.behaviours.append(comp)

```

```

51 class AgenteOperador(Agent):
52     """Definicao do Agente Operador"""
53
54     def __init__(self, aid):
55         super(AgenteOperador, self).__init__(aid=aid, debug=False)
56
57         message = ACLMessage(ACLMessage.REQUEST)
58         message.set_protocol(ACLMessage.FIPA_REQUEST_PROTOCOL)
59         message.add_receiver(AID('agente_chave'))
60
61         s = loremipsum.get_paragraphs(40)
62         ss = ''
63         for i in s:
64             ss += i
65
66         message.set_content(ss)
67
68         comp = ComportAgenteOperador(self, message)
69
70         self.behaviours.append(comp)
71         comp.on_start()
72
73 if __name__ == '__main__':
74
75     set_ams('localhost', 8000, debug=False)
76
77     agentes = list()
78
79     agente_chave = AgenteChave(AID(name='agente_chave'))
80     agente_chave.ams = {'name': 'localhost', 'port': 8000}
81
82     agente_operador = AgenteOperador(AID(name='agente_operador'))
83     agente_operador.ams = {'name': 'localhost', 'port': 8000}
84
85     agentes.append(agente_chave)
86     agentes.append(agente_operador)
87
88     start_loop(agentes, gui=True)

```

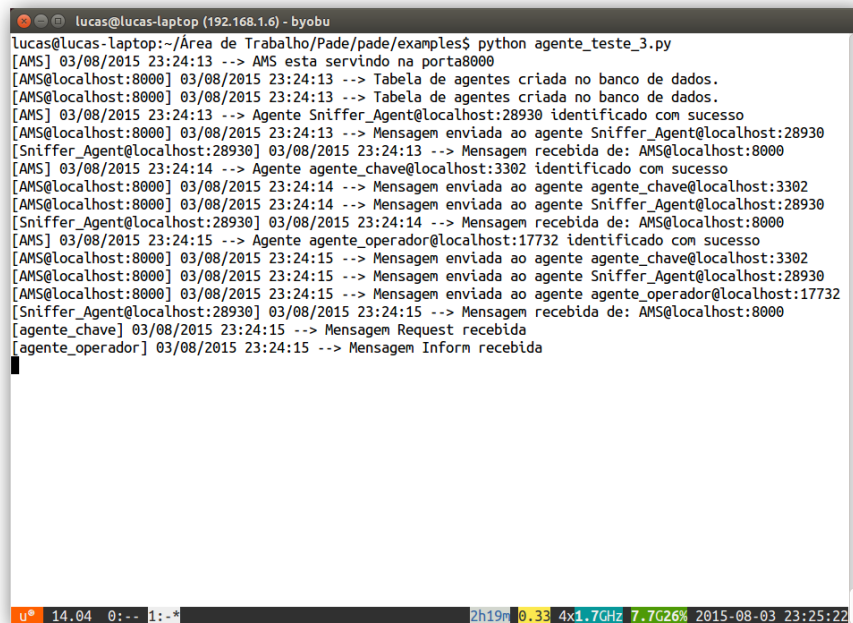
## 3.5 Conclusão

Ter controle total e conhecer todos os mecanismos disponíveis das ferramentas com que se trabalha é de extrema importância para o desenvolvimento de qualquer tipo de trabalho. É neste contexto que o PADE se apresenta como uma alternativa às plataformas de desenvolvimento e execução de agentes já existentes, além de possibilitar que o SMRA possa ser implementado em uma linguagem de programação moderna, de fácil aprendizado e com muitos recursos para o desenvolvimento de sistemas distribuídos, que é o Python.

O PADE procura utilizar muitos dos aspectos recomendados pela FIPA na implementação de plataformas multiagentes, mas, por vezes, fazendo uso de uma abordagem mais direta e objetiva, sem seguir o padrão em seus mínimos detalhes. Isso confere ao PADE a



Figura 3.10 – Mensagens exibidas pelos agentes chave e Operador.



```
lucas@lucas-laptop:~/Área de Trabalho/Pade/pade/examples$ python agente_teste_3.py
[AMS] 03/08/2015 23:24:13 --> AMS esta servindo na porta8000
[AMS@localhost:8000] 03/08/2015 23:24:13 --> Tabela de agentes criada no banco de dados.
[AMS@localhost:8000] 03/08/2015 23:24:13 --> Tabela de agentes criada no banco de dados.
[AMS] 03/08/2015 23:24:13 --> Agente Sniffer_Agent@localhost:28930 identificado com sucesso
[AMS@localhost:8000] 03/08/2015 23:24:13 --> Mensagem enviada ao agente Sniffer_Agent@localhost:28930
[Sniffer_Agent@localhost:28930] 03/08/2015 23:24:13 --> Mensagem recebida de: AMS@localhost:8000
[AMS] 03/08/2015 23:24:14 --> Agente agente_chave@localhost:3302 identificado com sucesso
[AMS@localhost:8000] 03/08/2015 23:24:14 --> Mensagem enviada ao agente agente_chave@localhost:3302
[AMS@localhost:8000] 03/08/2015 23:24:14 --> Mensagem enviada ao agente Sniffer_Agent@localhost:28930
[Sniffer_Agent@localhost:28930] 03/08/2015 23:24:14 --> Mensagem recebida de: AMS@localhost:8000
[AMS] 03/08/2015 23:24:15 --> Agente agente_operador@localhost:17732 identificado com sucesso
[AMS@localhost:8000] 03/08/2015 23:24:15 --> Mensagem enviada ao agente agente_chave@localhost:3302
[AMS@localhost:8000] 03/08/2015 23:24:15 --> Mensagem enviada ao agente Sniffer_Agent@localhost:28930
[AMS@localhost:8000] 03/08/2015 23:24:15 --> Mensagem enviada ao agente agente_operador@localhost:17732
[Sniffer_Agent@localhost:28930] 03/08/2015 23:24:15 --> Mensagem recebida de: AMS@localhost:8000
[agente_chave] 03/08/2015 23:24:15 --> Mensagem Request recebida
[agente_operador] 03/08/2015 23:24:15 --> Mensagem Inform recebida
```

Fonte: Própria do autor.

característica de simplicidade de implementação e de execução.

Alguns dos aspectos disponibilizados pelo PADE para o desenvolvimento e execução de agentes, são:

- Definição do padrão FIPA-ACL de mensagens entre agentes;
- Definição de alguns dos protocolos definidos pela FIPA para comunicação entre agentes: FIPA-Request, FIPA-ContractNet, FIPA-Subscribe;
- Definição de uma estrutura de classes que organizam a implementação dos comportamentos dos agentes;
- interface gráfica para a visualização de troca de mensagens entre agentes;
- Execução dos agentes em um loop de eventos assíncrono de thread única;
- Disponibilização de uma plataforma de execução dos agentes em ambiente distribuído de comunicação.

---

# Capítulo 4

## Sistema de Recomposição Automático

---

O sistema elétrico de potência (SEP), geralmente é sub-dividido em três grandes segmentos, cada um deles com uma função bem definida. A primeira parte é a geração, responsável por transformar a energia, seja ela mecânica, térmica ou química em energia elétrica. Como na maioria das vezes não é possível gerar energia próximo dos grandes centros consumidores, por razões de recursos naturais, ou mesmo de logística, a energia gerada precisa ser transportada por longas distâncias, até chegar aos grandes centros consumidores. Assim a segunda parte do SEP é a transmissão. Como é necessário transportar a energia por longas distâncias, a tensão é elevada a níveis muito altos, geralmente tensões entre 138 kV e 750 kV. A elevação da tensão diminui o valor da corrente elétrica e conseqüentemente as perdas por efeito Joule.

Ao chegar próximo dos grandes centros consumidores as linhas de transmissão dão lugar às linhas de distribuição de energia, que é a terceira parte do SEP. Como as tensões do sistema de transmissão são muito elevadas, existem subestações abaixadoras no sistema, de modo que a tensão é baixada para níveis mais seguros para distribuição de eletricidade para os consumidores finais.

Devido ao elevado grau de complexidade e aos grandes investimentos necessários para implantação e operação dos sistemas de geração e transmissão, por muito tempo, a aplicação de tecnologias de comunicação e automação nos sistemas elétricos de potência (SEP) ficaram restritas a esses dois subsistemas. Além disso, problemas nos subsistemas de geração ou transmissão afetam uma quantidade muito maior de consumidores, quando comparados com problemas no subsistema de distribuição (CURCIC et al., 1995).

## 4.1 O problema da Restauração de SED

Durante muito tempo, o setor de distribuição de eletricidade foi negligenciado em diversos aspectos, em especial no que tange à modernização e automação da rede elétrica, constituindo hoje uma das partes mais vulneráveis do SEP.

A tendência atual é a inserção de novas tecnologias no SED. Além disso, a complexidade na operação destes sistemas vem crescendo, e chamando a atenção de pesquisadores para esta parte do SEP, principalmente devido a penetração de geração distribuída na rede.

Segundo Curcic et al. (1995), no Reino Unido, em média 90% do tempo que um consumidor passava sem fornecimento de energia durante o ano eram em decorrência de defeitos ocorridos na rede de distribuição. Este dado, embora não seja recente, mostra a necessidade de serem realizados investimentos constantes também no SED.

Dessa forma, os problemas na rede elétrica de distribuição de energia causam graves transtornos e precisam ser contornados no menor tempo possível, ou ao menos terem seus efeitos reduzidos a uma parcela menor de consumidores.

### 4.1.1 Descrição do Problema

Quando ocorrem faltas temporárias no sistema, os religadores devem atuar, tentando reenergizar os setores que ocasionaram a falta. Se a falta for de natureza temporária, como por exemplo galhos de árvores que tocam os cabos de eletricidade, o setor será reenergizado, sem maiores danos ao sistema. Entretanto, quando o defeito é permanente, por exemplo, cabos rompidos, os operadores precisam seguir uma série de passos até que o sistema volte ao seu estado normal de operação.

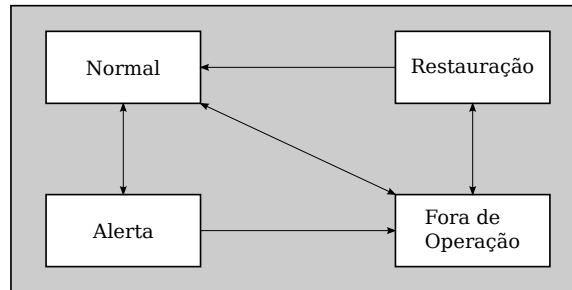
O primeiro passo para recompor o sistema é diagnosticar o problema (SAMPAIO, 2002). Para isso, existem diversas ferramentas que podem auxiliar os operadores na tomada de decisão, como por exemplo, alarmes gerados pelos equipamentos de proteção e adquiridos pelo SCADA e ligações telefônicas de consumidores. Apesar destes recursos, na maioria dos casos, o diagnóstico só pode ser concluído com o envio de uma equipe a campo para verificar a situação *in loco*.

Após o diagnóstico, o setor defeituoso deve ser localizado e isolado do restante da rede elétrica, o que também pode demandar muito tempo por parte da equipe que vai a campo. Como os sistemas de distribuição de energia elétrica, em sua grande maioria são radiais com recurso, todos os setores a jusante do setor faltoso também ficarão desenergizados, mesmo não possuindo trechos defeituosos. Uma vez isolado o setor em falta, os demais setores desenergizados podem ser novamente energizados por outros alimentadores, por meio de chaves de encontro de alimentadores.

O sistema deve então sofrer uma reconfiguração a fim de atender com segurança e qualidade os setores sem defeito. Tão logo o setor defeituoso seja reparado, o sistema pode retornar a sua configuração original.

De acordo com Curcic et al. (1995), para fins de análise, a rede elétrica pode estar em um dos estados mostrados na Figura 4.1.

Figura 4.1 – Estados de operação do sistema



Fonte: (CURCIC et al., 1995)

No estado de operação *Normal*, uma análise da rede pode ser efetuada a fim de reconfigurá-la para diminuir perdas e reduzir o desbalanceamento do sistema. O estado de *Alerta* é verificado quando algumas restrições são violadas, por exemplo, sobrecarga de condutores ou transformadores, o que pode causar a atuação da proteção e retirar a porção afetada do sistema.

Quando a proteção atua e isola uma porção afetada por problema, então o estado *Fora de Operação* é atingido. Quando o problema é simples e pode ser resolvido rapidamente, então o sistema pode voltar diretamente para o estado normal de operação. Entretanto há casos que a resolução do problema na rede pode demorar várias horas (de 4 até 10 horas em média). Neste caso o sistema deve ser restaurado, de modo que o mínimo de consumidores sejam afetados, este é o estado de *Restauração*.

Desse modo, de acordo com Curcic et al. (1995) a restauração da rede elétrica mediante uma situação de contingência é um procedimento de emergência e temporário, devendo ser preservado, enquanto a porção defeituosa da rede sofre reparos. A reconfiguração deve ser realizada rapidamente e restaurar o máximo de consumidores possíveis ou priorizar cargas essenciais, segundo os critérios da concessionária.

### 4.1.2 Classificação do Problema

A restauração do SED tem como objetivos minimizar o número de clientes sem fornecimento, obedecendo critérios que priorizem cargas essenciais e minimizar o número total de operações de manobra em equipamentos de disjunção.

Estes objetivos são conflitantes entre si e estão sujeitos às seguintes restrições:

- ausência de sobrecarga em condutores e transformadores;
- manutenção dos níveis de tensão dentro dos limites estabelecidos pela legislação;
- manutenção da radialidade da rede.

Assim, o problema da restauração da rede elétrica de distribuição é tratado como um problema de otimização multi-objetivo, sujeito a múltiplas restrições (MARQUES, 2013), além de ser um problema combinatorial do número de chaves, uma vez que em um SED com  $n$  chaves existem  $2^n$  configurações possíveis de serem avaliadas. De acordo com Viana (1998), considerando que cada configuração fosse analisada em  $1\mu s$ , se  $n = 50$ , então seriam necessários 35 anos para que a análise fosse concluída.

Tendo em vista este cenário, muitas técnicas de otimização vêm sendo aplicadas com o propósito de resolver este problema, conforme descrito no Capítulo 1. Porém, para que qualquer técnica seja passível de implementação, o algoritmo, ou software a ser implementado precisa ter uma visão da rede a ser manipulada para que a partir da configuração inicial do sistema em contingência, seja gerada uma configuração ótima. Neste trabalho a rede elétrica será representada utilizando conceitos da teoria dos grafos por meio da estrutura de dados Representação Nó-Profundidade (RNP) (DELBEM et al., 2004).

### 4.1.3 Descrição Matemática de Recomposição de um SED

Nesta seção a problemática da recomposição automática de alimentadores de distribuição de energia é apresentada e definida matematicamente. Para isso, é necessário que alguns conceitos de teoria de grafos sejam apresentados, baseados em Marques (2013):

**Definição 4.1** Um **grafo**, definido por  $G = (N, E)$  consiste de um conjunto finito de pontos, denominados **nós**  $N(G)$  interligados ou não por linhas, denominadas **arestas**  $E(G)$ .

**Definição 4.2** Em um grafo  $G$  qualquer, um **caminho** é uma sequência de nós, tal que, de cada um dos nós exista uma única aresta distinta para o nó seguinte.

**Definição 4.3** Se o nó inicial e final de um caminho são iguais, então o caminho é chamado **ciclo**.

**Definição 4.4** Se o nó inicial e final de um caminho não forem iguais, então o caminho é chamado de **cadeia**.

**Definição 4.5** Um **par conexo** é um par de nós com pelo menos um caminho entre eles.

**Definição 4.6** Um **grafo conexo** é um grafo no qual todos os pares de nós são um par conexo.

**Definição 4.7** Uma **árvore** é um grafo acíclico e conexo.

**Definição 4.8** Uma **subárvore** é um subgrafo de uma árvore.

**Definição 4.9**

**Definição 4.10** *Nó **raiz** é um nó de uma árvore tomado como referência.*

**Definição 4.11** ***Profundidade** de um nó em uma árvore é o número de arestas do caminho formado entre este nó e o nó raiz.*

Para tratar computacionalmente o problema da restauração, a rede elétrica de distribuição será representada por meio da teoria dos grafos. A rede elétrica é modelada como uma floresta de grafo (um conjunto de árvores de grafo) em que os nós dos grafos representam os setores, i.e, porções da rede elétrica delimitadas por chaves, e as arestas dos grafos representam as chaves do sistema. O problema da restauração pode então ser definido matematicamente, de acordo com o exposto em Marques et al. (2014):

**Definição 4.12** *Dado  $G$  ser uma floresta de grafos acíclicos e conexos e  $G^o$  a condição inicial de  $G$ , que representa uma configuração radial da rede, então o problema de restauração da rede elétrica de distribuição de energia com topologia radial pode ser representado pelo seguinte equacionamento:*

$$\text{Min. } \phi(G), \psi(G, G^o)$$

$$\text{d.m.q : } X(G) \leq 1$$

$$B(G) \leq 1$$

$$V(G) \leq 1$$

Em que:

- $\phi(G)$  é o número de consumidores são fora de serviço em  $G$ ;
- $\psi(G, G^o)$  é o número de operações de manobra em chaves no sistema para se chegar em  $G$  a partir de  $G^o$  após a localização e isolamento da falta;
- $X(G)$  é o máximo carregamento da rede em  $G$  dado pelo máximo valor da razão  $x_j/\bar{x}_j$ , sendo  $\bar{x}_j$  um limitante superior de corrente em uma linha  $j$  que possui em  $G$  uma corrente  $x_j$ ;
- $B(G)$  é o máximo carregamento de subestação em  $G$  dado pelo máximo valor da razão  $b_s/\bar{b}_s$ , sendo  $\bar{b}_s$  um limitante superior de injeção de corrente em uma subestação  $s$  que possui em  $G$  uma magnitude de injeção de corrente igual a  $b_s$ ;
- $V(G)$  é a máxima queda de tensão em  $G$  dada por  $V(G) = \text{MAX}(|v_s - v_k|)/\delta$ , sendo  $v_s$  a magnitude de tensão em uma subestação  $s$ ,  $v_k$  a magnitude de tensão em uma barra  $k$  e  $\delta$  a máxima queda de tensão admissível.

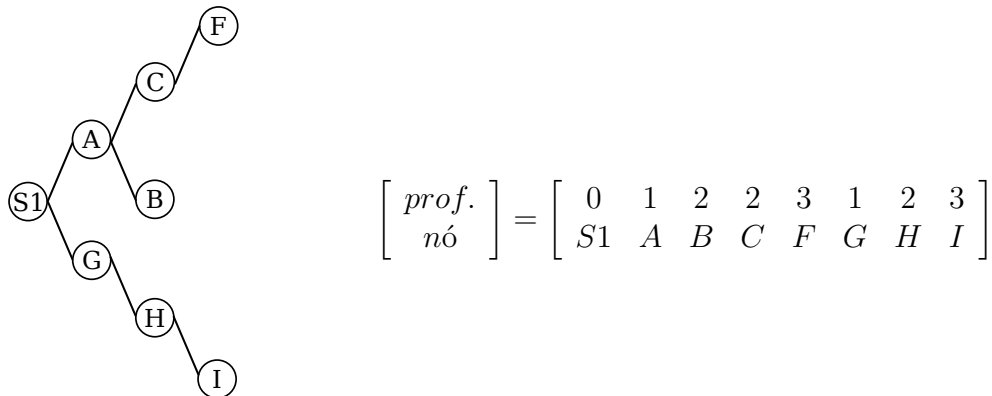
### 4.1.4 Estrutura de Dados RNP

Para representar os grafos utilizados na descrição da rede elétrica, utilizou-se a representação nó-profundidade (RNP). A RNP, proposta por (DELBEM et al., 2004), baseia-se nos conceitos de caminho e profundidade de nó em uma árvore de grafo, apresentados nas Definições 4.2 e 4.11, respectivamente. Uma nova definição será adicionada para incluir o conceito de RNP.

**Definição 4.13** A RNP consiste em uma lista linear, contendo os nós da árvore e suas respectivas profundidades, formando pares do tipo  $(n_x; p_x)$ , sendo  $n_x$  o nó da árvore e  $p_x$  a profundidade do nó;

Computacionalmente a lista contendo os pares  $(n_x; p_x)$  é representada por uma matriz de dimensão  $2 \times n$ . A ordem em que os pares  $(n_x; p_x)$  são dispostos na matriz é importante e deve ser determinada por um algoritmo de **busca em profundidade**. Na Figura 4.2 é apresentada uma árvore de grafo juntamente com sua representação RNP em forma de matriz  $2 \times n$  de acordo com a definição 4.13.

Figura 4.2 – Grafo simples e sua representação matricial em RNP.



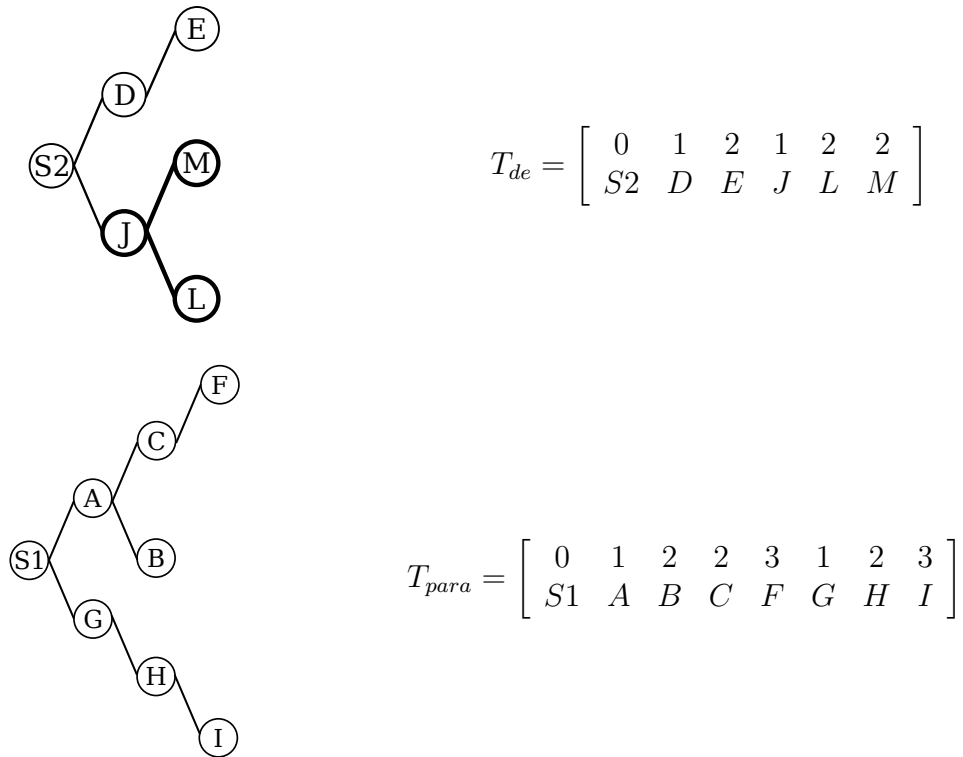
Fonte: Própria do autor.

Por meio dos grafos em RNP é possível realizar manipulações nas árvores de grafos utilizando operadores de poda e de inserção, representando alterações na configuração de uma rede elétrica. Por exemplo, na Figura 4.3 são mostradas duas árvores de grafo com suas respectivas representações matriciais em RNP. As árvores são chamadas de  $T_{de}$  e  $T_{para}$ .

Na Figura 4.3 também é mostrado em destaque um dos ramos da árvore  $T_{de}$ . Este ramo sofrerá uma operação de poda, ou seja, será retirado da árvore  $T_{de}$  e inserido na árvore  $T_{para}$ .

Na Figura 4.4 são mostradas as duas novas árvores de grafo. Trata-se das árvores  $T'_{de}$  que teve um de seus ramos podados e  $T'_{para}$  que recebeu o ramo podado de  $T_{de}$ . Na Figura 4.4 também é mostrada que a estrutura matricial da RNP permanece coerente com seu respectivo grafo.

Figura 4.3 – Operação de poda em um grafo.



Fonte: Própria do autor.

Existem duas possíveis situações de inserção de ramos em uma árvore de grafo. Na primeira situação, o nó raiz do ramo podado continua sendo o nó raiz do ramo após sua inserção em outra árvore de grafo, esta é a situação descrita nas Figuras 4.3 e 4.4. A outra situação acontece, quando o ramo podado ao ser inserido em outra árvore tem seu nó raiz modificado.

Em Marques (2013) essas duas situações são chamadas, respectivamente, de operadores PAO (*Preserve Ancestor Operator*) e CAO (*Change Ancestor Operator*) e são utilizados para gerar novas configurações da rede a ser restaurada. Cada um destes operadores foi implementado na classe que representa o grafo da rede.

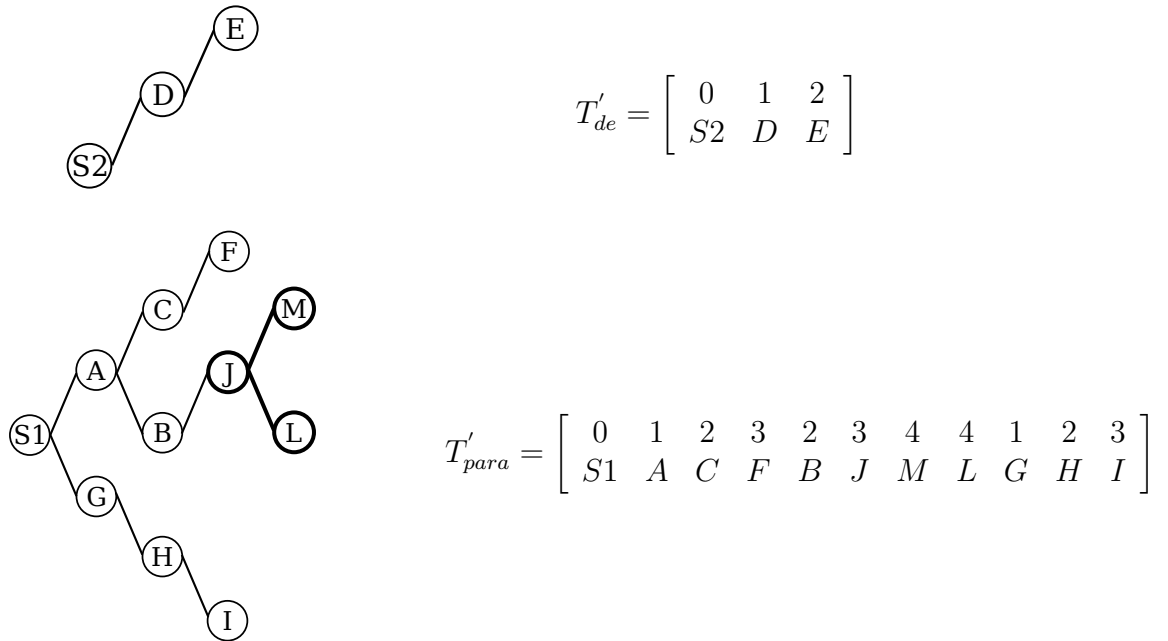
Dado que a representação de uma árvore de grafo foi definida no padrão RNP, é necessário estabelecer uma conexão entre os elementos da rede elétrica e os elementos que compõem a árvore de grafo. Para isso, segue-se a Definição 4.14

**Definição 4.14** Para representar um Sistema de Distribuição Radial, define-se a **RNP de alimentador**, de forma que:

- Alimentadores são considerados árvores;
- Setores são considerados nós;
- Chaves são consideradas arestas;



Figura 4.4 – Operação de poda em um grafo.

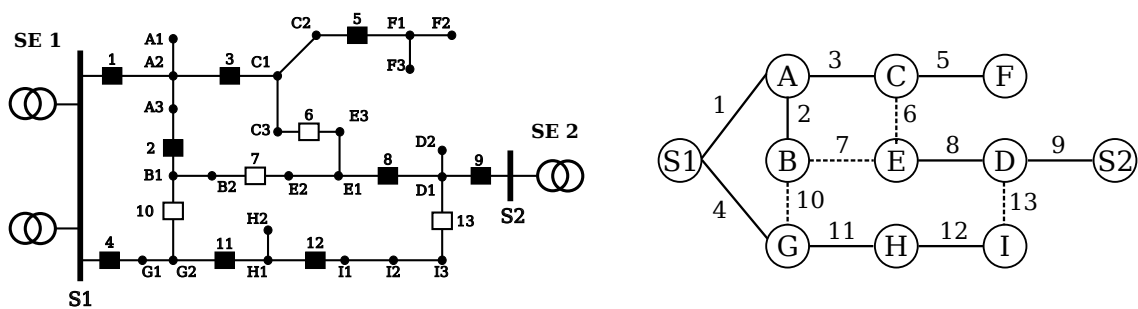


Fonte: Própria do autor.

- As barras das subestações são consideradas os nós raízes das árvores.

Dessa forma um SDR composto de vários alimentadores pode ser representado como uma floresta de grafo formada por várias árvores. Na Figura 4.5 é dado um exemplo de representação da rede elétrica por meio de grafos, que podem ser escritos com a representação RNP de alimentador.

Figura 4.5 – Exemplo de representação por grafo dos elementos que compõem uma rede elétrica teste.



Fonte: Própria do autor.

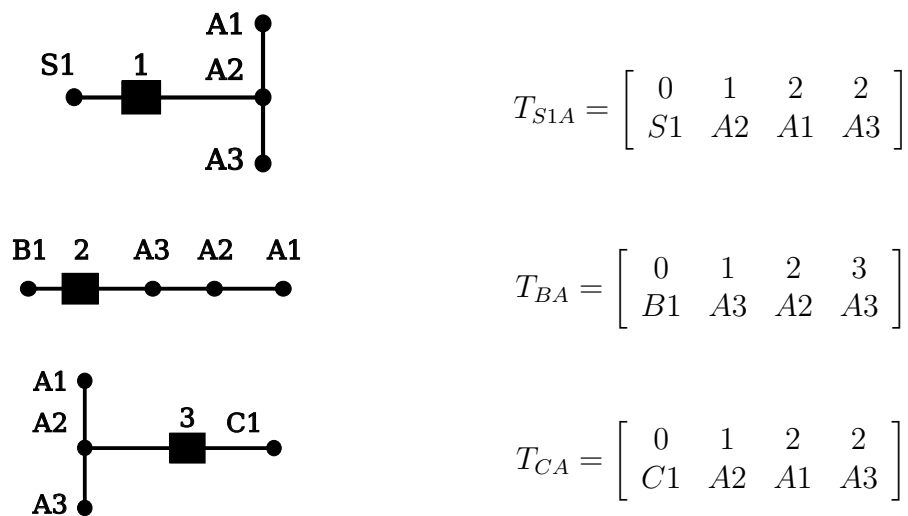
Com o objetivo de representar a rede elétrica da maneira mais fiel possível, Marques (2013) define também a RNP de setor, uma vez que a RNP de alimentador não permite a representação da distribuição de cargas ao longo dos setores de cada alimentador.

**Definição 4.15** Na **RNP de setor** codifica-se uma árvore de grafo, na qual os nós representam barras de carga e as arestas representam trechos de linha não separados por chave (à exceção da aresta que conecta o nó raiz à árvore).

Na Figura 4.6 são mostradas as três possibilidades de alimentação do setor  $A$  da rede mostrada na Figura 4.5 e suas respectivas representações em RNP de setor. Dessa forma, é possível concluir que:

- o número de representações em RNP de um setor é igual ao número de alternativas de alimentação do setor;
- a RNP de setor permite a representação detalhada da topologia das cargas distribuídas em um setor de acordo com a maneira pela qual ele é alimentado.

Figura 4.6 – Três possíveis configurações de RNP de setor.



Utilizando os conceitos de teoria dos grafos em associação com o conceito de RNP é possível ter uma representação fiel da rede elétrica, sendo a RNP de Alimentador utilizada para representar computacionalmente a topologia da rede de maneira macro, permitindo ainda sua manipulação por meio dos operadores PAO e CAO. A RNP de Setor, por outro lado, permite a execução de um fluxo de potência de Varredura Direta/Inversa, considerando-se todas as barras de carga do sistema e todos os trechos de linha, sem recorrer a simplificações (MARQUES, 2013).

## 4.2 Interface de Programação de Aplicações para Representação da Rede Elétrica

Para tornar a representação nó-profundidade (RNP) uma realidade, é necessário implementar esta estrutura em alguma linguagem de programação. Uma vez que Python é a linguagem de programação utilizada no desenvolvimento do SMA, Python também é a linguagem de programação adotada para a construção da estrutura de dados que representa a rede.

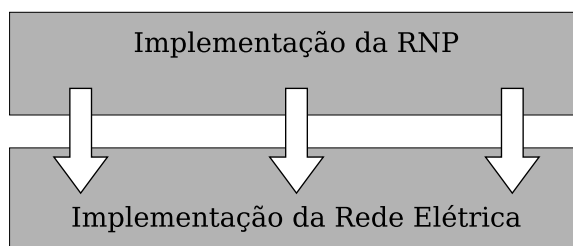
Entretanto, mais que representar a estrutura de dados RNP, é necessário que o sistema multiagente (SMA) tenha uma visão da rede elétrica e de seus parâmetros para que seja possível a realização de análises e alterações na topologia, ou seja, o SMA deve ser provido de um conjunto de aplicações que tenham o propósito de representar a rede nos aspectos que permitam a realização dessas análises, como por exemplo verificação de nível de tensão, carregamento de condutores, radialidade da rede, entre outros.

A representação em nó-profundidade por si só não possui essas características de representação da rede elétrica, uma vez que trata-se apenas de uma estrutura de dados para representar grafos, dessa forma, é preciso carregar estas estruturas de grafos com dados que tenham significado prático.

Com este propósito foi desenvolvido uma interface de programação de aplicações (API) para representação da rede elétrica, que tem como base a estrutura RNP e que pode ser utilizada pelo sistema multiagente ou por qualquer aplicação de software. A API tem uma estrutura orientada a objetos sendo composta de módulos que possuem classes e que por sua vez possuem atributos e métodos.

A API está dividida em dois módulos conforme mostrado na Figura 4.7. O primeiro destes módulos possui as classes que têm por objetivo representar grafos no padrão RNP. Este módulo possui as classes *Arvore*, *No* e *Aresta*. Cada uma dessas classes possui métodos e atributos que retornam informações sobre a estrutura do grafo ou realizam alterações em sua estrutura, por meio dos métodos de poda e inserção.

Figura 4.7 – Camadas da API de representação da rede elétrica.



Fonte: Própria do Autor.

No Algoritmo Algoritmo 4.1 é mostrado o código de entrada de dados para a classe *Arvore* do módulo que faz as associações com uma estrutura de dados de grafo no padrão RNP. Neste algoritmo são declarados dois objetos: `arvore_1` e `arvore_2`. Como entrada de parâmetros, estes objetos recebem uma estrutura de dados Python chamada `dicionário`. Um `dicionário` possui pares do tipo `(chave, valor)`, onde o atributo chave será o nome do nó do grafo e o atributo valor será uma lista com os vizinhos do nó. Com esses dados a classe *Arvore* gera uma estrutura de dados RNP que pode ser acessada pela chamada deste atributo e oferece uma série de métodos que podem ser utilizados para manipular a árvore de grafo, com operações do tipo poda ou inserção.

**Algoritmo 4.1:** Código com a instanciação dos objetos que representam a RNP das árvores apresentadas na Figura 4.5

```

1 from rnp import Arvore
2
3 nos_arvore_1 = {'S1': ['A', 'G'],
4                 'A': ['S1', 'B', 'C'],
5                 'B': ['A'],
6                 'C': ['A', 'F'],
7                 'F': ['C'],
8                 'G': ['S1', 'H'],
9                 'H': ['G', 'I'],
10                'I': ['H']}
11
12 nos_arvore_2 = {'S2': ['D', 'J'],
13                 'D': ['S2', 'E'],
14                 'E': ['D'],
15                 'J': ['S2', 'M', 'L'],
16                 'M': ['J'],
17                 'L': ['J']}
18
19 arvore_1 = Arvore(nos_arvore_1, dtype=str)
20 arvore_1.ordenar(raiz='S1')
21
22 arvore_2 = Arvore(nos_arvore_2, dtype=str)
23 arvore_2.ordenar(raiz='S2')
```

Na Figura 4.8 são mostrados alguns procedimentos realizados nos objetos `arvore_1` e `arvore_2`, declarados no Algoritmo Algoritmo 4.1, executados em tempo real por meio do terminal de comandos Ipython. Na linha de comando `In[12]`, o atributo `arvore` é chamado e seu retorno exibido na linha `Out[12]` que mostra uma estrutura dicionário com os nós dos grafos e seus respectivos vizinhos. Na linha `In[13]`, o atributo `rnp` do objeto `arvore_1` é chamado e retorna uma matriz com a estrutura RNP de seu grafo, na linha `Out[13]`.

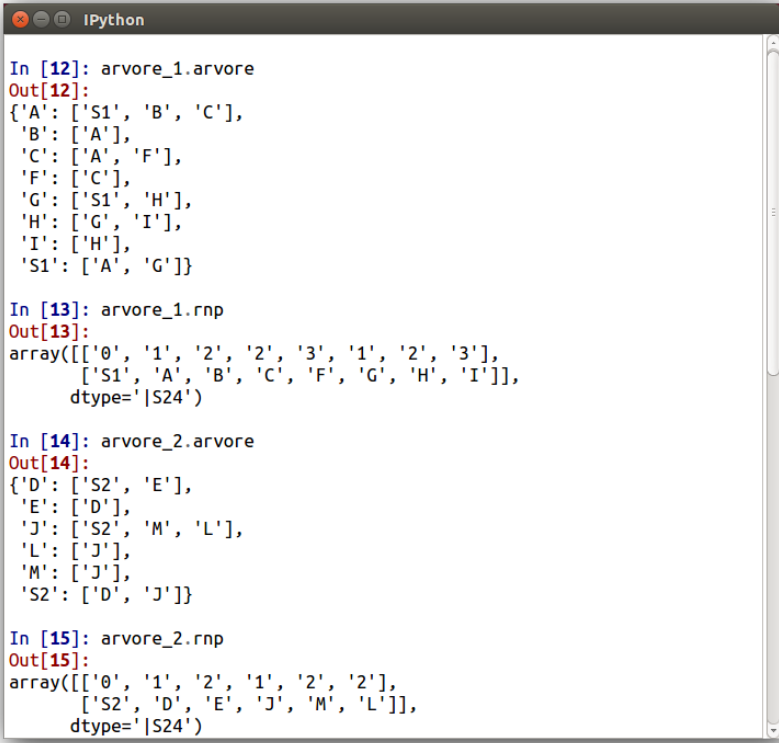
As linhas `In[14]`, `Out[14]` e `Out[15]`, `Out[15]` apresentam os mesmos procedimentos descritos no parágrafo anterior, mas aplicados ao objeto `arvore_2`.

O segundo módulo da API de representação da rede elétrica possui as classes que descrevem os componentes do sistema elétrico, e herdam das classes do primeiro módulo. As classes desenvolvidas são: `Subestacao`, `Transformador`, `Alimentador`, `Setor`, `NoDeCarga`, `Trecho` e `Chave`. Por exemplo, a classe `Chave` herda da classe `Aresta` do módulo de RNP, já a classe `Setor` herda da classe `No` e a classe `Alimentador` herda da classe `Arvore`. O diagrama UML das classes da API está mostrado na Figura 4.9, sendo que as classes que representam a API da rede elétrica estão destacadas.

Utilizando as classes disponibilizadas pela API é possível que o SMA tenha uma visão computacional de diversos aspectos da rede, como, por exemplo, carregamento dos condutores, potência consumida pelas cargas, topologia dos alimentadores, entre outras.

Com a API é possível a realização de alterações topológicas na rede por meio dos operadores poda e inserção, essenciais para sistemas de recomposição realizarem suas

Figura 4.8 – Tela do terminal de comandos IPython com utilização de objetos árvore e alguns de seus atributos



```

In [12]: arvore_1.arvore
Out[12]:
{'A': ['S1', 'B', 'C'],
 'B': ['A'],
 'C': ['A', 'F'],
 'F': ['C'],
 'G': ['S1', 'H'],
 'H': ['G', 'I'],
 'I': ['H'],
 'S1': ['A', 'G']}

In [13]: arvore_1.rnp
Out[13]:
array([[0, 1, 2, 2, 3, 1, 2, 3],
       ['S1', 'A', 'B', 'C', 'F', 'G', 'H', 'I']],
      dtype='|S24')

In [14]: arvore_2.arvore
Out[14]:
{'D': ['S2', 'E'],
 'E': ['D'],
 'J': ['S2', 'M', 'L'],
 'L': ['J'],
 'M': ['J'],
 'S2': ['D', 'J']}

In [15]: arvore_2.rnp
Out[15]:
array([[0, 1, 2, 1, 2, 2],
       ['S2', 'D', 'E', 'J', 'M', 'L']],
      dtype='|S24')

```

Fonte: Própria do autor.

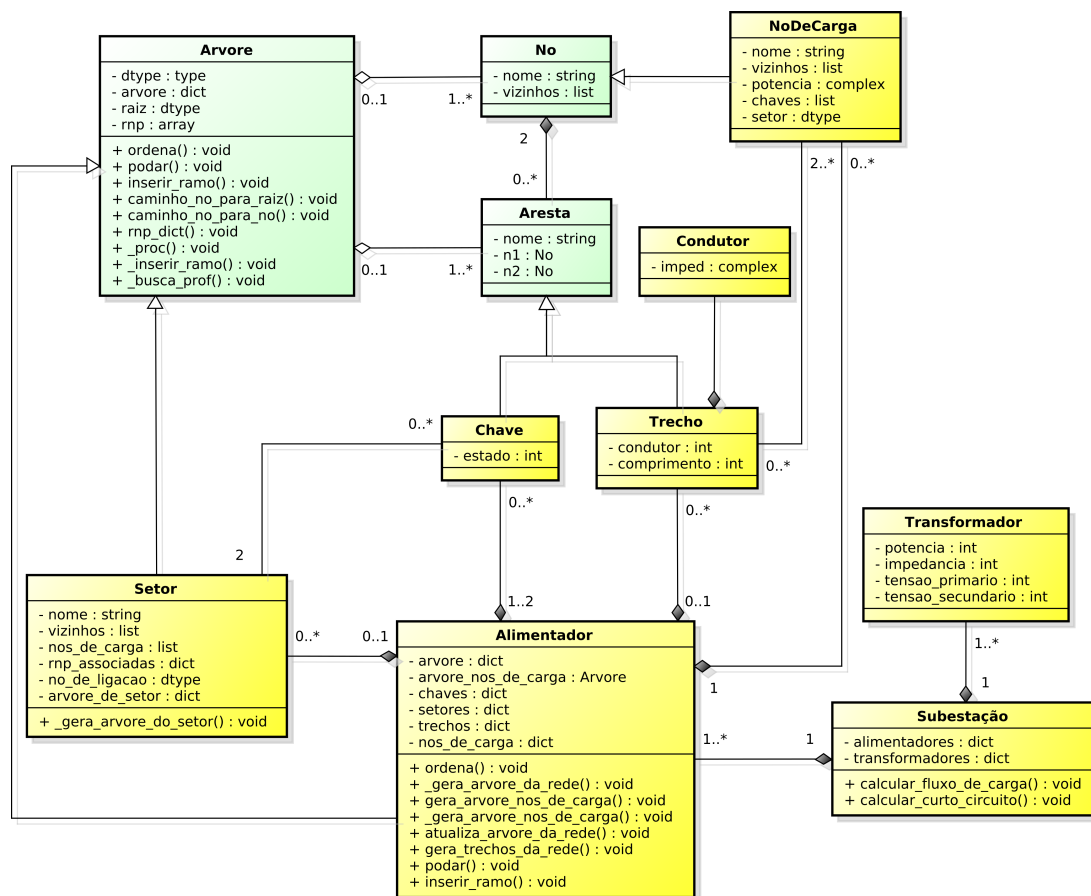
análises.

Tomando como exemplo a rede elétrica mostrada na Figura 4.5 e supondo a operação de transferência dos setores C e F do alimentador da subestação S1 para o alimentador da subestação S2, seriam necessárias apenas as operações de poda e de inserção sob os objetos que representam estes alimentadores.

Na Figura 4.10 as operações com as estruturas de dados que representam os alimentadores, denominados de `sub_1_al_1` e `sub_2_al_1` são realizadas no terminal de comandos IPython. Na linha de comando `ln[10]` é inserido um comando que retorna a matriz RNP de alimentador do alimentador representado pelo objeto `sub_1_al_1`, que é exibida na saída do comando, representada por `Out[10]`.

Na linha de comando `ln[11]` é realizada a operação de poda sob o objeto `sub_1_al_1` e o resultado armazenado na variável `poda`. Em seguida, na linha de comando `ln[12]` é chamada a estrutura RNP do objeto após a operação de poda e seu resultado mostrado na linha `Out[12]`. Na linha `ln[13]` a matriz RNP do objeto `sub_2_al_1` é chamada e mostrada na linha `Out[13]`, seguida da operação de inserção de ramo, realizada na linha `ln[14]`. Por fim, a matriz RNP do alimentador representado pelo objeto `sub_2_al_1` é chamada na linha `ln[15]` e mostrada na linha `Out[15]`.

Figura 4.9 – Diagrama UML das classes desenvolvidas para a API.



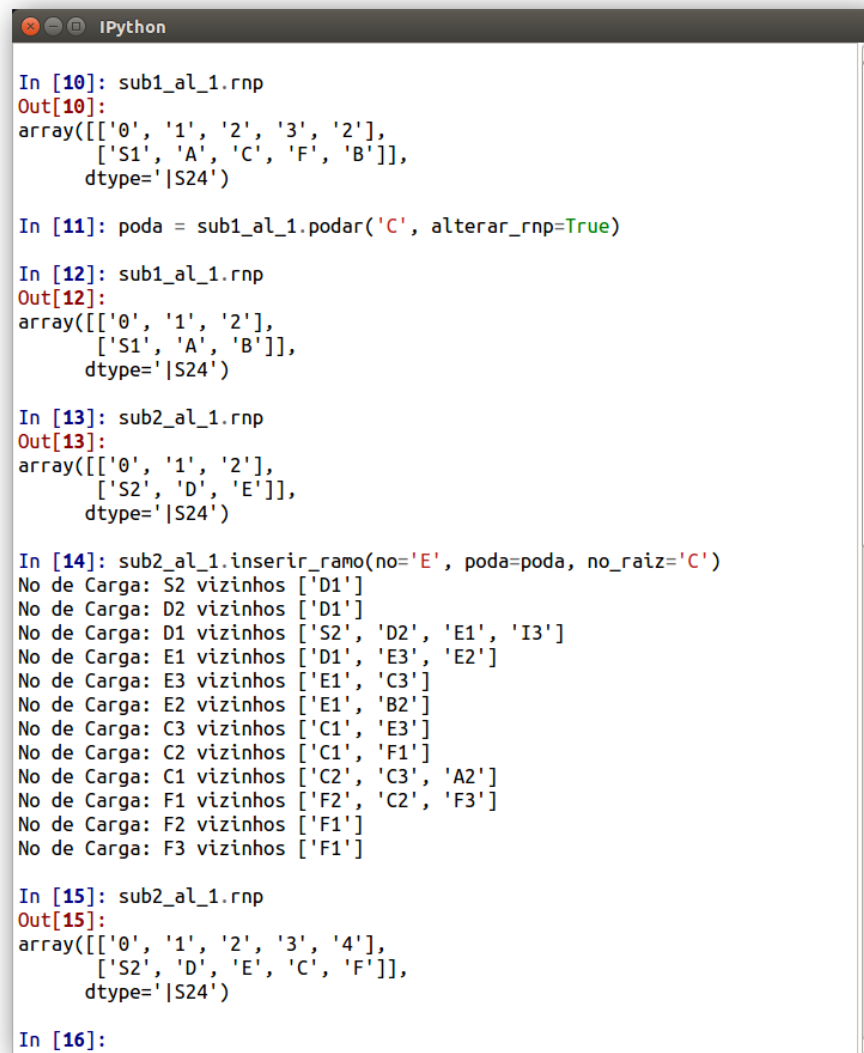
Fonte: Própria do autor.

## 4.2.1 XML para Entrada de Dados

Para que a entrada de dados se desse de maneira mais intuitiva e fácil de ser alterada, foi desenvolvida uma estrutura em XML que permitisse a representação de todos os elementos da API de representação da rede elétrica, estabelecendo seus elementos principais e a interligação entre eles.

Para que estes dados possam ser lidos e interpretados em objetos da API, um módulo de conversão que lê o arquivo em XML e carrega os elementos que estão descritos foi implementado. Dessa forma é possível armazenar as informações da rede que se pretende representar em arquivo de texto no formato XML, um padrão internacional para representação de informações, de fácil leitura, tanto para pessoas como para softwares, e que permite sua manipulação de forma intuitiva. Esse processo de conversão das informações contidas no arquivo XML em objetos dinâmicos da API de representação da rede elétrica, desenvolvida em Python, está representado na Figura 4.11. Já a estrutura do arquivo XML está descrita no Apêndice A.

Figura 4.10 – Tela do terminal de comandos IPython com utilização dos comandos poda e inserção de ramos nos alimentadores.



```

IPython

In [10]: sub1_al_1.rnp
Out[10]:
array([[ '0', '1', '2', '3', '2'],
       ['S1', 'A', 'C', 'F', 'B']],
      dtype='|S24')

In [11]: poda = sub1_al_1.podar('C', alterar_rnp=True)

In [12]: sub1_al_1.rnp
Out[12]:
array([[ '0', '1', '2'],
       ['S1', 'A', 'B']],
      dtype='|S24')

In [13]: sub2_al_1.rnp
Out[13]:
array([[ '0', '1', '2'],
       ['S2', 'D', 'E']],
      dtype='|S24')

In [14]: sub2_al_1.inserir_ramo(no='E', poda=poda, no_raiz='C')
No de Carga: S2 vizinhos ['D1']
No de Carga: D2 vizinhos ['D1']
No de Carga: D1 vizinhos ['S2', 'D2', 'E1', 'I3']
No de Carga: E1 vizinhos ['D1', 'E3', 'E2']
No de Carga: E3 vizinhos ['E1', 'C3']
No de Carga: E2 vizinhos ['E1', 'B2']
No de Carga: C3 vizinhos ['C1', 'E3']
No de Carga: C2 vizinhos ['C1', 'F1']
No de Carga: C1 vizinhos ['C2', 'C3', 'A2']
No de Carga: F1 vizinhos ['F2', 'C2', 'F3']
No de Carga: F2 vizinhos ['F1']
No de Carga: F3 vizinhos ['F1']

In [15]: sub2_al_1.rnp
Out[15]:
array([[ '0', '1', '2', '3', '4'],
       ['S2', 'D', 'E', 'C', 'F']],
      dtype='|S24')

In [16]:

```

Fonte: Própria do autor.

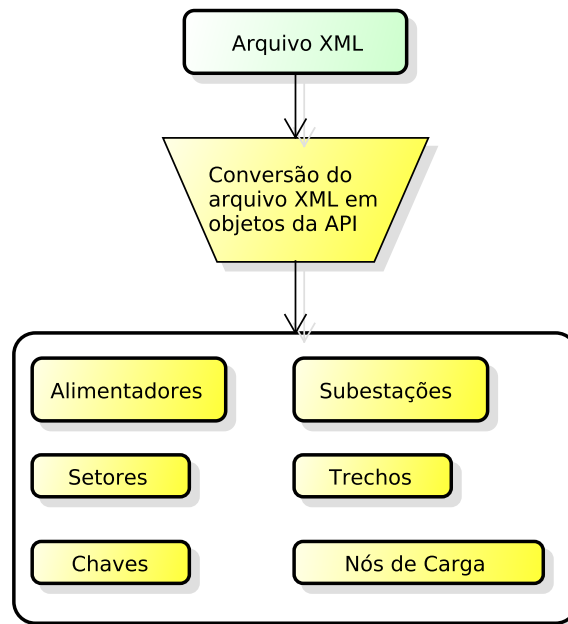
### 4.3 Estrutura de SMA Proposta

O sistema multiagente proposto neste trabalho foi desenvolvido utilizando o framework PADE, que permite a construção e execução dos agentes em um ambiente distribuído, além de proporcionar o monitoramento dos agentes e de suas trocas de mensagens.

Apenas dois tipos de agentes foram desenvolvidos para a realização da restauração dos alimentadores de distribuição da rede elétrica, são eles os agentes Alimentador (AA) e Dispositivo (AD). Estes agentes foram embarcados em hardwares de desenvolvimento que executam o sistema operacional Linux e que por seu tamanho reduzido podem facilmente ser conectados via interface de rede para que possam comunicar-se com outros dispositivos.

O agente Alimentador possui as principais funcionalidades do sistema de recomposição

Figura 4.11 – Processo de conversão XML para objetos dinâmicos da API em Python.



Fonte: Própria do autor.

automático. É de responsabilidade do agente alimentador, negociar potência com os demais agentes alimentadores, que podem realizar uma restauração e também analisar as restrições da rede por meio da execução de ferramentas como fluxo de carga.

O agente Dispositivo tem o papel de detectar qualquer problema que cause a desenergização dos setores do alimentador, isolar o setor do sistema que está sob falta e enviar comandos de abertura ou fechamento para os IED que controlam as chaves presentes na rede. É o agente Dispositivo que se comunica diretamente com os IED do sistema, portanto, os AD precisam ter implementados os protocolos de comunicação utilizados pelos IED. Neste trabalho o foco será nos protocolos estabelecidos pela norma IEC 61850, GOOSE e MMS, que serão detalhados no Capítulo 5.

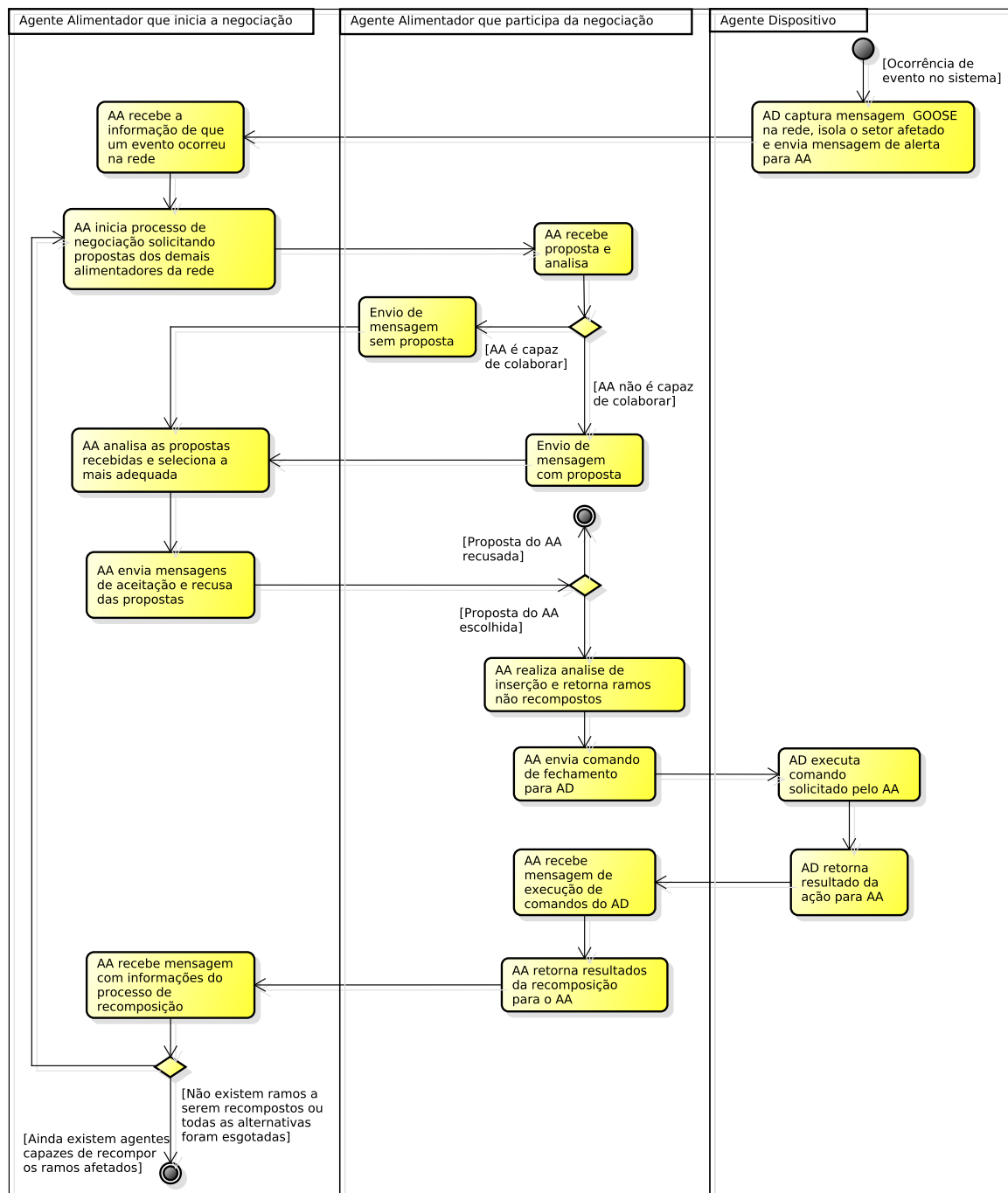
Na Figura 4.12 é mostrado um diagrama de atividades geral do SMRA. Neste diagrama é possível observar as atividades desenvolvidas tanto pelo AA, quanto pelo AD. Para cada alimentador físico de uma subestação deve existir um módulo que embarca um AA e um AD. Dessa forma, as atividades de um agente Alimentador são distribuídas entre diferentes AA, assim como as atividades de um agente Dispositivo também são distribuídas entre diferentes AD.

Quais agentes Alimentadores e Dispositivos desempenharão as funções do SMRA, dependerá de dois fatores: primeiro em que setor e alimentador ocorreu a falta e segundo por qual alimentador será realizada a recomposição.

No modelo desenvolvido para o SMRA, todos os agentes têm uma visão topológica da rede e sincronizam entre si suas informações. Sempre que algum dos AA for avisado por um AD sobre a abertura ou fechamento de um equipamento, o AA altera seu estado



Figura 4.12 – Diagrama de atividades do SMRA.



Fonte: Própria do autor.

topológico e então envia mensagem de atualização da rede para os outros AA. Desse modo todos os agentes têm uma visão atualizada da rede.

### 4.3.1 Agente Alimentador

O agente Alimentador possui as principais atribuições do sistema de recomposição automática. No SMRA proposto, existe um AA para cada alimentador das subestações.

Agentes Alimentadores têm comportamentos distintos quando realizam negociações que utilizam o protocolo FIPA ContractNet. Quando o AA atua como iniciante da negociação (AA responsável pelo setor atingido), suas principais atribuições são:

- requisitar ajuda de outros AA que possam restaurar a rede;
- analisar e selecionar as melhores propostas;
- procurar novas alternativas de recomposição caso ainda existam setores são desenergizados que não puderam ser recompostos na primeira tentativa de recomposição.

Quando o AA atua como participante da negociação, ou seja, é o agente solicitado por ajuda, então suas principais atribuições são:

- enviar proposta de ajuda ao AA iniciante;
- analisar as restrições operacionais da rede, como sobrecarga de condutores e de transformadores, e nível de tensão, por meio da execução de fluxo de carga;
- realizar a recomposição segundo as análises de restrições.

Esta divisão de tarefas entre AA iniciante e AA participante é fundamental para a eficiência do processo de restauração da rede, pois divide o mesmo em duas etapas bem distintas. A primeira etapa, realizada pelo AA iniciante, é a etapa de chamada por propostas, que busca escolher a melhor possibilidade de restauração entre todas as propostas oferecidas pelos agentes, isso gera uma análise de recomposição macro.

A segunda etapa de restauração é realizada pelo AA participante, ou seja, aquele que teve sua proposta escolhida. Nesta etapa o ramo de alimentador desenergizado é inserido virtualmente no alimentador para que possam ser realizadas análises de diversos critérios, tais como as restrições operacionais do alimentador e também o nível de prioridade de recomposição dos setores que compõem o ramo de alimentador, isso gera uma análise de recomposição local.

#### **4.3.1.1 Descrição dos Comportamentos do AA**

Como descrito no Capítulo 3, no PADE os comportamentos de um agente são implementados por classes que herdam seus métodos de outras classes do tipo comportamento. Na Tabela 4.1 são enumerados e descritos os comportamentos do agente Alimentador.

Para o agente Alimentador, dois comportamentos merecem destaque, o comportamento de chamada e análise de propostas e o comportamento de inserção de ramos desenergizados.

No comportamento de chamada e análise de propostas, que tem seu diagrama de atividade mostrado na Figura 4.13, o agente Alimentador envia mensagens para todos os agentes Alimentadores registrados no AMS e aguarda a chegada das propostas, começando

Tabela 4.1 – Comportamentos do agente Alimentador.

| Comportamento | Descrição   |
|---------------|---|
| CompRequest1  | Comportamento do tipo FIPA-Request Participante que o agente AA executa quando recebe mensagem do Agente AD com informações de TRIP no sistema. Este comportamento também lança o comportamento ContractNet Iniciante, realiza a poda dos setores desenergizados e envia mensagem de atualização da topologia da rede para outros AA. |
| CompRequest2  | Comportamento do tipo FIPA-Request Participante que recebe mensagens de atualização da topologia da rede.   |
| CompRequest3  | Comportamento do tipo FIPA-Request Participante que recebe pedido de identificação do agente Dispositivo associado.   |
| CompContNet1  | Comportamento FIPA-ContractNet Iniciante que envia mensagens CFP para outros agentes alimentadores solicitando propostas de restauração. Este comportamento também faz a análise das propostas e analisa-as selecionando a que julga ser a melhor.  |
| CompContNet2  | Comportamento FIPA-ContractNet Participante que é acionado quando um agente recebe uma mensagem do Tipo CFP enviando logo em seguida uma proposta e caso esta seja selecionada realiza as análises de restrição para que seja possível a restauração.   |

Fonte: Própria do autor.

sua análise assim que recebe todas as propostas, ou quando um tempo limite de espera é atingido. A prioridade de análise é dada às propostas de agentes pertencentes à mesma subestação, assim o alimentador atingido deverá preferencialmente ser recomposto pela mesma fonte que o alimentava anteriormente.

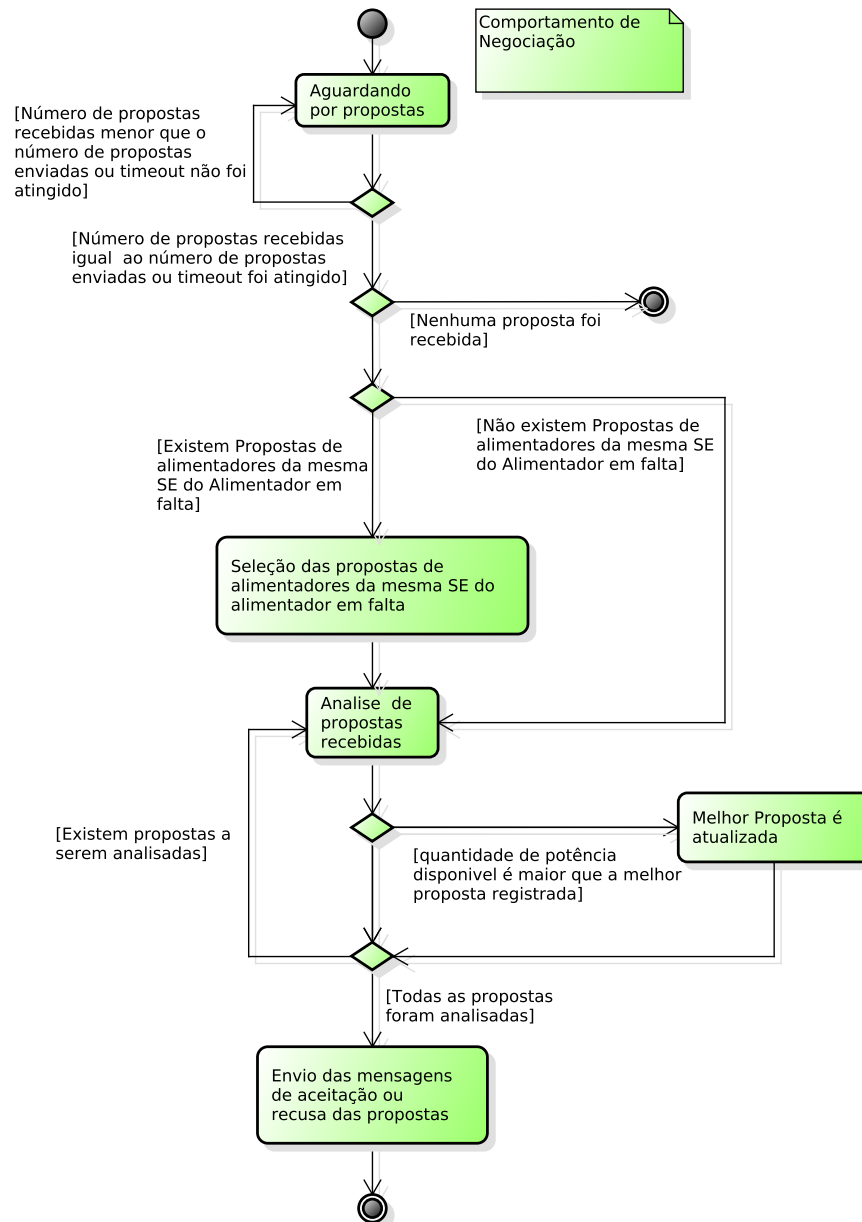
Após a análise e escolha da proposta mais adequada, uma resposta de aceitação ou recusa é enviada aos agentes, que dependendo da resposta irão realizar a restauração, se tiverem a proposta aceita, ou não fazer nada caso tenham a proposta recusada.

O comportamento do agente Alimentador ao realizar análises de inserção de ramos desenergizados na rede elétrica é mostrado por meio do diagrama de atividades da Figura 4.14. Este comportamento é um dos comportamentos mais importantes do AA, pois é nele que serão analisadas as restrições da rede, e caso este comportamento seja falho, uma recomposição mal sucedida será realizada, causando ainda mais danos ao sistema.

No comportamento de inserção, a primeira coisa a ser feita é a seleção de ramos de alimentador com possibilidade de serem reenergizados pelo alimentador escolhido para realizar a recomposição. Em seguida é feita uma análise de prioridade, caso existam mais de um ramo de alimentador possíveis de serem recompostos. Uma vez que o ramo tenha sido escolhido, a inserção virtual é realizada, seguida de três tipos de análise:

- carregamento dos transformadores da subestação a que pertence o alimentador de recomposição;

Figura 4.13 – Análise de propostas pelo AA.

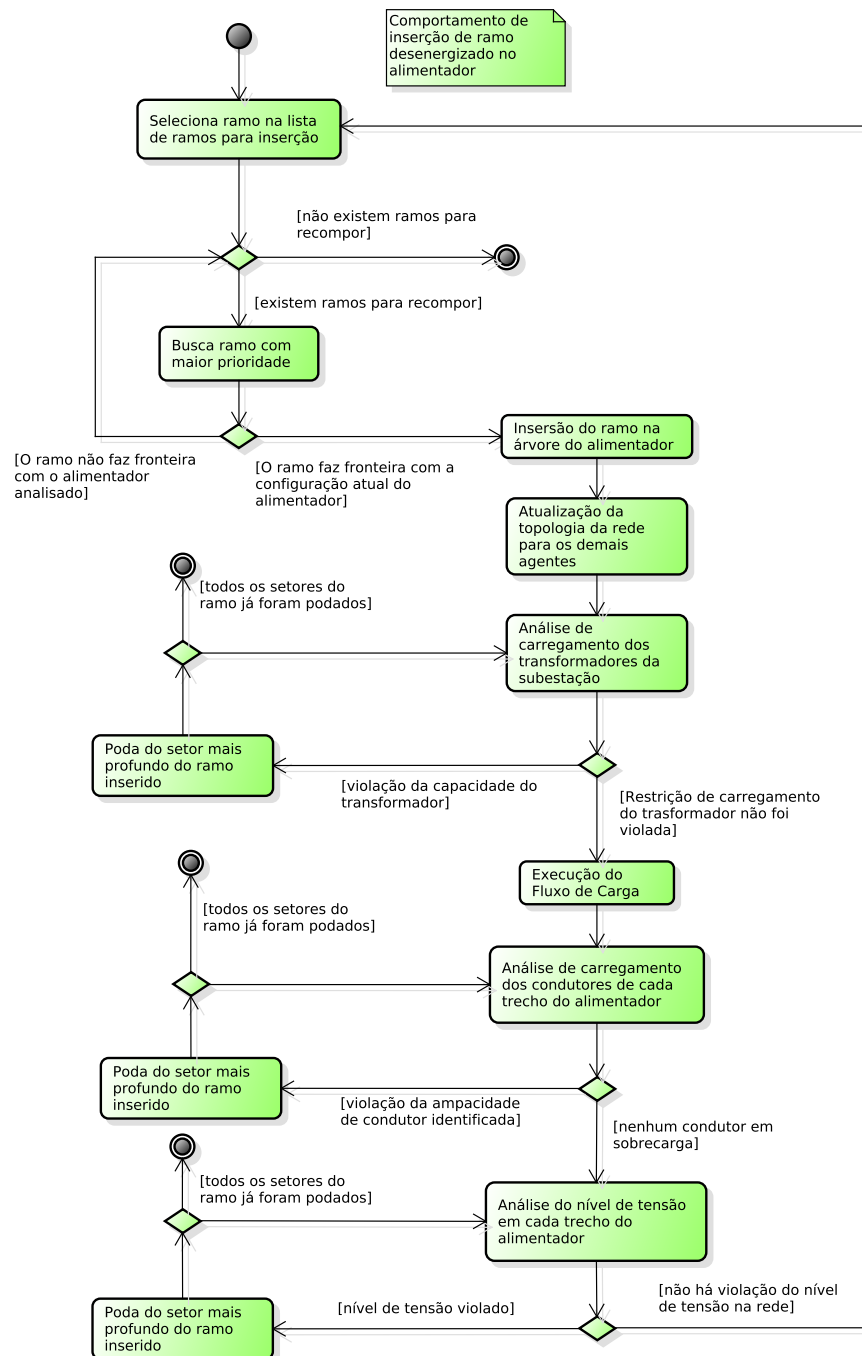


Fonte: Própria do autor.

- carregamento dos condutores do alimentador de recomposição e do ramo de alimentador inserido;
- queda de tensão ao longo do alimentador de recomposição e do ramo de alimentador inserido.

Caso qualquer uma destas análises indique uma violação das restrições da rede, o setor com maior profundidade na RNP de alimentador do ramo inserido será podado e as mesmas análises são realizadas novamente, com um setor a menos, até que não haja mais nenhum setor a ser podado do ramo de alimentador.

Figura 4.14 – Análise de inserção pelo AA.



Fonte: Própria do autor.

As análises de carregamento de condutores e de queda de tensão são realizadas por meio da execução de um algoritmo de fluxo de carga de varredura direta-inversa, que está associado à estrutura de subestação da API de representação da rede elétrica e sua descrição encontra-se no Apêndice B.

### 4.3.2 Agente Dispositivo

O agente dispositivo é o responsável pela comunicação direta com os IED da rede elétrica e por isso precisa ter conhecimento dos protocolos de comunicação utilizados por eles. O agente Dispositivo funciona como uma interface entre SMA e os equipamentos da rede. Sendo assim, alterações nos protocolos de comunicação dos IED acarretam alterações somente no agente Dispositivo.

As principais atribuições do agente Dispositivo são:

- receber mensagens de alerta dos IED na ocorrência de algum problema detectado na rede elétrica;
- isolar o setor que está sob falta;
- enviar mensagem de alerta ao agente Alimentador na ocorrência de algum problema na rede, para que este possa tomar medidas para que o sistema volte o mais rápido possível da normalidade;
- enviar comando de abertura ou fechamento para IED quando for solicitado por agentes Alimentador.

Os comportamentos de um agente Dispositivo estão descritos na Tabela 4.2.

Tabela 4.2 – Comportamentos do Agente Dispositivo

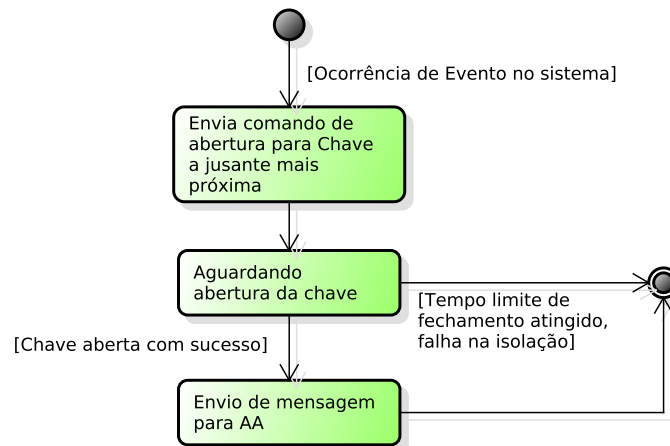
| Comportamento | Descrição   |
|---------------|---|
| CompRequest1  | Comportamento do tipo FIPA-Request Participante que o agente AD executa quando recebe uma mensagem GOOSE de um IED. Este comportamento identifica o setor sob falta e envia comando de abertura para os dispositivos de fronteira do setor faltoso. |
| CompRequest3  | Comportamento do tipo FIPA-Request Iniciante que o agente AD executa ao enviar a mensagem que informa ao agente AA sobre a ocorrência de uma falta no sistema.  |
| CompRequest3  | Comportamento do tipo FIPA-Request Iniciante que o agente AD executa ao enviar mensagem de identificação junto ao agente AA.  |

Fonte: Própria do autor.

O comportamento do agente Dispositivo, representado em forma de diagrama de atividades, é mostrado na Figura 4.15. Como mostrado no diagrama, o comportamento do agente Dispositivo é simples de ser representado, estando sua complexidade na implementação das rotinas de comunicação com os IED, que devem ser especificadas de acordo com o protocolo do dispositivo.

Como já foi mencionado neste capítulo, a ênfase do SMRA será a comunicação via protocolos IEC 61850, que serão descritos no Capítulo 5.

Figura 4.15 – Diagrama de atividades desenvolvidas pelo AD.



Fonte: Própria do autor.

## 4.4 Comunicação entre Agentes e IED

Conforme mostrado na Figura 4.16 o processo de comunicação entre agentes e entre agentes e IED ocorre em dois níveis distintos. No primeiro nível existe a comunicação entre AA e AD, que utilizam as funcionalidades disponibilizados pelo PADE para trocar mensagens nos protocolos especificados pela FIPA no padrão FIPA-ACL. No segundo nível existe a comunicação direta entre IED e AD, que ocorre com base nos protocolos especificados pela norma IEC 61850 e descritos no Capítulo 5, sendo o protocolo GOOSE utilizado para detectar uma possível atuação da proteção no alimentador e o protocolo MMS para o envio de comandos para os religadores comandados pelos IED.

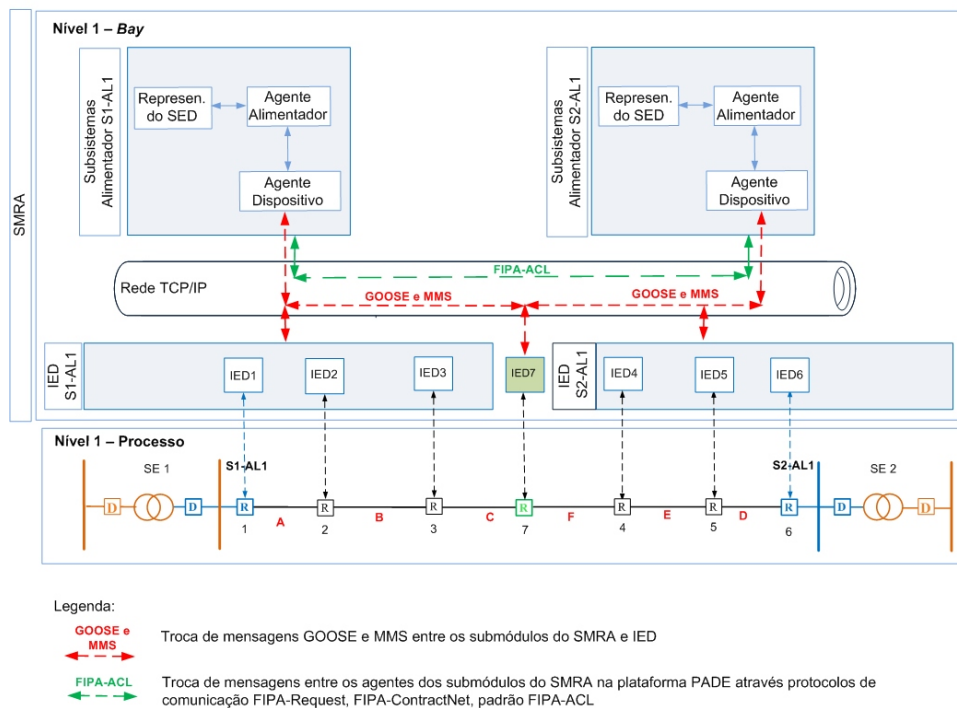
Com base nesta perspectiva de comunicação em dois níveis, nível de comunicação Agente-Agente e nível de comunicação AD-IED, é possível encapsular o processo de comunicação com os IED exclusivamente no AD que passa a funcionar como um driver de comunicação para o SMRA. Neste trabalho foram utilizados, para comunicação entre IED e SMRA os protocolos GOOSE e MMS, mas a comunicação também pode ser realizada por meio de outros protocolos, como por exemplo DNP3 (*Distributed Network Protocol*), *Modbus*, entre outros.

## 4.5 Conclusão

Neste capítulo uma parcela importante dos princípios base do SMRA foram apresentadas e descritas. Entre os principais assuntos discutidos estão:

- Definição matemática do problema da recomposição de alimentadores de distribuição de energia elétrica;

Figura 4.16 – Representação gráfica dos processos de comunicação entre IED e agentes que compõem o SMRA.



Fonte: Própria do autor.

- As definições necessárias para a representação da topologia de um alimentador de distribuição de energia elétrica por meio de teoria dos grafos utilizando a representação nó-profundidade, mais precisamente os conceitos de RNP de alimentador e RNP de setor, assim como a definição dos operadores de poda e inserção de ramos em uma árvore de grafo;
- Apresentação do diagrama de classes UML que descrevem a estrutura da API de representação da rede elétrica, que faz uma associação da RNP de alimentador e de setor com estruturas de dados que representam os componentes reais de um sistema de distribuição de energia elétrica;
- Definição dos agentes do SMRA, que são: agente alimentador e agente dispositivo. Os comportamentos destes agentes são descritos por meio de diagramas de atividade que mostram o fluxo das ações definidas para cada agente.

Com base nestes conceitos o SMRA tem as ferramentas necessárias para ser implementado por meio da plataforma PADE, podendo ser embarcado nos dispositivos de hardware conectados via rede de dados. Entretanto, para que o SMRA possa atuar sob os dispositivos da rede elétrica e realizar a função de *self-healing*, terá que se comunicar via protocolo de comunicação, que são os definidos pela norma IEC 61850, tema do Capítulo 5.



---

## Capítulo 5

# A Norma IEC 61850: Redes de Comunicação e Sistemas em Subestações

---

Para que um sistema de recomposição automática se torne realidade, a comunicação com os IED que controlam os dispositivos presentes na rede é de extrema importância. Os agentes que compõem o sistema de recomposição automática devem ser capazes de adquirir dados dos IED, como estados de chaves e medidas de corrente, tensão ou potência, e detectar a partida e atuação de funções de proteção. Além disso, os agentes que gerenciam o sistema de recomposição do sistema elétrico também devem ser capazes de enviar comandos de abertura e de fechamento para os IED que comandam os dispositivos de seccionamento do sistema.

Para que esses objetivos sejam alcançados, é necessário que os agentes consigam se comunicar com os IED por meio de protocolos de comunicação. Existem diversos protocolos de comunicação que são utilizados em ambientes de subestação, como MODBUS, DNP3, IEC 60870-5-101, e outros protocolos proprietários, SEL *Mirrored Bits* (SEL, 2013).

Estes protocolos muitas vezes apresentam dificuldade de integração entre dispositivos de fabricantes diferentes e alguns não apresentam os requisitos de confiabilidade que a rede elétrica necessita, pois são voltados para aplicações industriais.

### 5.1 O surgimento da Norma IEC 61850

Com o advento das redes de comunicação e com os benefícios da automação dos sistemas elétricos cada vez mais evidentes, um esforço conjunto de pesquisadores, desenvolvedores e

fabricantes de IED deu origem a duas abordagens que tinham como objetivo facilitar a automação de subestações e ao mesmo tempo definir e padronizar os protocolos necessários à essa tarefa. Nasceu assim o padrão IEC 61850, desenvolvido pelo comitê técnico 57 (TC57) do IEC, e o padrão Utility Communication Architecture (UCA) criado em 1990 pelo Electric Power Research Institute (EPRI) com objetivos semelhantes. Os dois grupos, TC57 e EPRI, uniram esforços em 1997 e no ano de 2003 foi lançada a primeira edição da norma IEC 61850 (LOPES et al., 2012).

A norma IEC 61850 tem o propósito de estabelecer padrões de comunicação, por meio de modelos de dados e protocolos, que permitam a interoperabilidade entre IED de diferentes fabricantes, utilizando para isso padrões e princípios de comunicação consolidados. O foco da norma é padronizar as funções envolvidas nos procedimentos comuns realizados em uma subestação, como por exemplo: envio de comando para os IED, coleta de medidas analógicas/digitais e realização de lógicas de automação entre IED (OZANSOY, 2010).

Além disso a norma também estabelece os requisitos mínimos de velocidade do sistema de automação e padroniza a forma como as ferramentas de engenharia podem ser utilizadas para configurar os IED, definindo para isso uma série de arquivos XML que armazenam as informações necessárias para configurar o dispositivo.

Neste trabalho os protocolos da norma IEC 61850 são utilizados, por diversos motivos, conforme descrito em (KIMURA et al., 2008):

- A norma vem sendo apontada como uma tendência para automação em subestações, vários fabricantes adotam em seus IED com os protocolos estabelecidos pela norma;
- Além de estabelecer os protocolos utilizados para comunicação de IED, a norma também estabelece um modelo de dados padronizado adotado para o acesso de qualquer tipo de dados disponibilizado pelo IED. Sendo assim, a norma tenta garantir a interoperabilidade entre IED de diferentes fabricantes;
- Estabelece e padroniza conceitos para a implementação de lógicas de automação por meio de mensagens de alta prioridade trocadas entre dispositivos, as mensagens Generic Object Oriented Substation Event (GOOSE), e também mensagens de menor prioridade que podem ser utilizadas para envio de informações a aplicações de supervisão e também para o envio de comando de controle aos IED, as mensagens Manufacturing Message Specification (MMS).

Atualmente não é comum a utilização dos protocolos da norma IEC 61850 para automação da rede elétrica de distribuição, principalmente por conta dos requisitos mínimos de velocidade de comunicação exigidos pela norma, mas num contexto de redes elétricas inteligentes e para redes de distribuição de médio porte ou em ambiente controlados como é o caso de refinarias de petróleo a norma IEC 61850 vem sendo utilizada (MOHAGHEGHI et al., 2009).

## 5.2 Estrutura da Norma IEC 61850

A norma IEC 61850 está dividida em dez módulos, conforme mostrado na Tabela 5.1. Cada um destes módulos trata de um assunto específico relacionado à automação de subestações. Vale destacar aqui, os seguintes módulos:

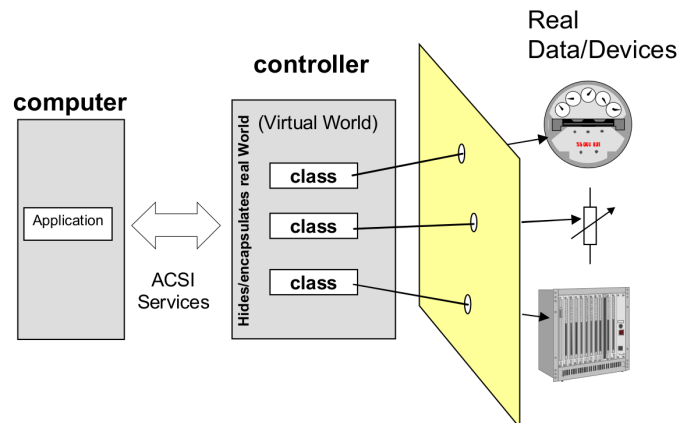
Tabela 5.1 – Estrutura da norma IEC 61850

|               |  |
|---------------|--|
| IEC 61850-1   | Introdução e Visão Geral   |
| IEC 61850-2   | Glossário  |
| IEC 61850-3   | Requisitos Gerais  |
| IEC 61850-4   | Gerenciamento de Sistema e Projetos  |
| IEC 61850-5   | Requisitos de Comunicação para Funções e Modelos de Dispositivos                 |
| IEC 61850-6   | Linguagem de Configuração para IEDs (SCL)  |
| IEC 61850-7   | Estrutura de Comunicação Básica para Equipamentos de Subestações e Alimentadores |
| IEC 61850-7-1 | Princípios e Modelos   |
| IEC 61850-7-2 | Serviços de Comunicação Abstratos (ACSI)   |
| IEC 61850-7-3 | Classes de Dados Comuns  |
| IEC 61850-7-4 | Classes de Nós Lógicos e de Dados Compatíveis                                    |
| IEC 61850-8   | Specific communication service mapping (SCSM)                                    |
| IEC 61850-8-1 | Mapeamento para MMS (ISO/IEC 9506 Part 1 and Part 2)                             |
| IEC 61850-9   | Specific communication service mapping (SCSM)                                    |
| IEC 61850-9-1 | Valores Amostrais sobre Enlace Serial Unidirecional Multidrop Ponto-a-Ponto      |
| IEC 61850-9-2 | Valores Amostrais sobre ISO/IEC 8802-3   |
| IEC 61850-10  | Testes de Conformidade   |

Fonte: (IEC/TR 61850-1, 2003).

- Módulo 6: este módulo trata da linguagem de configuração de subestação, ou *substation configuration language* (SCL), uma linguagem baseada nos padrões XML, que descreve o IED e suas relações com outros IED. Os arquivos XML no padrão SCL são utilizados para descrever a subestação e todas suas funções de comunicação, assim como para armazenar o modelo de dados do IED.
- Módulo 7.1: apresenta uma visão geral sobre o modelo de dados adotado pelos dispositivos IEC 61850, mostrando e descrevendo como é feita a comunicação entre IED em diferentes situações;
- Módulo 7.2: descreve os componentes base do modelo de dados, mostrando como cada componente deve ser inserido no contexto de uma interface abstrata de serviços de comunicação, ou *Abstract Communication Service Interface* (ACSI);
- Módulo 7.3: descreve as classes comuns de dados utilizadas pelos elementos que integram as classes de dados e os atributos de dados, indispensáveis na representação de pontos analógicos, digitais e de lógicas de controle e proteção;

Figura 5.1 – Interface de virtualização proposta pela norma IEC 61850.



Fonte: (IEC/TR 61850-7-1, 2003).

- Módulo 7.4: descreve todas as classes de nós lógicos e de dados por meio de tabelas que apresentam as informações que devem estar contidas, de forma obrigatória ou não, nas instâncias que representam os dados a serem disponibilizados pelo IED por meio da ACSI.

A partir destes módulos, a norma estabelece serviços de comunicação via ACSI que podem ser utilizados pelos sistemas de automação. A norma não estabelece como deve ser o funcionamento interno dos IED e de suas funções, mas apenas como estes devem disponibilizar os dados que são processados por seus componentes e como deve ocorrer a troca de dados entre dispositivos e aplicações. Este comportamento equivale a uma virtualização dos dispositivos físicos presentes no campo, supervisionados pelo sistema de automação, conforme é mostrado na Figura 5.1.

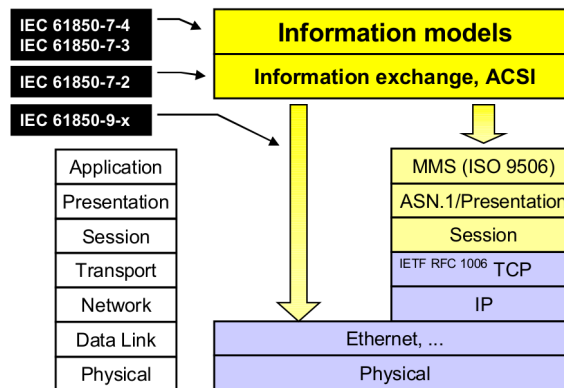
A ACSI estabelece dois tipos de interfaces para automação e controle dos IED (IEC/TR 61850-7-2, 2003):

- Interface abstrata que descreve a comunicação entre clientes e servidores, que disponibiliza serviços de acesso em tempo real aos dados do IED, envio de sinais de comando ao IED, registro de eventos, modelo de comunicação editor/assinante, auto descrição do modelo de dados, descoberta de dados e transferência de arquivos;
- Interface abstrata que estabelece comunicação rápida e segura de uma aplicação executada em um IED para aplicações executadas em diferentes IED, utilizando o modelo editor/assinante para troca de mensagens. Esta interface pode ser utilizada tanto para distribuição de eventos, como também para transmissão de valores de medição amostrados pelos IED.

Na Figura 5.2 as duas divisões da ACSI estão representadas com a indicação de onde ocorre o mapeamento de cada uma na pilha de camadas de protocolos de rede. A primeira

divisão da ACSI que estabelece comunicação cliente/servidor está mapeada na camada de aplicação por meio do protocolo Manufacturing Message Specification (MMS). A segunda divisão da ACSI está mapeada na camada de enlace por meio dos protocolos estabelecidos pela norma: Generic Object Substation Event Generic Object Oriented Substation Event (GOOSE) e Sampled Values (SV).

Figura 5.2 – Mapeamento da ACSI nas camadas dos protocolos de rede de computadores.



Fonte: (IEC/TR 61850-7-1, 2003).

## 5.3 Modelo de Dados da IEC 61850

Uma importante contribuição da norma IEC 61850 para estabelecer a interoperabilidade entre IED de um sistema de automação de subestação é criar um modelo de dados padronizado que atenda todas as especificidades dos equipamentos e entidades contidos no ambiente de subestações.

Na IEC/TR 61850-7-2 (2003) são descritos todos os conceitos base, que quando reunidos formam um modelo de dados padronizado que possibilita a definição de uma ACSI.

### 5.3.1 Modelo de Representação da Informação

De acordo com a norma IEC 61850, cada IED pode ser considerado como um Dispositivo Físico, ou *Physical Device* (PD), isto é, um hardware que executa todos os processos relacionados à sua função, seja ela de proteção ou de automação. Um PD se conecta fisicamente a um barramento de dados, tipicamente uma barramento TCP/IP via conexão Ethernet e troca dados nos padrões definidos pela norma.

Um Servidor é o componente da ACSI que armazena todos as instâncias dos modelos de dados definidos pela norma. É a partir do servidor que as conexões entre IED são criadas e gerenciadas. O servidor sempre terá um endereço IP e uma porta de comunicação, que serão utilizados para identificar os IED na rede e assim ser possível o estabelecimento de conexões entre eles.

Dispositivos Lógicos, ou *Logical Devices* (LD), são contêineres de Nós Lógicos e de Conjuntos de Dados, que são instâncias de dados menores. Um servidor pode conter um ou mais LD, de modo que todo LD deve possuir, necessariamente, um nó lógico LLN0 (nó lógico responsável pelo registro de mensagens GOOSE, *DataSets* e Relatórios), um nó lógico LPHD (nó lógico responsável pelas características de hardware do dispositivo) e, pelo menos, mais um nó lógico qualquer. Porém, além de servir como um agrupamento de instâncias de dados, cada LD possui serviços como *Generic Oriented Substation Event* (GOOSE), *Sampled Values* (SV) e *Group Settings* (GS), que não estão diretamente associados ao LD, mas ao nó lógico LLN0, uma vez que é mandatória a existência de um LLN0 em cada LD.

De acordo com Ozansoy (2010), a norma IEC 61850 classifica as funções do SAS e subdivide-as em sub-partes chamadas de nós-lógicos. Portanto, um Nó Lógico, ou *Logical Node* (LN), pode ser definido como uma subfunção de um dispositivo que troca informações com outros nós lógicos. Também existem nós lógicos que podem ser definidos simplesmente como representações virtuais de dispositivos físicos, como por exemplo, os nós lógicos que representam disjuntor (XCBR), transformador de corrente (TCTR) e transformador de potencial (TVTR), entre outros. Dessa forma, diferentes funções de automação e proteção em um ambiente de subestação podem ser modeladas como uma associação de nós-lógicos que trocam dados entre si, com o objetivo de realizar uma determinada ação.

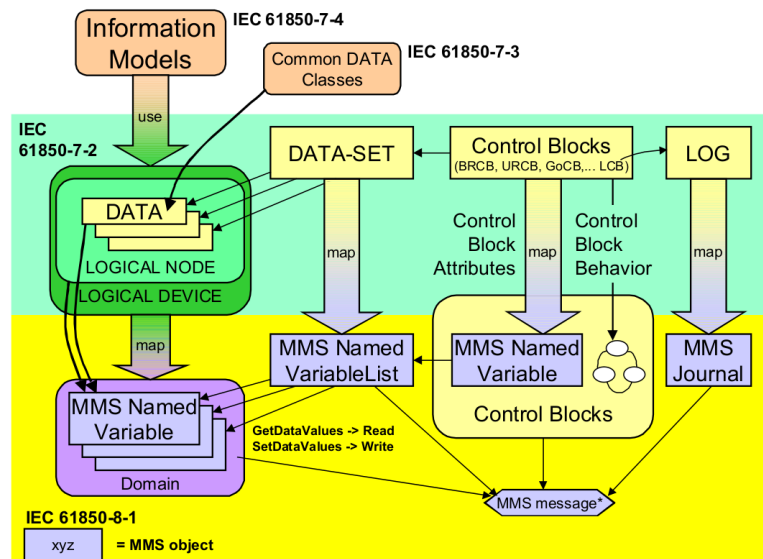
O modelo de dados definido pela ACSI segue o paradigma de orientação a objetos, principalmente ao definir um componente chamado de Dados, ou *Data*. O componente *Data* representa uma instância de informação com atributos e métodos. Seguindo o modelo de orientação a objetos, a norma estabelece que um componente *Data* pode estender classes comuns, que definem padrões encontrados com frequência em sistemas de automação de subestações. Dessa forma, é definido o conceito de Classe Comum de Dados, ou *Common Data Class* (CDC), abordado no módulo 7.3 da norma. Um exemplo claro da utilidade de definir-se CDC é o dado *Pos*, que indica o estado de um equipamento de disjunção, encontrado nos nós lógicos XCBR, XSWI, CSWI, que indicam disjuntor, controle de disjuntor e chave seccionadora automática, respectivamente, assim evita-se a necessidade de repetição dos mesmos atributos comuns. O componente *Data* também pode ter atributos de dados que herdaram de CDC e assim definem atributos compostos.

Por fim, o elemento mais interno do modelo de dados definido pela norma é o Atributo de Dado, ou *Data Attributes* (DA) que representa a informação propriamente dita. Por exemplo, em um LN MMXU, um atributo de dados da CDC poderia ser um valor analógico (*mag*) representado por um ponto flutuante (*f*).

Na Figura 5.3 são mostrados todos os elementos que compõem a ACSI, e como eles se relacionam, acompanhados de uma legenda que mostra em qual das subdivisões do módulo 7 da norma eles estão descritos.

Qualquer dado disponibilizado por um servidor IEC 61850 tem uma **referência** única,

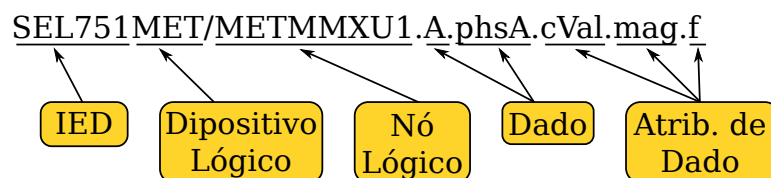
Figura 5.3 – Modelo de dados da ACSI descritos no módulo 7 da norma IEC 61850.



Fonte: (IEC/TR 61850-7-1, 2003)

que possibilita seu acesso de qualquer ponto da rede. Por exemplo, uma medida de corrente da fase A supervisionada por um Servidor IEC 61850, identificado pelo nome SEL\_751 e que está em um dispositivo lógico MET, pode ser solicitada através da seguinte referência: *SEL\_751\_MET/METMMXU1.A.phsA.cVal.mag.f*. Na Figura 5.4 é mostrado cada elemento da referência do dado de corrente.

Figura 5.4 – Modelo de referência de uma informação no padrão IEC 61850.



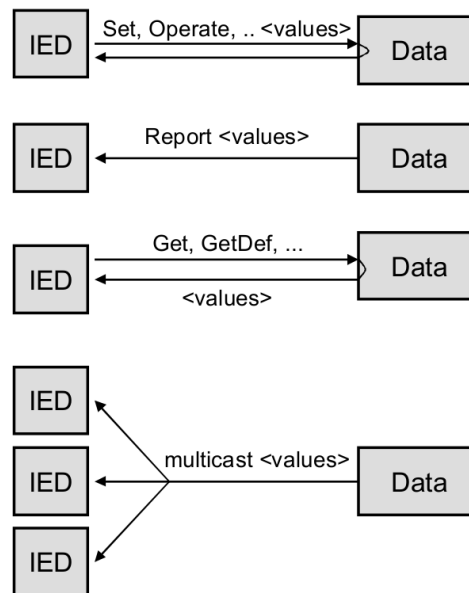
Fonte: Própria do autor.

Dois aspectos devem ser destacados na Figura 5.4, o primeiro diz respeito ao nome do nó lógico que na norma é padronizado por MMXU, mas na referência apresenta o nome de METMMXU1. O nome deste nó lógico apresenta um prefixo MET, e um sufixo 1, que são previstos por norma para poder diferenciar nós lógicos presentes num dispositivo lógico que tenham as mesmas características, mas que sirvam para funções diferentes. O segundo aspecto que pode ser destacado é a existência de mais de um dado e atributo de dado. Isso ocorre devido ao paradigma de orientação a objetos adotado pela norma, assim, a norma prevê a existência de dados e atributos de dados compostos, em que um dado pode ser composto de diversas instâncias de dados.

### 5.3.2 Modelo de Troca de Informações

Uma vez definido o modelo de informação padronizado, a norma IEC 61850 também define de que maneira estas informações podem ser trocadas entre os dispositivos que contêm as instâncias de dados. Na Figura 5.5 são mostradas todas as possíveis formas de comunicação estabelecidas pela norma.

Figura 5.5 – Possíveis formas de envio e recebimento de informação com IEC 61850.



Fonte: (IEC/TR 61850-7-1, 2003).

A maneira mais simples de realizar troca de informações entre dispositivos IEC 61850 é através dos serviços de **Get/Set-Operate** disponibilizados pelos servidores IEC 61850. Estes serviços são utilizados para adquirir (Get) e alterar dados ou comandar o IED (Set-Operate) e são realizados por meio do protocolo MMS.

MMS é um protocolo de camada de aplicação escolhido pelo comitê de desenvolvimento da norma IEC 61850 como a melhor opção para realizar a comunicação vertical entre IED e aplicações no nível de estação. Mais especificamente, o MMS é um padrão internacional (ISO 9506) para troca de mensagens em tempo real e para aquisição de dados e envio de comandos por sistemas supervisórios entre dispositivos e aplicações computacionais. Mensagens MMS são independentes das funções executadas pelos dispositivos comunicantes e da forma como o dispositivo ou aplicação foram desenvolvidos (MACKIEWICZ, 1994).

Portanto, o que a norma IEC 61850 faz é mapear o modelo de dados definido pela ACSI no protocolo MMS, de modo que os dispositivos comunicantes tenham um padrão de envio e aquisição de dados via ambiente de rede. Esse mapeamento é definido no módulo 8.1 da norma.

Serviços Get representam operações de coleta de dados por pooling, ou seja, caso algum sistema necessite supervisionar algum estado ou grandeza analógica, pode fazê-lo, por



meio de solicitações periódicas ao dispositivo que fornece o dado. Nem sempre esta é a maneira mais eficiente de realizar a supervisão de uma grandeza, por isso existe também a coleta de dados por demanda, em que o valor da variável supervisionada só será enviada ao sistema de supervisão, quando esta variável mudar de estado, no caso de grandezas digitais, e no caso de grandezas analógicas, quando esta ultrapassar uma faixa de valores especificado, chamado de banda morta. No IEC 61850 este tipo de supervisão é realizado por meio de um serviço específico, os *Relatórios*.

Relatórios podem ser criados em um servidor IEC 61850 e enviados a um cliente sempre que alguma das grandezas associadas ao relatório sofrer alterações. Entretanto, para que um relatório possa ser criado, é necessário que exista uma estrutura que agrupe os dados a serem anexados ao relatório. Esta estrutura é definida pela norma e tem o nome de *DataSet* (DS). Os DataSet agrupam dados e atributos de dados. DS são utilizados em mensagens MMS por meio dos relatórios e também podem ser associados a mensagens do tipo GOOSE.

Mensagens GOOSE são mensagens de extrema prioridade e com limitações de tempo máximo. As mensagens GOOSE devem ser trocadas entre IED, em uma espécie de comunicação horizontal, ao contrário das mensagens MMS que devem ser utilizadas para uma comunicação vertical. Devido suas limitações extremas de tempo, o protocolo GOOSE é um protocolo da camada de enlace, ao contrário do MMS que pertence à camada de aplicação.

Outra particularidade importante das mensagens GOOSE é que elas não seguem a arquitetura cliente/servidor, mas a arquitetura editor/assinante, que funciona da seguinte forma: o editor disponibiliza os dados para aqueles que assinam a informação, dessa forma, sempre que um editor envia uma mensagem no barramento de dados aqueles que a assinam a receberão.

O mecanismo de distribuição de mensagens GOOSE funciona da seguinte forma: primeiro, os assinantes devem conhecer as características da mensagem que devem assinar, como por exemplo, o número de identificação da mensagem e o endereço de máquina (MAC) do editor; dessa forma o assinante armazena as informações das mensagens que assina. As mensagens GOOSE são enviadas ao barramento de dados via broadcast, assim, todos os dispositivos que estão conectados na rede podem receber a mensagem. Por isso, ao receber a mensagem, o dispositivo deve verificar se a mensagem recebida contém as características de alguma das mensagens por ele assinadas.

Como o protocolo GOOSE é implementado na camada de enlace, não há estabelecimento de conexão entre os dispositivos comunicantes, e nem garantias de que a mensagem chegará aos respectivos assinantes. Como resolver esse problema, já que mensagens GOOSE são mensagens de alta prioridade e de grande importância?

A seguinte estratégia é adotada: todas as mensagens GOOSE têm um de seus parâmetros que se chama tempo máximo. Esse parâmetro estabelece a taxa de transmissão

das mensagens em estado normal de operação. Dessa forma as mensagens GOOSE estão sendo constantemente transmitidas na rede a uma taxa que obedece o parâmetro de tempo máximo. Caso algum dos assinantes não recebe a mensagem dentro deste intervalo, então o dispositivo sinaliza falha de comunicação.

Segundo Ozansoy (2010), quando um evento ocorre e altera algum dos dados do DS associado à mensagem GOOSE, então a taxa de transmissão das mensagens é alterada e as mensagens passam a ser transmitidas a uma taxa que obedece à Equação 5.1. Quando o valor de  $t$  ultrapassa o tempo máximo, a mensagem volta a ser transmitida periodicamente no intervalo de tempo estabelecido por este parâmetro.

$$t = (1 + n^{R-1}) \times 0,0001 \quad (5.1)$$

Em que:

$t$  representa o tempo de transmissão em relação à transmissão anterior;

$n$  é um fator de transmissão que pode ser escolhido entre 2 e 7;

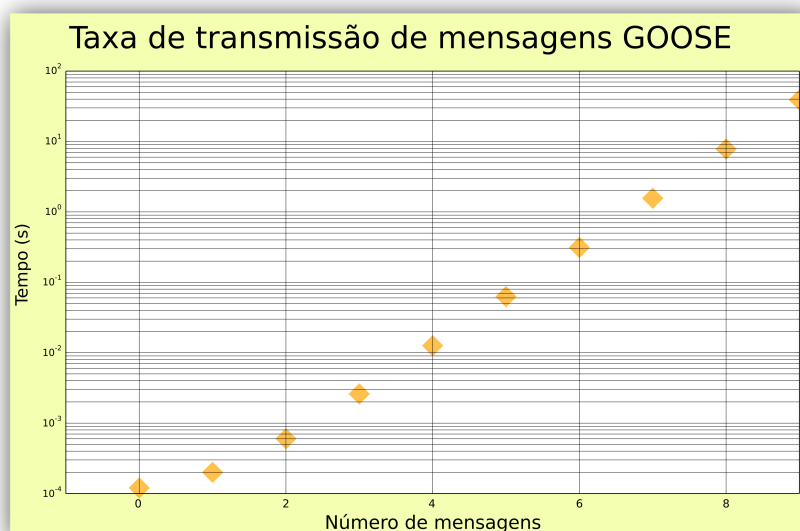
$R$  é o número da mensagem após a modificação dos valores do DS.

Na Figura 5.6 é mostrado um gráfico de tempo por mensagens transmitidas, em que o tempo de cada mensagem é calculado pela Equação 5.1 com o parâmetro  $n = 5$  e o parâmetro  $R$  sendo variado de acordo com os valores do eixo x. No gráfico, que possui o eixo y em escala logarítmica, é possível observar que as primeiras sete mensagens são transmitidas em menos de um segundo, enquanto a taxa de transmissão da oitava para a nona mensagens é de aproximadamente nove segundos.

Um último tipo de mensagem especificado pela norma IEC 61850 que merece destaque neste texto, é o *Sampled Values* (SV). Mensagens do tipo SV têm como propósito transportar informações de medidas realizadas pelos equipamentos no nível de processo aos IED que irão supervisionar e atuar no sistema caso alguma falha seja detectada. As SV são bem semelhantes às mensagens do tipo GOOSE, pois também seguem a arquitetura de editor/assinante e são mapeadas diretamente na camada de enlace, uma vez que também são mensagens de alta prioridade, já que levam as informações que servirão para, por exemplo, tomar a decisão de envio de TRIP para equipamentos de disjunção.

Na Figura 5.7 são mostrados, por meio de diagramas esquemáticos, os serviços de troca de dados disponibilizados pela norma IEC 61850. É possível observar na Figura 5.7 os serviços de comunicação disponibilizados pela ACSI, por meio das interações de requisição MMS, representadas pelas setas ligando o cliente aos dois servidores; pelos relatórios representados pelo bloco de relatório, ligando o servidor ao cliente, via MMS; e pelo serviço de comunicação multicast adotado pelas mensagens GOOSE e Sampled Values, que utilizam uma abordagem editor/assinante.

Figura 5.6 – Gráfico de tempo por mensagens GOOSE transmitidas após a modificação de um dos parâmetros do Data Set associado.



Fonte: Própria do autor.

## 5.4 Desenvolvimento de uma Biblioteca para Comunicação Via Protocolos de Comunicação IEC 61850

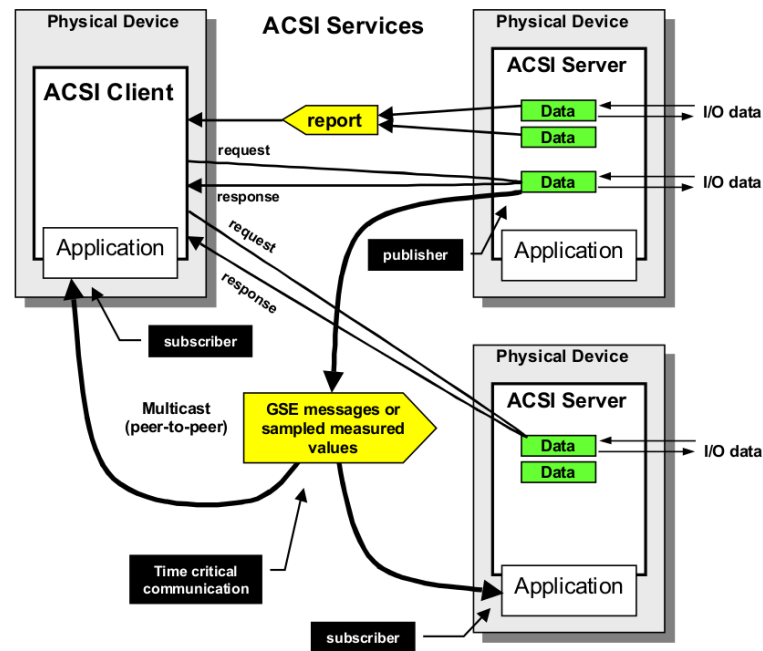
Para tornar possível a comunicação entre um sistema qualquer e os dispositivos que se comunicam via protocolos IEC 61850, é necessário que seja implementada uma biblioteca que seja capaz de oferecer os serviços estabelecidos pela ACSI. Essa biblioteca existe e está licenciada sob os termos da licença de código livre GPLv3, portanto, disponível a qualquer um que queira utilizá-la, modificá-la ou melhorá-la para fins não comerciais.

O nome desta biblioteca é `libiec61850` e sua implementação está em código escrito na linguagem C. Por ser a linguagem C de amplo uso em todas as áreas da computação, a linguagem Python, que também é escrita em C, possui uma série de facilidades para que módulos em C sejam traduzidos para Python sem grandes dificuldades.

Portanto, como o SMRA foi desenvolvido utilizando frameworks escritos em Python, as principais funcionalidades da biblioteca `libiec61850` foram traduzidas para métodos em Python. As funcionalidades traduzidas, até o momento foram:

- Estabelecimento e encerramento de conexão MMS;
- Descoberta de modelo de dados dos IED;
- Envio de comandos;
- Leitura de dados analógicos e digitais.

Figura 5.7 – Modelo de comunicação da norma IEC 61850.



Fonte: (IEC/TR 61850-7-1, 2003).

Estas são as funcionalidades essenciais para que possam ser realizadas as tarefas que um SMRA necessita, as quais são: recebimento e envio de informações, das grandezas supervisionadas, sejam elas analógicas ou digitais.

#### 5.4.1 Biblioteca libiec61850

O projeto libiec61850 (ZILLGITH, 2015) oferece implementações de uma biblioteca para os níveis de cliente e de servidor dos protocolos IEC 61850/MMS e IEC 61850/GOOSE que se baseia nos conceitos descritos pela ACSI. As funcionalidades disponibilizadas pela biblioteca são:

- Implementação dos protocolos MMS/GOOSE utilizando a pilha de protocolos TCP/IP;
- Implementação estática do modelo de dados do IED por meio da geração de código em C a partir de um arquivo em SCL;
- Implementação dinâmica do modelo de dados do IED por meio de chamadas de API ou por arquivo de configuração;
- Serviços de leitura e escrita tanto para variáveis MMS simples como para variáveis complexas;
- Implementação de serviços para identificação de modelo de dados (self-identification);

- Implementação de serviços para manipulação de Data Sets inclusive criação e exclusão dinâmica de Data Sets;
- Implementação das API de clientes e servidores IEC 61850;
- Implementação dos serviços de Relatórios;
- Suporte para todos os modelos de controle descritos pela IEC 61850;
- Implementação dos serviços GOOSE editor/assinante;
- Implementação de camada de abstração de hardware que permite a execução dos serviços disponibilizados pela biblioteca nas plataformas Linux, OS X e Windows;
- Ferramenta de conversão de arquivos SCL para modelos de dados estáticos e dinâmicos de IED;
- Possibilidade de funcionamento em sistemas embarcados, que possuem limitações de hardware.

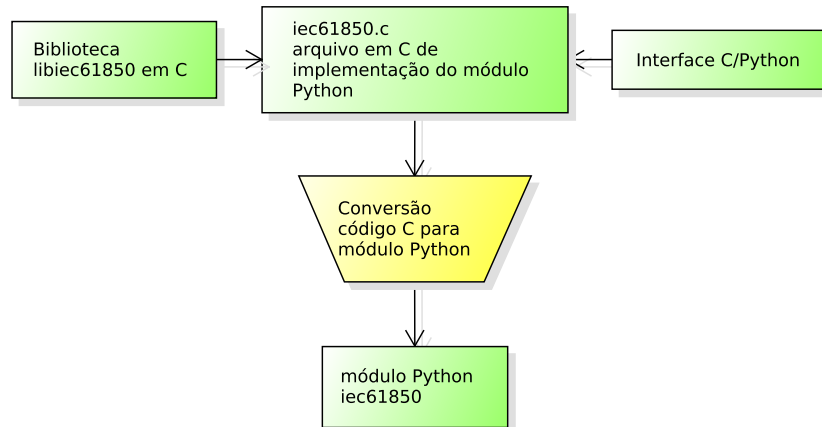
## 5.4.2 API IEC 61850 em Python

Como descrito na seção anterior, a biblioteca `libiec61850` oferece uma API com diversos recursos para implementação de várias formas de aplicações tanto cliente como servidore IEC 61850. Neste trabalho foram desenvolvidas algumas funcionalidades desta API na linguagem Python, de modo que os agentes pudessem enviar comandos para os IED e solicitar informações destes.

O processo de implementação da API IEC 61850 em Python está mostrado na Figura 5.8. Primeiro um arquivo em C é criado, o `iec61850.c`. Neste arquivo a API em Python é implementada utilizando o módulo Python para interface com C e utilizando também a API da biblioteca `libiec61850` em C. Em seguida, o arquivo `iec61850.c` é compilado e convertido para um módulo Python, através das ferramentas de interface C/Python da linguagem. Assim, a API IEC 61850 pode ser utilizada em programas Python normalmente, sem a necessidade de interação nenhuma do usuário com a implementação em C.

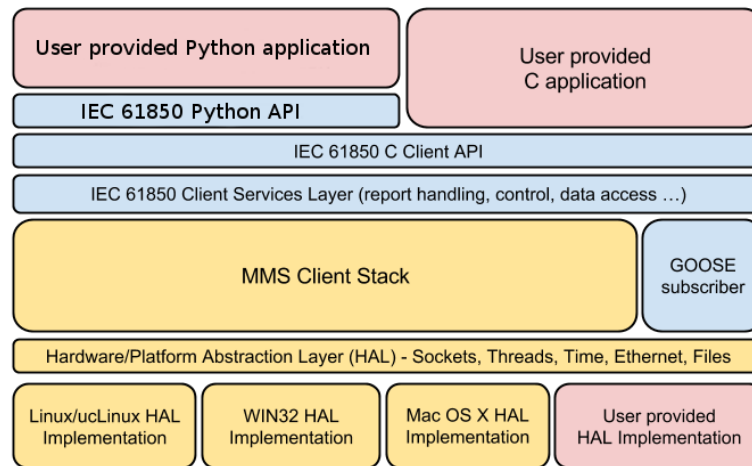
O diagrama da Figura 5.9 mostra a pilha com os módulos de implementação da biblioteca. No topo da pilha, no lado direito, estão as aplicações em C que utilizam as funcionalidades da API implementada pela biblioteca. Também no topo da pilha, no lado esquerdo, estão os dois módulos implementados durante o desenvolvimento deste trabalho, que permitem qualquer aplicação Python utilizar as funcionalidades da biblioteca. Uma das camadas, a mais inferior, implementa a API em Python com base na API em C; a camada superior representa uma aplicação genérica Python implementada com a API já convertida.

Figura 5.8 – Processo de conversão da biblioteca libiec61850 em C para módulo Python.



Fonte: Própria do autor.

Figura 5.9 – Pilha de implementação da biblioteca libiec61850



Fonte: Adaptado de (ZILLGITH, 2015).

No Apêndice C está a demonstração de código exclusivamente em Python e construído utilizando a API baseada em libiec61850, que faz a descoberta de todo o modelo de dados de um IED qualquer.

## 5.5 Conclusão

Neste capítulo foram mostrados os principais conceitos da norma IEC 61850 utilizados no trabalho para que a comunicação entre SMRA e IED de proteção e controle pudesse ser estabelecida. Os tópicos de maior interesse foram:

- o modelo de dados definido pela norma IEC 61850 que representa qualquer informação que deve ser utilizada pelo sistema de automação;

- a definição de uma ACSI que define os protocolos e as estratégias de comunicação entre dispositivos e aplicações.

Com base nestes dois conceitos, modelo de dados e ACSI, a comunicação bidirecional pode ser estabelecida entre IED e SMRA utilizando os protocolos GOOSE, para detecção de atuação da proteção, e MMS para envio de comandos. para a construção de uma biblioteca Python

---

# Capítulo 6

## Testes e Simulações

---

Neste Capítulo são apresentados os testes e simulações realizados com o SMRA. Numa primeira fase, os agentes foram embarcados em um laptop, um hardware Raspberry-Pi e um hardware BeagleBone Black. Os IED comunicantes via IEC 61850 são todos simulados no laptop por meio da utilização da biblioteca libiec61850. Os IED simulam o modelo de dados do relé numérico SEL-751 por meio de uma das funcionalidades da biblioteca libiec61850 que permite que o modelo de dados de um dispositivo seja carregado dinamicamente por meio de seu arquivo de configuração XML, o arquivo CID (*Configured IED Description*).

Na segunda fase de testes, são utilizados além dos IED virtualizados no laptop, também três IED SEL-751 reais e comunicando via protocolos IEC 61850, interconectados por meio de um switch de rede também da SEL. Aos IED estará conectada uma mala de testes Conprove CE-6006 que simula o secundário do TC na ocorrência de uma falta na rede elétrica.

### 6.1 Descrição da Rede Elétrica Utilizada nos casos de Teste

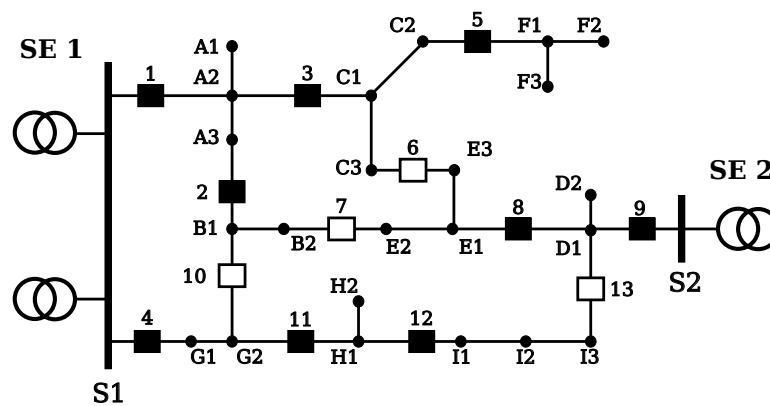
Os testes do SMRA são realizados levando em consideração a rede mostrada na Figura 6.1. Na Figura 6.1a é mostrada a representação detalhada de todos os setores, chaves, trechos e nós de carga da rede, na Figura 6.1b está uma representação simplificada da rede, que mostra os setores e chaves por meio de nós e arestas, respectivamente. As arestas tracejadas representam chaves abertas, enquanto as contínuas representam chaves fechadas.

A rede representada na Figura 6.1 possui algumas características que permitem a realização de testes diversos:

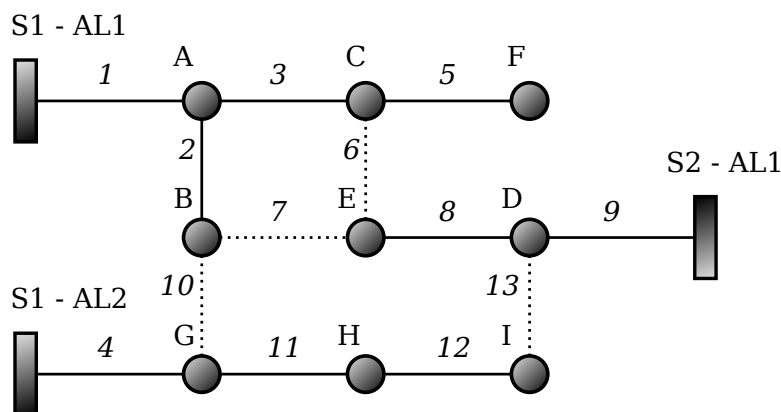


- presença de três alimentadores (S1\_AL1, S1\_AL2 e S2\_AL1), sendo dois da mesma subestação (S1) e um terceiro de outra subestação (S2);
- presença de chaves de encontro tanto entre alimentadores da mesma subestação quanto entre alimentadores de subestações diferentes;
- presença de setores com diferentes possibilidades de alimentação.

Figura 6.1 – Rede elétrica de distribuição considerada para realização dos testes.



(a) Representação detalhada da rede elétrica.



(b) Representação simplificada, baseada em grafos, dos setores, subestações e chaves da rede elétrica.

Fonte: Própria do autor.

Na Tabela 6.1 é mostrada a quantidade de consumidores atendidos em cada setor dos alimentadores das duas subestações presentes na rede de testes da Figura 6.1.

Em estado normal de operação, a situação de carregamento da rede é indicada pelas tabelas geradas pela ferramenta de fluxo de carga acoplada à API de representação da rede elétrica<sup>1</sup>. Todas as informações referentes às características da rede, como potência e nível de tensão dos nós de carga, tipo de condutor de cada trecho, comprimento dos trechos e carregamento dos condutores estão expostos no Apêndice D. Na Tabela 6.2 é

<sup>1</sup> O algoritmo de fluxo de carga utilizado para o cálculo das tensões nos nós de carga e do fluxo de corrente nos trechos está descrito em detalhes no Apêndice B.

Tabela 6.1 – Distribuição de consumidores por setor na rede teste.

| Subestação | Alimentador | Setor | Qtd. de consumidores |
|------------|-------------|-------|----------------------|
| S1         | S1-AL1      | A     | 300                  |
|            |             | B     | 200                  |
|            |             | C     | 350                  |
|            | S1-AL2      | F     | 400                  |
|            |             | G     | 350                  |
|            |             | H     | 300                  |
| S2         | S2-AL1      | I     | 250                  |
|            |             | D     | 200                  |
|            |             | E     | 300                  |

Fonte: Própria do autor.

mostrado um resumo com os valores mínimos, máximos e médios de carregamento e níveis de tensão ao longo dos alimentadores que compõem a rede teste, e também o carregamento das subestações.

Tabela 6.2 – Dados de carregamento e de queda de tensão da rede teste em condições normais.

| Subestação | Alimentador | Restrição                | Mínimo  | Médio   | Máximo  |
|------------|-------------|--------------------------|---------|---------|---------|
| S1         | S1-AL1      | Queda de Tensão(%)       | 1,1561  | 2,4958  | 3,8365  |
|            |             | Carreg. do Alim.(%)      | 7,1747  | 33,6064 | 94,5829 |
|            |             | Carreg. da Subestação(%) | 95,1969 | 95,1969 | 95,1969 |
|            | S1-AL2      | Queda de Tensão(%)       | 0,8111  | 2,7046  | 3,9014  |
|            |             | Carreg. do Alim.(%)      | 5,7604  | 39,9304 | 67,5261 |
|            |             | Carreg. da Subestação(%) | 95,1969 | 95,1969 | 95,1969 |
| S2         | S2-AL1      | Queda de Tensão(%)       | 0,5023  | 0,9288  | 1,2172  |
|            |             | Carreg. do Alim.(%)      | 5,4873  | 18,4994 | 34,4036 |
|            |             | Carreg. da Subestação(%) | 45,3063 | 45,3063 | 45,3063 |

Fonte: Própria do autor.

## 6.2 Montagem de uma Plataforma para Realização de Testes

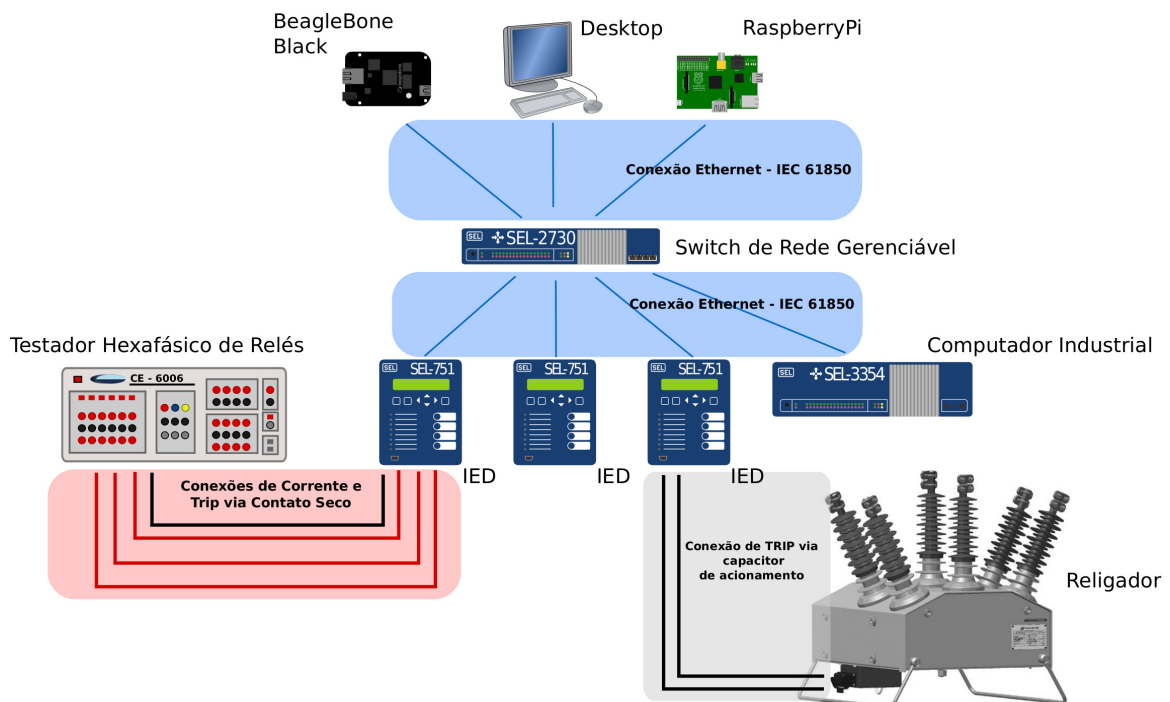
O Grupo de Redes Elétricas Inteligentes (GREI) da Universidade Federal do Ceará possui uma estrutura de dispositivos de automação que permite a montagem de uma plataforma de testes que simula, em muitos aspectos, as condições reais encontradas em uma subestação de distribuição de energia elétrica.

Entre os dispositivos presentes na plataforma estão:

- IED de proteção SEL-751, com possibilidade de se comunicar via protocolos estabelecidos pela norma IEC-61850;
- switch de rede gerenciável SEL-2730, com possibilidade de configuração de redes virtuais (VPN);
- computador industrial SEL-3354;
- testador Hexafásico de Relés Conprove CE-6006;
- religador Tavrida Electric modelo OSM/TEL-15.5-16/630-220;
- desktop usual;
- placa de desenvolvimento Raspberry-Pi modelo B;
- placa de desenvolvimento BeagleBone Black.

Na Figura 6.2 são mostrados todos os equipamentos utilizados na plataforma de testes e suas interligações de comunicação, via portas Ethernet IEC 61850 para transmissão de comandos e leitura de dados entre IED e dispositivos computacionais; entradas analógicas para transmissão de medidas de corrente entre o testador hexafásico e IED de proteção; e saídas digitais para o envio de comando entre IED de proteção e religador Tavrida.

Figura 6.2 – Topologia física de testes do SMRA.



Fonte: Própria do autor.

Os agentes foram embarcados nos dispositivos computacionais (desktop, computador industrial, Raspberry Pi e BeagleBone Black) simulando um ambiente computacional distribuído e conectado via rede Ethernet por meio do switch SEL-2730, conforme mostrado na Figura 6.2. Na Tabela 6.3 são mostrados os dispositivos que serão utilizados como contêineres de agentes e que farão parte da infraestrutura de execução do SMRA, integrados em rede Ethernet e comunicando-se via protocolos estabelecidos pela FIPA. Os IED que também fazem parte da infraestrutura de comunicação Ethernet, podendo receber comandos por meio do protocolo MMS e publicar mensagens utilizando o protocolo GOOSE, também estão listados na Tabela 6.3.

Tabela 6.3 – Dispositivos conectados em rede Ethernet para execução dos testes.

| Dispositivo      | Funcionalidade                     | Endereço IP  |
|------------------|------------------------------------|--------------|
| Relé SEL 751     | Dispositivo de proteção e controle | 192.168.1.51 |
| Relé SEL 751     | Dispositivo de proteção e controle | 192.168.1.52 |
| Relé SEL 751     | Dispositivo de proteção e controle | 192.168.1.53 |
| Laptop           | Contêiner de agentes               | 192.168.1.10 |
| Raspberry Pi     | Contêiner de agentes               | 192.168.1.11 |
| BeagleBone Black | Contêiner de agentes               | 192.168.1.12 |

Fonte: Própria do autor.

Levando em consideração a rede de testes mostrada na Figura 6.1, a plataforma de testes do SMRA terá três módulos de execução de agentes, sendo um por alimentador. Esta abordagem proporciona um ambiente de processamento distribuído para a execução dos agentes. Na Tabela 6.4 são mostrados os agentes presentes em cada módulo de execução, assim como o alimentador associado e seu identificador, que funciona como um endereço vinculado aos hardwares listados na Tabela 6.3.

Tabela 6.4 – Agentes presentes na plataforma de testes do SMRA.

| Alimentador | AID                      | Tipo        |
|-------------|--------------------------|-------------|
| S1-AL1      | S1_AL1@192.168.1.10:4001 | Alimentador |
|             | S1_AD1@192.168.1.10:4002 | Dispositivo |
| S1-AL2      | S1_AL2@192.168.1.11:4001 | Alimentador |
|             | S1_AD2@192.168.1.11:4002 | Dispositivo |
| S2-AL1      | S2_AL1@192.168.1.12:4001 | Alimentador |
|             | S2_AD1@192.168.1.12:4002 | Dispositivo |

Fonte: Própria do autor.

## 6.3 Descrição do Processo de Recomposição Automática

O procedimento para a realização dos testes do SMRA é iniciado com a simulação da atuação de algum dos elementos de proteção parametrizados no IED que controla algum dos religadores da rede. Quando o IED atua e envia comando de TRIP para o religador, um ou mais setores da rede são desenergizados e o sistema entra em estado de restauração.

Paralelamente à atuação da função de proteção do IED, uma mensagem GOOSE é gerada. A mensagem GOOSE possui um Data Set associado a ela com diversos parâmetros que podem ser úteis para implementação de lógicas de automação no sistema. Um desses parâmetros é o atributo de dado associado à atuação da função de proteção, que tem seu valor alterado assim que a função atua.

Essa mensagem é assinada pelo agente dispositivo responsável pelo alimentador onde a falta aconteceu. O agente recebe a mensagem e percebe a alteração no atributo de dado que indica a atuação da função de proteção.

Logo após a identificação da atuação da função de proteção e do TRIP do relé, o agente Dispositivo executa uma rotina de identificação das chaves para as quais será necessário enviar comando de abertura, de modo que o setor faltoso seja isolado do restante do sistema. Quando isso acontece a recomposição pode ser iniciada.

Uma mensagem REQUEST por meio do protocolo FIPA-Request é enviada para o agente Alimentador responsável pelo alimentador faltoso. O agente Alimentador recebe a mensagem e executa o protocolo FIPA-ContractNet. Este comportamento envia mensagens do tipo CFP para todos os agentes alimentadores presentes na plataforma solicitando por propostas de recomposição. Os agentes Alimentadores, recebem as mensagens, verificam a possibilidade de recomposição procurando por religadores de fronteira no alimentador.

Caso haja interligação entre os alimentadores, uma mensagem PROPOSE é retornada ao agente que solicita ajuda. A mensagem tem como conteúdo a disponibilidade da subestação do alimentador solicitado, que consiste na diferença entre a potência nominal dos transformadores da subestação a qual o agente que envia a proposta pertence e a potência consumida pelos alimentadores da mesma. Caso não haja interligação entre os alimentadores, uma mensagem REFUSE é enviada.

Ao receber todas as propostas, o agente Alimentador que solicitou ajuda irá optar inicialmente por mensagens vindas de alimentadores pertencentes à subestação da qual ele faz parte, caso elas existam. Em seguida, uma análise é realizada com base no valor de potência disponibilizada pelos agentes propositores. O agente que oferecer o maior montante de potência vencerá a competição e receberá uma mensagem ACCEPT-PROPOSE que autoriza o início das análises de recomposição; os demais agentes receberão mensagem do tipo REJECT-PROPOSE.

Ao receber a mensagem de autorização de recomposição, o agente Alimentador escolhido irá realizar uma análise detalhada da recomposição, que envolve os seguintes passos:

- Inserção virtual do ramo desenergizado ao alimentador receptor, por ordem de prioridade dos setores;
- Execução de um fluxo de carga no alimentador com o ramo a ser recomposto inserido;
- Análise de carregamento dos transformadores da subestação após a inserção do ramo;
- Análise de carregamento dos condutores de cada um dos trechos do alimentador;
- Análise de queda de tensão em cada um dos nós de carga do alimentador.

Caso alguma das análises de carregamento ou de queda de tensão gere resultado não satisfatório, então, o agente realiza operações de podas sucessivas no ramo inserido no alimentador, realizando a análise novamente cada vez que um setor é podado do ramo, até que uma das duas situações seja atingida:

- Todos os resultados das análises são satisfatórios;
- Não existem mais setores no ramo inserido.

No caso da análise retornar resultado positivo, então a recomposição é considerada válida e um comando de abertura é enviado caso o ramo necessite ser podado em algum ponto e logo em seguida o comando de fechamento é enviado realizando a recomposição.

Se ainda existirem ramos de alimentador para serem recompostos, então o agente Alimentador que possui os setores desenergizados, inicia novo procedimento de solicitação de propostas, excluindo dessa vez os alimentadores que já estão recompondo algum ramo do alimentador afetado.

## 6.4 Caso Teste

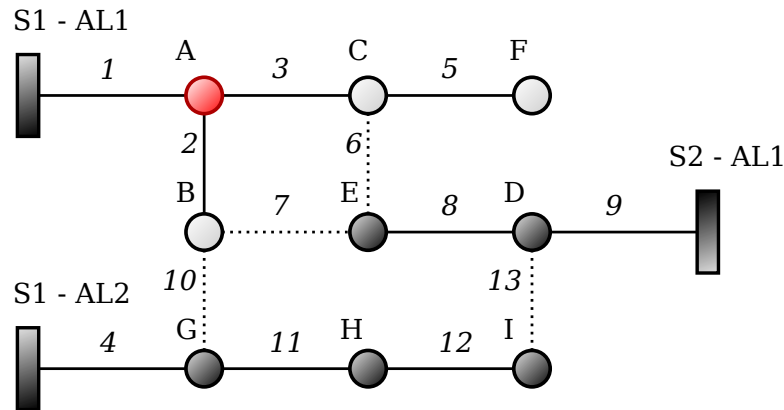
Um curto-circuito é considerado no setor A, levando à atuação da proteção de sobrecorrente do religador 1 e, conseqüentemente à desenergização de todo o alimentador S1-AL1, conforme mostrado na Figura 6.3

O processo de recomposição é inicializado com a identificação da falta por meio da análise da mensagem GOOSE pelo agente Dispositivo, que isola o setor sob falta, no caso o setor A, enviando comando de abertura para os dispositivos que controlam as chaves 2 e 3. A chave 1 já está aberta devido à atuação da proteção.

Após isolado o setor A, então uma mensagem é enviada ao agente S1\_AL1 que toma conhecimento do ocorrido e dá início ao processo de negociação de potências por meio do protocolo FIPA-ContractNet, enviando mensagens aos agentes S1\_AL2 e S2\_AL1.

Os agentes recebem a mensagem CFP e retornam com propostas. Na Figura 6.4 estão mostradas duas janelas gráficas com os dados das mensagens de propostas enviadas pelos agentes S1\_AL2 e S2\_AL1.

Figura 6.3 – Atuação da proteção sobre a chave 1.



Fonte: Própria do autor.

Figura 6.4 – Mensagens de proposta recebidas pelo agente S1\_AL1.

(a) Proposta enviada pelo agente S2\_AL1.

(b) Proposta enviada pelo agente S1\_AL2.

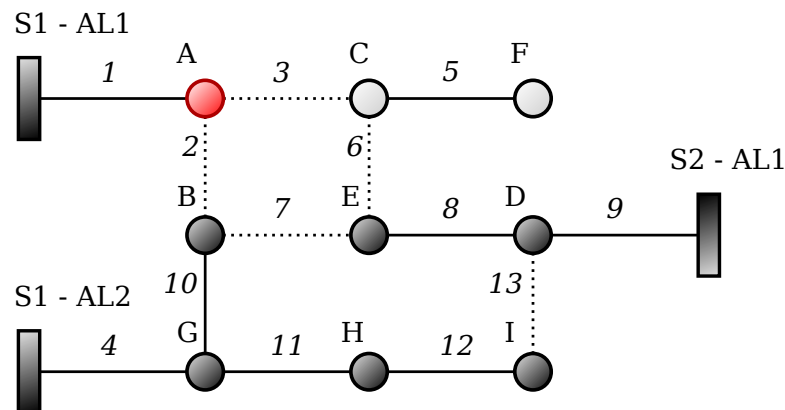
Fonte: Própria do autor.

Como pode ser observado na Figura 6.4, o agente S1\_AL2 disponibiliza uma potência de 6,0456 MVA, enquanto o agente S2\_AL1 disponibiliza uma potência de 2,7651 MVA. A quantidade de potência disponibilizada pelo agente S2\_AL1, além de ser menor que a do agente S1\_AL2, é proveniente de outra subestação. Dessa forma o agente S1\_AL2 possui prioridade no processo de seleção do agente que irá realizar a recomposição, sendo o escolhido para iniciá-la.

Logo que é selecionado, o agente S1\_AL2 inicia o processo de buscar quais dos ramos

podem ser recompostos por ele. Encontra então o ramo que contém apenas o setor B, que pode ser inserido no alimentador S1-AL2 através do setor G. O agente faz a inserção virtual do ramo em sua árvore de alimentador. A rede analisada pelo agente tem a topologia indicada pela Figura 6.5. Com base nesta topologia de rede o agente Alimentador executa o fluxo de carga para poder realizar as análises de carregamento e queda de tensão.

Figura 6.5 – Topologia da rede após a inserção do setor B no alimentador S1-AL2.



Fonte: Própria do autor.

Na Tabela 6.5 são mostrados os dados de carregamento e de queda de tensão mínimos, médios e máximos para a rede da Figura 6.5. Como pode ser visto, não há sobrecarga nos transformadores nem nos condutores da rede, também não há queda de tensão excessiva nos nós de carga. Dessa forma, a topologia é considerada válida e um comando de fechamento do religador 10 é enviado pelo agente Alimentador S1\_AL2 para o agente dispositivo.

Tabela 6.5 – Dados de carregamento e de queda de tensão da rede teste após a inserção do setor B no alimentador S1-AL2.

| Subestação | Alimentador | Restrição                | Mínimo  | Médio   | Máximo  |
|------------|-------------|--------------------------|---------|---------|---------|
| S1         | S1-AL2      | Queda de Tensão(%)       | 1,0411  | 2,9843  | 4,367   |
|            |             | Carreg. do Alim.(%)      | 5,7878  | 38,328  | 85,3711 |
|            |             | Carreg. da Subestação(%) | 60,1952 | 60,1952 | 60,1952 |

Fonte: Própria do autor.

Como o agente S1\_AL2 não pode recompor mais nenhum ramo desenergizado do alimentador S1\_AL1, então o processo de análise é encerrado e uma mensagem INFORM é enviada ao agente S1\_AL1 indicando que a recomposição foi finalizada pelo agente S1\_AL2.

Ao receber a mensagem INFORM, o agente S1\_AL1 verifica que ainda existe um ramo desenergizado, o ramo que contém os setores C e F. Assim, um novo processo de negociação é realizado, excluindo o agente Alimentador S1\_AL2, que já foi selecionado uma vez para fazer a recomposição. A única alternativa agora é o agente S2\_AL1, que responde com uma proposta de 2,7651 MVA, conforme mostrado na Figura 6.6.

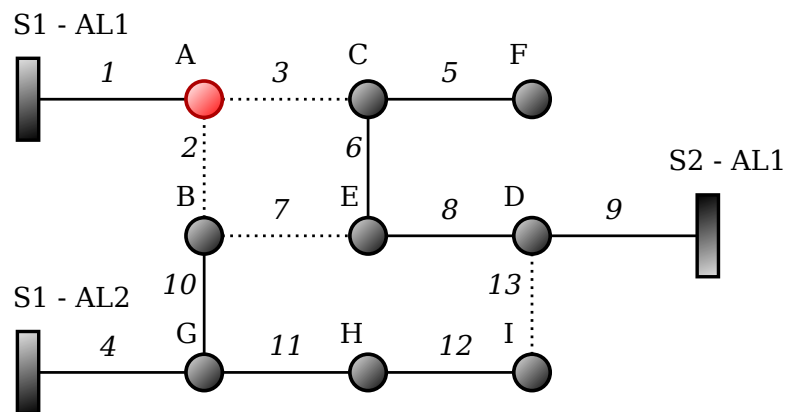


Figura 6.6 – Proposta enviada pelo agente S2\_AL1.

Fonte: Própria do autor.

Como não existe nenhuma outra proposta para ser analisada, o agente S2\_AL1 é escolhido para realizar a recomposição e inicia suas análises inserindo o ramo que contém os setores C e F no alimentador S2-AL1 através do setor E. Na Figura 6.7 é mostrada a rede analisada pelo agente S2\_AL1.

Figura 6.7 – Topologia da rede após a inserção do ramo C-F no alimentador S2-AL1.



Fonte: Própria do autor.

Após executar fluxo de carga, o agente obtém os resultados mostrados na Tabela 6.6. Como pode ser observado, existem violações de restrições em todos os níveis: sobrecarga de 29% nos transformadores da subestação, sobrecarga de 4% nos condutores da rede com carregamento e queda de tensão máxima de 8,9% nos nós de carga.

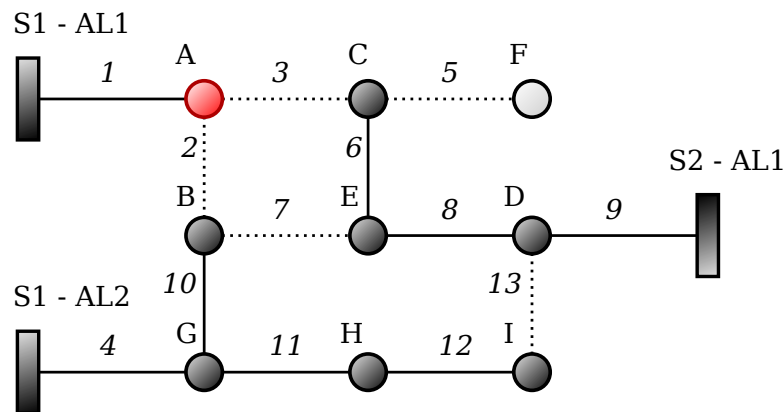
Tabela 6.6 – Dados de carregamento e de queda de tensão da rede teste após a inserção do ramo C-F no alimentador S2-AL1.

| Subestação | Alimentador | Restrição                | Mínimo   | Médio    | Máximo   |
|------------|-------------|--------------------------|----------|----------|----------|
| S2         | S2-AL1      | Queda de Tensão(%)       | 1,5284   | 6,0087   | 8,9261   |
|            |             | Carreg. do Alim.(%)      | 5,6606   | 52,4847  | 104,6134 |
|            |             | Carreg. da Subestação(%) | 129,3104 | 129,3104 | 129,3104 |

Fonte: Própria do autor.

Com base nestes resultados o agente S2\_AL1 realiza a poda do setor com maior profundidade no ramo inserido, no caso o setor F. Após a realização da poda do setor F, uma nova análise de fluxo de carga é executada na topologia de rede mostrada na Figura 6.8, cujos resultados são mostrados na Tabela 6.7.

Figura 6.8 – Topologia da rede após a poda do setor F do alimentador S2-AL1.



Fonte: Própria do autor.

Como nenhuma violação às restrições pode ser observada então a recomposição é realizada com a energização somente do setor C, excluindo portanto o setor F. Isso é feito quando o agente Alimentador S2\_AL1 envia mensagem ao agente Dispositivo solicitando abertura do religador 5 e fechamento do religador 6.

Tabela 6.7 – Dados de carregamento e de queda de tensão da rede após a poda do setor F do alimentador S2-AL1.

| Subestação | Alimentador | Restrição                | Mínimo  | Médio   | Máximo  |
|------------|-------------|--------------------------|---------|---------|---------|
| S2         | S2-AL1      | Queda de Tensão(%)       | 1,0277  | 3,0214  | 4,6608  |
|            |             | Carreg. do Alim.(%)      | 5,5749  | 37,3543 | 70,7682 |
|            |             | Carreg. da Subestação(%) | 90,6025 | 90,6025 | 90,6025 |

Fonte: Própria do autor.

Dessa forma, a topologia final de recomposição é a mostrada na Figura 6.8.

Após a inserção do setor C no alimentador S2-AL1, o processo de recomposição é finalizado, uma vez que não existem mais possibilidades de recomposição para o setor F que deverá permanecer desenergizado, até que a rede volte ao seu estado normal de operação.

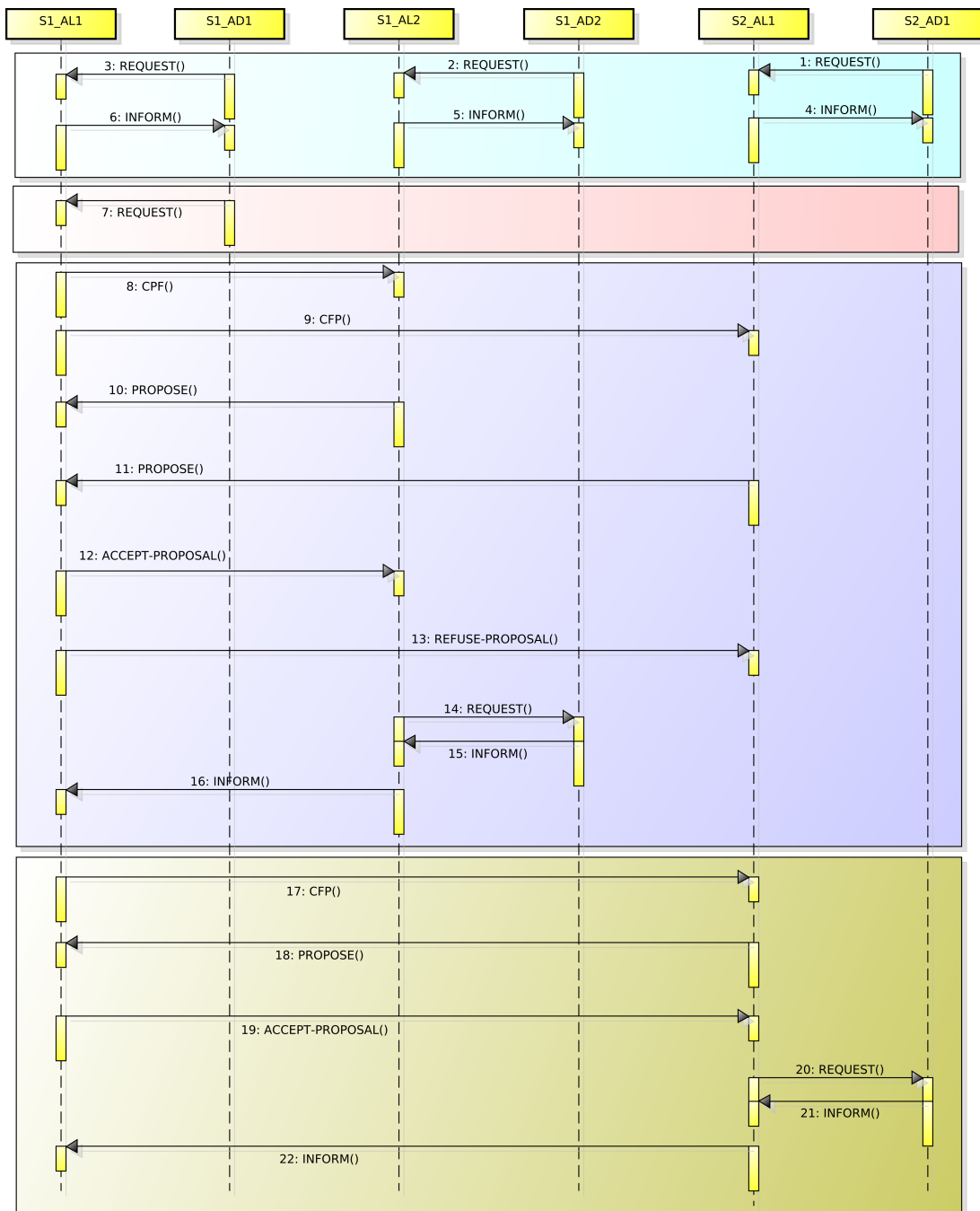
Todo o processo de comunicação entre os agentes é mostrado na Figura 6.9, que mostra um diagrama de sequência em quatro etapas: identificação dos AD junto ao AA; alerta do AD ao AA sobre ocorrência de falta no alimentador; primeira rodada de negociação entre agentes; e segunda rodada de comunicação entre agentes.

## 6.5 Resultados da Recomposição

De acordo com os dados mostrados na Tabela 6.1 1250 clientes foram desenergizados com a ocorrência da falta no setor A do alimentador S1-AL1. É possível realizar um levantamento a respeito do procedimento de recomposição da rede elétrica:

- 550 consumidores são reenergizados, o que representa 44% dos consumidores do alimentador S1-AL1;
- Apesar de estar são, o setor F não foi restaurado em decorrência de violação de restrições na rede elétrica, caso sua reenergização ocorresse pela subestação S2. A não possibilidade de recomposição do setor F representa 400 consumidores fora de serviço em decorrência da falta no setor A, isso representa 32% dos consumidores do alimentador S1-AL1;
- O tempo total para que todas as análises pudessem ser realizadas e o sistema recomposto foi de aproximadamente 10 s.

Figura 6.9 – Diagrama de comunicação entre agentes.



- Identificação dos agentes dispositivos junto aos agentes alimentadores
- Alerta de falta no alimentador
- Primeira fase de negociação entre agentes alimentadores e consolidação da proposta escolhida
- Segunda fase de negociação entre agentes alimentadores e consolidação da proposta escolhida

Fonte: Própria do autor.

---

## Capítulo 7

# Conclusões, Contribuições e Trabalhos Futuros

---

### 7.1 Conclusões

Neste trabalho foi apresentada uma abordagem para a recomposição automática de alimentadores de distribuição de energia elétrica, com topologia radial com recurso, utilizando a técnica de sistemas multiagentes para realizar os procedimentos de identificação, isolamento e restauração da porção da rede desenergizada que não contém trechos defeituosos. Para a recomposição é levado em consideração aspectos de restrição operacional, como sobrecarga de transformadores das subestações e de condutores inferior a 100%, assim como, queda de tensão inferior a 7% ao longo dos alimentadores.

Estes objetivos foram alcançados por meio da implementação de algoritmos de análise computacional executados por agentes em um ambiente de comunicação distribuído, onde os agentes foram embarcados em plataformas de hardware modulares e conectados via rede Ethernet.

Um enfoque especial foi dado à construção de uma plataforma de simulação que se aproximasse o máximo possível de uma situação real. Foram utilizados dispositivos reais de proteção e controle (IED) com capacidade de comunicação via protocolos estabelecidos pela norma IEC 61850 (MMS e GOOSE). Os agentes foram adicionados de módulos de comunicação MMS e GOOSE e a função de *self-healing* foi implementada.

A topologia proposta para a implementação do SMRA tem uma característica importante para sistemas que visam qualidade e eficiência: extensibilidade. Dessa forma foi utilizada uma estrutura de dados computacional, baseada na teoria dos grafos, para a representação dos alimentadores da rede elétrica de distribuição radial. Esta estrutura

denominada RNP apresenta aspectos importantes que justificam sua utilização neste trabalho, tais como eficiência computacional, representação simples, fácil entendimento de seus conceitos e definição de operadores de poda e inserção que facilitam a manipulação das árvores de grafos que representam a rede elétrica. Dessa forma, a teoria dos grafos permite que qualquer topologia de rede radial seja representada e manipulada pelos agentes.

O sistema multiagente foi desenvolvido utilizando o conceito de orientação a objeto utilizando a linguagem de programação Python e uma plataforma para execução de agentes em ambiente distribuído que também foi desenvolvido para atender aos objetivos deste trabalho. O framework tem o nome de Python Agent DEvelopment (PADE).

O modelo adotado de sistemas multiagentes para recomposição de alimentadores foi definido com dois tipos de agentes: agente Alimentador e agente Dispositivo. Para cada alimentador da rede é proposto um hardware com agente Alimentador e agente Dispositivo, responsáveis por esta porção do sistema. Ao agente Dispositivo cabe detectar ocorrências de defeito no alimentador, capturando mensagens GOOSE e analisando seus parâmetros, isolar o setor faltoso, informar a ocorrência ao agente Alimentador e enviar comandos ao IED via protocolo de comunicação MMS. Ao agente Alimentador cabem as funções de verificar quantos setores foram atingidos pela falta, calcular a potência necessária à recomposição, solicitar ajuda de outros agentes para restaurar a rede, realizar as análises de restrições operacionais e enviar comandos de fechamento e abertura das chaves necessárias para efetivar a recomposição.

Foram realizados testes com IED reais e agentes embarcados nas placas de desenvolvimento Raspberry Pi e BeagleBone Black, utilizando para isso a infraestrutura de testes do Grupo de Redes Elétrica Inteligentes (GREI) da Universidade Federal do Ceará (UFC), que conta com mala de testes de relés, equipamento que gera sinais de corrente e tensão nas entradas analógicas dos IED e simula condições normais de funcionamento e também situações diversas de faltas, com possibilidade de detectar atuação do IED tanto por contato seco de tensão quanto por mensagens GOOSE; IED SEL-751 com possibilidade de comunicação via protocolos IEC 61850; e religador de média tensão associado a um dos relés SEL-751.

Para a realização dos testes uma rede simples com duas subestações e três alimentadores foi modelada. Faltas foram simuladas para que as análises de recomposição pudessem ser realizadas. Os resultados obtidos foram satisfatórios comprovando a eficácia do SMRA desenvolvido.

## 7.2 Contribuições

Para que este trabalho alcançasse seus objetivos, muitos componentes tiveram que ser desenvolvidos, os quais formam a base de contribuições obtidas com o desenvolvimento da pesquisa em sistemas de recomposição automática para redes elétricas. As principais

contribuições podem ser citadas:

1. Desenvolvimento de uma estratégia de recomposição de alimentadores de distribuição de energia baseada em sistemas multiagentes com uma topologia que utiliza dois tipos de agentes: Alimentador e Dispositivo, sendo um AA e um AD responsáveis por um alimentador para tomada de decisão para a recomposição;
2. Desenvolvimento de um procedimento de análise de restrições operacionais que avalia carregamento dos transformadores e condutores ao longo da rede bem como queda de tensão nas cargas do alimentador;
3. Desenvolvimento de uma plataforma de desenvolvimento, execução e gerenciamento de sistemas multiagentes escrita em linguagem Python, o PADE;
4. Desenvolvimento de uma API de representação da rede elétrica com estrutura orientada a objetos e classes que modelam componentes de uma rede de distribuição de energia em média tensão e baseada em uma representação de grafos chamada RNP;
5. Desenvolvimento de um algoritmo de fluxo de carga de varredura direta-inversa associado à API de representação da rede elétrica;
6. Desenvolvimento de um componente de software que permite acessar os protocolos IEC 61850, MMS e GOOSE, em Python e baseado na biblioteca `libiec61850`, desenvolvida em linguagem C;
7. Estabelecimento de um procedimento de teste e validação do SMRA, que envolve IED comerciais SEL-751, switch de rede SEL-2730, mala de testes Conprove CE-6006, religador Tavrida Electric modelo OSM/TEL-15.5-16/630-220, hardwares embarcados Raspberry-Pi e BeagleBone-Black e computador de testes, os três últimos executando os agentes e simulando a comunicação dos IED que não estão representados fisicamente.

## 7.3 **Trabalhos Futuros**

O sistema desenvolvido neste trabalho produz resultados satisfatórios, mas existem muitos pontos que precisam ser melhorados e novas funcionalidades a serem agregadas, algumas delas são:

1. Verificação de descoordenação da proteção no momento da ocorrência de uma falta que dá início ao processo de recomposição da rede elétrica;
2. Estabelecer procedimento de análise para recomposição de alimentadores, após perda da subestação;

3. Associar o sistema de recomposição automática com um sistema de proteção adaptativa, essencial para deixar o sistema coordenado após uma recomposição;
4. Desenvolver uma estratégia de recomposição que associe sistemas multiagentes com métodos de otimização, como por exemplo algoritmos genéticos ou redes neurais artificiais;
5. Estender a API de representação da rede elétrica para sistemas com maior nível de tensão e com topologia em anel fechado;
6. Inserir geração distribuída nas análises de fluxo de carga e nas análises do SMRA, verificando seu impacto durante e após a recomposição;
7. Continuar o desenvolvimento da ferramenta PADE, corrigindo possíveis falhas, afim de torná-la cada vez mais estável, inserindo funcionalidades e melhorando sua usabilidade;
8. Realizar testes em redes com um número maior de nós de carga;
9. Desenvolver uma estratégia de aquisição de dados em tempo real dos IED da rede elétrica;
10. Adicionar comportamentos aos agentes que tratem os seguintes problemas: ocorrência de faltas múltiplas e restabelecimento da topologia normal do sistema após normalização da rede.

## 7.4 Trabalho Publicado

MELO, L. S.; SAMPAIO, R. F.; BARROSO, G. C.; LEO, R. P. S.; BEZERRA, J. R.; MEDEIROS, E. B. de. Modelagem e implementação de um sistema de recomposição automática usando multiagentes. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, XX., 2014, Belo Horizonte. *Anais...* Belo Horizonte: SBA, 2014.



---

## Referências

---

ADAFRUIT. *BeagleBone Black Rev. B*. 2015. Disponível em: <<https://www.adafruit.com/products/1278>>. Acesso em: 4. ago. 2015. 10

ADAFRUIT. *Raspberry Pi Model B*. 2015. Disponível em: <<https://www.adafruit.com/products/998>>. Acesso em: 4. ago. 2015. 10

BARROS, J. V. C. *Sistema Multiagente para Recomposição Automática de Subestação e Redes de Distribuição de Energia Elétrica*. Dissertação (Dissertação) — Departamento de Engenharia Elétrica, Universidade Federal do Ceará - UFC, 2013. vii, 13, 15, 16

BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. *Developing Multi-Agent Systems with JADE*. [S.l.]: Wiley, 2007. ISBN 0470057475. 19, 22, 32

CENTRO DE GESTAO E ESTUDOS ESTRATEGICOS. *Redes elétricas inteligentes: contexto nacional*. 2012. Disponível em: <<http://www.cgee.org.br>>. Acesso em: [www.cgee.org.br/atividades/redirect/8050](http://www.cgee.org.br/atividades/redirect/8050). 5, 6

CURCIC, S.; ÖZVEREN, C.; CROWE, L.; LO, P. Electric power distribution network restoration: a survey of papers and a review of the restoration problem. *Electric Power Systems Research*, v. 35, n. 2, p. 73 – 86, 1995. ISSN 0378-7796. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0378779695009914>>. 47, 48, 49

DELBEM, A.; CARVALHO, A. de; POLICASTRO, C.; PINTO, A.; HONDA, K.; GARCIA, A. Node-depth encoding for evolutionary algorithms applied to network design. In: DEB, K. (Ed.). *Genetic and Evolutionary Computation – GECCO 2004*. Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3102). p. 678–687. ISBN 978-3-540-22344-3. Disponível em: <[http://dx.doi.org/10.1007/978-3-540-24854-5\\_70](http://dx.doi.org/10.1007/978-3-540-24854-5_70)>. 9, 50, 52

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00023*: Fipa agent management specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00023/>>. Acesso em: 09 fev. 2015. 25, 27, 37, 38

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00026*: Fipa request interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00023/>>. Acesso em: 09 fev. 2015. 27, 37

- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00027*: Fipa query interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00027/>>. Acesso em: 09 fev. 2015. 27
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00028*: Fipa request when interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00028/>>. Acesso em: 09 fev. 2015. 27
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00029*: Fipa contract net interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00029/>>. Acesso em: 09 fev. 2015. 27, 28, 37
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00030*: Fipa iterated contract net interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00030/>>. Acesso em: 09 fev. 2015. 27
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00033*: Fipa brokering interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00033/>>. Acesso em: 09 fev. 2015. 27
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00034*: Fipa recruiting interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00034/>>. Acesso em: 09 fev. 2015. 27
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00035*: Fipa subscribe interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00035/>>. Acesso em: 09 fev. 2015. 27, 37
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00036*: Fipa propose interaction protocol specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00036/>>. Acesso em: 09 fev. 2015. 27
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00037*: Fipa communicative act library specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00037/>>. Acesso em: 09 fev. 2015. 24
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00061*: Fipa acl message structure specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00061/>>. Acesso em: 09 fev. 2015. 25, 37
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00070*: Fipa acl message representation in string specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00070/>>. Acesso em: 09 fev. 2015. 24, 37
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *FIPA 00071*: Fipa acl message representation in xml specification. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00071/>>. Acesso em: 09 fev. 2015. 25, 37
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. *The Foundation for Intelligent Physical Agents standards*. 2015. Disponível em: <<http://www.fipa.org/bdt/acarosp>>. Acesso em: 20 jan. 2015. 22

INTERNATIONAL ELECTROTECHNICAL COMMISSION. *Parte 1: Iec 61850-1: Communication networks and systems in substations, part 1: introduction and overview*. [S.l.], 2003. Disponível em: <<http://webstore.iec.ch/webstore/webstore.nsf/artnum/033549!opendocument>>. 72

INTERNATIONAL ELECTROTECHNICAL COMMISSION. *Parte 7-1: Iec 61850-7-1: Communication networks and systems in substations, part: 7-1: Basic communication structure for substation and feeder equipment – principles and models*. [S.l.], 2003. Disponível em: <<http://webstore.iec.ch/webstore/webstore.nsf/artnum/033549!opendocument>>. 73, 74, 76, 77, 81

INTERNATIONAL ELECTROTECHNICAL COMMISSION. *Parte 7-2: Iec 61850-7-2: Communication networks and systems in substations, part 7-2: Basic communication structure for substation and feeder equipment – abstract communication service interface (acsi)*. [S.l.], 2003. Disponível em: <<http://webstore.iec.ch/webstore/webstore.nsf/artnum/033549!opendocument>>. 73, 74

KERANEN, R. *4 Ways to Pick Your First Programming Language*. 2015. Disponível em: <<http://blog.udacity.com/2015/05/pick-your-first-programming-language.html>>. Acesso em: 4. ago. 2015. 31

KIMURA, S.; ROTTA, A.; ABDOD, R.; MORAES, R.; ZANIRATO, E.; BAHIA, J. Aplicação do iec 61850 no mundo real: projeto de modernização de 30 subestações elétricas. 2008. Disponível em: <[http://www.selinc.com.br/art\\_tecnicos/IEC61850\\_6308.pdf](http://www.selinc.com.br/art_tecnicos/IEC61850_6308.pdf)>. Acesso em: 18. mai. 2015. 1, 71

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. 3.ed. ed. [S.l.]: Perason, 2006. ISBN 85-88639-18-1. 27

LOPES, Y.; FRANCO, R. H. F.; MOLANO, D. A.; SANTOS, M. A. dos; CALHAU, F. G.; BASTOS, C. A. M.; MASRTINS, J. S. B.; FERNANDES, N. C. Smart grid e iec 61850: novos desafios em redes e telecomunicações para o sistema elétrico. Brasília, Brazil, setembro 2012. Disponível em: <[http://sbprt.org.br/sbprt2012/publicacoes/99346\\_1.pdf](http://sbprt.org.br/sbprt2012/publicacoes/99346_1.pdf)>. Acesso em: 18. mai. 2015. 5, 71

MACKIEWICZ, R. *An Overview to the Manufacturing Message Specification*. 1994. Disponível em: <[http://www.sztaki.hu/~drozdik/MMS/store/mms\\_intr.txt](http://www.sztaki.hu/~drozdik/MMS/store/mms_intr.txt)>. Acesso em: 15 mai. 2015. 77

MARQUES, L. T. *Restabelecimento de Energia por Reconfiguração de Redes em Sistemas de Distribuição de Grande Porte com Priorização de Chaves, Consumidores e Definição de Sequência de Chaveamento*. Dissertação (Dissertação) — Departamento de Engenharia Elétrica e de Computação da Escola de Engenharia de São Carlos, Universidade de São Paulo - USP, 2013. 2, 50, 53, 54, 55

MARQUES, L. T.; ZAN, J. cesar; DELBEM, A. C. B.; JUNIOR, J. B. A. L. Restabelecimento de energia em sistemas de distribuição de grande porte com priorização de chaves. In: CONGRESSO BRASILEIRO DE AUTOMATICA, XX., 2014, Belo Horizonte. *Anais...* Belo Horizonte: SBA, 2014. p. 280–287. 5, 51

MCARTHUR, S.; DAVIDSON, E.; CATTERSON, V.; DIMEAS, A.; HATZIARGYRIOU, N.; PONCI, F.; FUNABASHI, T. Multi-agent systems for power engineering applications

- part ii: Technologies, standards, and tools for building multi-agent systems. *Power Systems, IEEE Transactions on*, v. 22, n. 4, p. 1753–1759, Nov 2007. ISSN 0885-8950. 7, 12
- MCARTHUR, S. D. J.; DAVIDSON, E. M.; CATTERSON, V. M.; DIMEAS, A. L.; HATZIARGYRIOU, N. D.; PONCI, F.; FUNABASHI, T. Multi-agent systems for power engineering applications - part i: Concepts, approaches and technical challenges. *IEEE J P WRS*, 2007. 7, 12
- MCKELLAR, J.; FETTIG, A. *Twisted Network Programming Essentials*. [S.l.]: O'Reilly Media, 2013. ISBN 1449326110. 32, 33, 34, 35
- MCMORRAN, A. W. *An Introduction to IEC 61970-301 & 61968-11: The common information model*. [S.l.], 2007. 21, 22
- MELO, L. S. *Abordagem para Desenvolvimento de Sistema de Recomposição Automática para Rede de Distribuição de Energia Baseado em Sistemas Multiagentes*. Dissertação (Monografia) — Curso de Engenharia Elétrica, Campus de Sobral, Universidade Federal do Ceará - UFC, 2013. vii, 5, 14, 15, 16
- MELO, L. S.; SAMPAIO, R. F.; BARROSO, G. C.; LEAO, R. P. S.; BEZERRA, J. R.; MEDEIROS, E. B. de. Modelagem e implementação de um sistema de recomposição automática usando multiagentes. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, XX., 2014, Belo Horizonte. *Anais...* Belo Horizonte: SBA, 2014. 101
- MOHAGHEGHI, S.; MOUSAVI, M.; STOUPIIS, J.; WANG, Z. Modeling distribution automation system components using iec 61850. In: *Power Energy Society General Meeting, 2009. PES '09. IEEE*. [S.l.: s.n.], 2009. p. 1–6. ISSN 1944-9925. 71
- MOMOH, J. *Smart Grid: Fundamentals of design and analysis*. [S.l.]: John Wiley and Sons, 2012. 1
- NAGATA, T.; SASAKI, H. A multi-agent approach to power system restoration. *Power Systems, IEEE Transactions on*, v. 17, n. 2, p. 457–462, May 2002. ISSN 0885-8950. 12, 16
- NAGATA, T.; TAO, Y.; SASAKI, H.; FUJITA, H. A multiagent approach to distribution system restoration. In: *Power Engineering Society General Meeting, 2003, IEEE*. [S.l.: s.n.], 2003. v. 2, p. –660 Vol. 2. 12, 13, 14, 16
- OZANSOY, C. *Modelling and Object Oriented Implementation of IEC 61850: The New International Standard on Substation Communications and Automation*. [S.l.]: LAP LAMBERT Academic Publishing, 2010. ISBN 3838358481. 71, 75, 79
- SAMPAIO, R. F. *Sistema de Diagnóstico de Falhas para Subestações Baseado em Redes de Preti Coloridas*. Dissertação (Dissertação) — Departamento de Engenharia Elétrica, Universidade Federal do Ceará - UFC, 2002. 48
- SANTOS, A.; NANNI, M.; MANSOUR, M.; DELBEM, A.; LONDON, J.; BRETAS, N. *A power flow method computationally efficient for large-scale distribution systems*. 2008. 1-6 p. 114
- SCHWEITZER ENGINEERING LABORATORIES, INC. *SEL-751 Feeder Protection Relay: Instruction manual*. [S.l.], 2013. 70

- SOLANKI, J.; KHUSHALANI, S.; SCHULZ, N. A multi-agent solution to distribution systems restoration. *Power Systems, IEEE Transactions on*, v. 22, n. 3, p. 1026–1034, Aug 2007. ISSN 0885-8950. vii, 1, 13, 14, 16
- SUDHAKAR, T. D.; SRINIVAS, K. N. Restoration of power network – a bibliographic survey. *European Transactions on Electrical Power*, John Wiley Sons, Ltd., v. 21, n. 1, p. 635–655, 2011. ISSN 1546-3109. Disponível em: <<http://dx.doi.org/10.1002/etep.467>>. 6
- VIANA, G. V. R. *Meta-Heurísticas e Programação Paralela em Otimização Combinatória*. [S.l.]: UFC Edições, 1998. 50
- WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. [S.l.]: Wiley, 2009. ISBN 0470519460. 19, 20, 21, 22, 23, 29
- WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: Theory and practice. *Knowledge engineering review*, Cambridge [England]; New York, NY: Cambridge, v. 10, n. 2, p. 115–152, 1995. 20
- XU, L.; CHOW, M.-Y.; TAYLOR, L. Data mining and analysis of tree-caused faults in power distribution systems. In: *Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES*. [S.l.: s.n.], 2006. p. 1221–1227. 1, 2
- ZIDAN, A.; EL-SAADANY, E.; CHAAR, L. E. A cooperative agent-based architecture for self-healing distributed power systems. In: *Innovations in Information Technology (IIT), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 100–105. 3, 6
- ZILLGITH, M. *libiec61850*. 2015. Disponível em: <<http://libiec61850.com/libiec61850/>>. Acesso em: 15. mai. 2015. 81, 83

# Apêndices

---

# APÊNDICE A

## XML de representação da rede elétrica

---

Este apêndice contém um exemplo do arquivo XML que contém a descrição da rede elétrica, utilizada como caso base neste trabalho, a ser carregada pelos elementos da API de representação da rede elétrica.

O arquivo XML está dividido em três sessões principais que estão dentro da sessão *rede*, a sessão raiz do arquivo definida pela *tag* `<rede></rede>`.

A primeira das três sessões principais é a sessão *elementos*, definida pela *tag* `<elementos></elementos>`. Nesta sessão são enumerados os elementos passíveis de representação pela API da rede elétrica e descritas suas características individuais. Os elementos representados na sessão *elementos* são:

- chave;
- nó;
- condutor;
- trecho;
- setor;
- alimentador;
- transformador;
- subestação.

Na seção `elementos` não há representação das associações existentes entre elementos, como, por exemplo, associações de vizinhança e de conteúdo. Essas associações são definidas na sessão `topologia`, definida pela tag `<topologia></topologia>`.

Por último existe a sessão de comunicação, definida pela tag `<comunicacao></comunicacao>`, que define os endereços IP dos IED que controlam os elementos de disjunção do sistema.

Como o arquivo XML que descreve os elementos da rede utilizada como caso base neste trabalho possui um número muito elevado de linhas para serem inseridas neste apêndice, está representado somente um tipo de elemento por categoria em cada sessão do arquivo XML. Por exemplo, na sessão `elementos` só está representado um elemento chave, um elemento nó, um elemento setor, e assim por diante. O mesmo aplica-se às sessões `topologia` e `comunicacao`.

[htb]

```
<?xml version="1.0" encoding="UTF-8"?>

<!-->
*****
** Informacoes a respeito da estrutura da rede
*****
</!-->
<rede>
  <!-->
  *****
  ** Elementos que compoem a rede
  *****
  </!-->
  <elementos>
    <chave nome="1" estado="fechado"/>

    <no nome="S1">
      <potencia tipo="ativa" multip="k" unid="W">0.0</potencia>
      <potencia tipo="reativa" multip="k" unid="VAr">0.0</potencia>
    </no>

    <condutor nome="CAA 266R" rp="0.2391" xp="0.37895" ampacidade="301"/>

    <trecho nome="S1CH1">
      <comprimento multip="k" unid="m">0.01</comprimento>
    </trecho>

    <setor nome="S1" prioridade="0"/>

    <alimentador nome="S1_AL1"/>

    <transformador nome="S1_T1">
      <potencia tipo="aparente" multip="M" unid="VA">6.0</potencia>

      <impedancia tipo="seq_pos">
        <resistencia multip="" unid="ohms">10.0</resistencia>
        <reatancia multip="" unid="ohms">3.0</reatancia>
      </impedancia>

      <impedancia tipo="seq_zero">
        <resistencia multip="" unid="ohms">10.0</resistencia>
        <reatancia multip="" unid="ohms">3.0</reatancia>
      </impedancia>
    </transformador>
  </elementos>
</rede>
```



```

    </impedancia>

    <enrolamento tipo="primario">
      <tensao multip="k" unid="V">69</tensao>
    </enrolamento>

    <enrolamento tipo="secundario">
      <tensao multip="k" unid="V">13.8</tensao>
    </enrolamento>
  </transformador>

  <substacao nome="S1"/>
</elementos>
<!-->
*****
** Relacionamento entre os elementos
** que compoem a rede
*****
</!-->
<topologia>
  <elemento tipo="no" nome="S1">
    <vizinhos>
      <no nome="A2"/>
      <no nome="G1"/>
    </vizinhos>
    <chaves>
      <chave nome="1"/>
      <chave nome="4"/>
    </chaves>
  </elemento>

  <!-->Setores</!-->
  <elemento tipo="setor" nome="S1">
    <vizinhos>
      <setor nome="A"/>
      <setor nome="G"/>
    </vizinhos>
    <nos>
      <no nome="S1"/>
    </nos>
  </elemento>

  <!-->Chaves</!-->
  <elemento tipo="chave" nome="1">
    <n1>
      <setor nome="S1"/>
    </n1>
    <n2>
      <setor nome="A"/>
    </n2>
  </elemento>

  <!-->Trechos</!-->
  <elemento tipo="trecho" nome="S1CH1">
    <n1>
      <no nome="S1"/>
    </n1>
    <n2>
      <chave nome="1"/>
    </n2>
  </elemento>

```

```
<condutores>
  <condutor nome="CAA 266R"/>
</condutores>
</elemento>

<!-->Alimentadores</!-->
<elemento tipo="alimentador" nome="S1_AL1">
  <setores>
    <setor nome="S1"/>
    <setor nome="A"/>
    <setor nome="B"/>
    <setor nome="C"/>
    <setor nome="F"/>
  </setores>
  <trechos>
    <trecho nome="S1CH1"/>
    <trecho nome="CH1A2"/>
    <trecho nome="A2A1"/>
    <trecho nome="A2A3"/>
    <trecho nome="A2CH3"/>
    <trecho nome="CH3C1"/>
    <trecho nome="C1C3"/>
    <trecho nome="C3CH6"/>
    <trecho nome="C1C2"/>
    <trecho nome="C2CH5"/>
    <trecho nome="CH5F1"/>
    <trecho nome="F1F2"/>
    <trecho nome="F1F3"/>
    <trecho nome="A3CH2"/>
    <trecho nome="CH2B1"/>
    <trecho nome="B1CH10"/>
    <trecho nome="B1B2"/>
    <trecho nome="B2CH7"/>
  </trechos>
  <chaves>
    <chave nome="1"/>
    <chave nome="2"/>
    <chave nome="3"/>
    <chave nome="5"/>
    <chave nome="6"/>
    <chave nome="7"/>
    <chave nome="10"/>
  </chaves>
  <raiz>
    <setor nome="S1"/>
  </raiz>
</elemento>

<!-->Subestações</!-->
<elemento tipo="substacao" nome="S1">
  <alimentadores>
    <alimentador nome="S1_AL1"/>
    <alimentador nome="S1_AL2"/>
  </alimentadores>

  <transformadores>
    <transformador nome="S1_T1"/>
    <transformador nome="S1_T2"/>
  </transformadores>
</elemento>
```

```
</topologia>
<!-->
*****
** Dados de Comunicacao dos IED
*****
</!-->
<comunicacao>
  <elemento tipo="chave" nome="1">
    <endereco>
      <ip>192.168.24.101</ip>
      <porta>5001</porta>
    </endereco>
  </elemento>
</comunicacao>
</rede>
```

Fonte: Própria do autor.

---

## APÊNDICE B

# Algoritmo de fluxo de carga

---

Para que os cálculos de carregamento de condutores e transformadores das subestações, e das quedas de tensão nos nós de carga da rede elétrica sejam realizados, é utilizado neste trabalho o algoritmo de fluxo de carga para redes de distribuição de energia elétrica chamado de varredura direta-inversa por soma de potências.

Na Figura B.1 é mostrado um diagrama de sequência com os passos necessários à execução do algoritmo de fluxo de carga de varredura direta-inversa. Inicialmente uma tensão é atribuída à barra da subestação, esta tensão é a tensão inicial de cada uma das barras do sistema. Em seguida as potências equivalentes das barras são calculadas de acordo com a Equação B.1 e com a Equação B.2, no sentido das barras terminais para a barra fonte.

$$P_k^{eq} = P_k + \sum P_j + \sum PL_{kj} \quad (\text{B.1})$$

$$Q_k^{eq} = Q_k + \sum Q_j + \sum QL_{kj} \quad (\text{B.2})$$

Onde,

$P_k^{eq}$  é a potência ativa equivalente na barra  $k$ ;

$P_k$  é a carga ativa na barra  $k$ ;

$\sum P_j$  é o somatório das potências equivalentes das barras  $j$  diretamente conectadas após a barra  $k$ ;

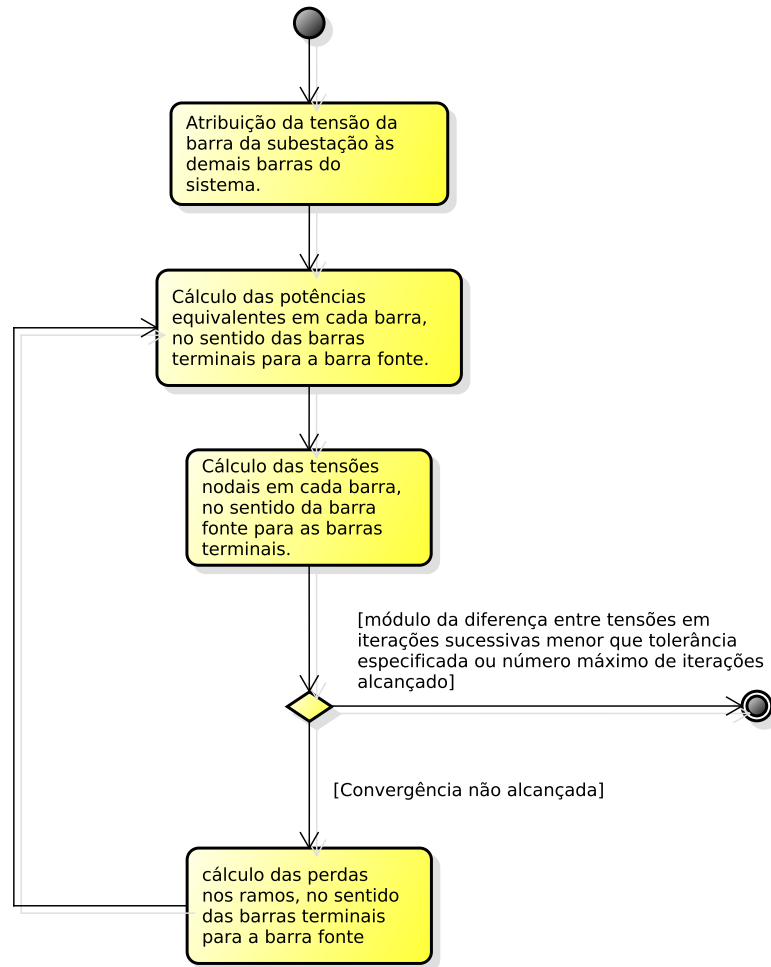
$\sum PL_{kj}$  é o somatório das perdas ativas nos ramos conectados entre a barra  $k$  e as barras  $j$  <sup>1</sup>.

Na equação de potência reativa o significado dos termos é equivalente.

---

<sup>1</sup> Na primeira iteração pode-se considerar que as perdas são nulas.

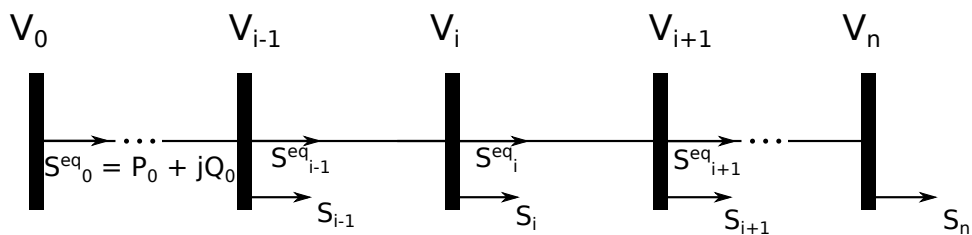
Figura B.1 – Diagrama de seqüência do cálculo do fluxo de carga de varredura direta-inversa.



Fonte: Própria do autor.

Após o cálculo das potências equivalentes procede-se o cálculo das tensões nodais em cada barra  $i$  do sistema, no sentido da barra fonte para as barras terminais. Na Figura B.2 a barra fonte é chamada de 0 e a barra terminal é chamada de  $n$ . Para cada barra do sistema está associada uma potência relacionada a carga conectada à barra e uma potência equivalente que flui de uma barra  $i - 1$  para uma barra  $i$  e desta para outra barra  $i + 1$  e assim por diante, até chegar à barra terminal  $n$ .

Figura B.2



Fonte: Adaptado de (SANTOS et al., 2008).

Para calcular as tensões nas barras de carga, utiliza-se as equações B.3 e B.4.

$$V_{i+1} = V_i - 2(r_i P_i^{eq} + x_i Q_i^{eq}) + (r_i^2 + x_i^2) \frac{(P_i^{eq})^2 + (Q_i^{eq})^2}{V_i^2} \quad (\text{B.3})$$

$$\delta_{i+1} = \delta_i - \tan^{-1}(k_1/k_2) \quad (\text{B.4})$$

$$k_1 = \frac{(P_i^{eq} x_i - Q_i^{eq} r_i)}{V_i}$$

$$k_2 = V_i - \frac{P_i^{eq} r_i - Q_i^{eq} x_i}{V_i}$$

Onde,

$V_i$  é a magnitude da tensão na barra  $i$ ;

$\delta_i$  é o ângulo da tensão na barra  $i$ ;

$r_i$  é a resistência em série do ramo  $i$ ;

$x_i$  é a reatância em série do ramo  $i$ .

---

# APÊNDICE C

## Algoritmo de descoberta de dados de um servidor IEC 61850

---

Algoritmo em Python para descoberta do modelo de dados de um servidor IEC 61850.

```
1 # coding=utf-8
2
3 import iec61850
4 from time import sleep
5
6 def get_data_directory(data_ref):
7     data_ref_ = iec61850.getDataDirectory(data_ref)
8     if data_ref_ != []:
9         for i in data_ref_:
10             get_data_directory(data_ref + '.' + i)
11     else:
12         sleep(0.1)
13         print data_ref
14         return
15
16 if __name__ == '__main__':
17
18     iec61850.createConnection('localhost', 102)
19
20     log_dev_list = iec61850.getLogicalDeviceList()
21     for i in log_dev_list:
22         print 'Dispositivo Logico: %s' % i
23         log_dev_direc = iec61850.getLogicalDeviceDirectory(i)
24         for j in log_dev_direc:
```

```
25         print 'No Logico: %s' % j
26         ln_ref = i + '/' + j
27         log_node_direc = iec61850.getLogicalNodeDirectory(ln_ref)
28         print 'Dados: '
29         for k in log_node_direc:
30             data_ref = ln_ref + '.' + k
31             get_data_directory(data_ref)
```



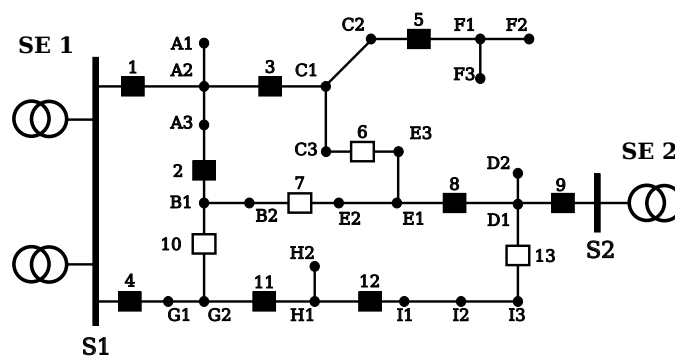
# APÊNDICE D

## Análises da rede teste

A rede elétrica utilizada para a realização de casos teste no trabalho é mostrada na Figura D.1. Neste apêndice são mostrados as telas de saída do algoritmo de cálculo de fluxo de carga, descrito no Apêndice B, utilizado pelo SMRA nas análises de recomposição para a rede elétrica da Figura D.1 em condições normais de operação.

Os resultados de fluxo de carga estão organizados por alimentador, sendo exibidas primeiro as análises de carregamento dos trechos da rede na seção D.1 por meio das Figuras D.2, D.3 e D.4. As análises de queda de tensão nos nós de carga estão mostradas na seção D.2 por meio das Figuras D.5, D.6 e D.7. A análise de carregamento dos transformadores das subestações está na seção D.3, mostrada nas Figuras D.8 e D.9.

Figura D.1 – Rede elétrica utilizada na realização de casos teste.



Fonte: Própria do autor.

### D.1 Análise de carregamento dos trechos

Figura D.2 – Análise de carregamento dos trechos do Alimentador S1\_AL2.

| trecho | condutor | comprimento (m) | ampacidade (A) | corrente (A) |
|--------|----------|-----------------|----------------|--------------|
| CH11H1 | CAA 266R | 1000.0          | 301.0          | 147.635      |
| I2I3   | CAA 266R | 1000.0          | 301.0          | 34.9945      |
| I3CH13 | CAA 266R | 1000.0          | 301.0          | 0.0          |
| H1CH12 | CAA 266R | 1000.0          | 301.0          | 100.3367     |
| H1H2   | CAA 266R | 1000.0          | 301.0          | 17.3389      |
| CH4G1  | CAA 266R | 10.0            | 301.0          | 203.2535     |
| I1I2   | CAA 266R | 1000.0          | 301.0          | 69.3872      |
| G1G2   | CAA 266R | 1000.0          | 301.0          | 177.7327     |
| CH12I1 | CAA 266R | 1000.0          | 301.0          | 100.3367     |
| G2CH11 | CAA 266R | 1000.0          | 301.0          | 147.635      |
| G2CH10 | CAA 266R | 1000.0          | 301.0          | 0.0          |
| S1CH4  | CAA 266R | 1000.0          | 301.0          | 203.2535     |

Fonte: Própria do autor.

Figura D.3 – Análise de carregamento dos trechos do alimentador S1\_AL1.

| trecho | condutor | comprimento (m) | ampacidade (A) | corrente (A) |
|--------|----------|-----------------|----------------|--------------|
| A2A3   | CAA 266R | 1000.0          | 301.0          | 70.0105      |
| A2A1   | CAA 266R | 1000.0          | 301.0          | 21.5959      |
| A3CH2  | CAA 266R | 1000.0          | 301.0          | 52.8095      |
| CH2B1  | CAA 266R | 1000.0          | 301.0          | 52.8095      |
| C2CH5  | CAA 266R | 1000.0          | 301.0          | 83.8719      |
| B1CH10 | CAA 266R | 1000.0          | 301.0          | 0.0          |
| CH3C1  | CAA 266R | 1000.0          | 301.0          | 180.3832     |
| B2CH7  | CAA 266R | 1000.0          | 301.0          | 0.0          |
| CH1A2  | CAA 266R | 1000.0          | 301.0          | 284.6946     |
| B1B2   | CAA 266R | 1000.0          | 301.0          | 27.4217      |
| A2CH3  | CAA 266R | 1000.0          | 301.0          | 180.3832     |
| C1C3   | CAA 266R | 1000.0          | 301.0          | 21.9146      |
| C1C2   | CAA 266R | 1000.0          | 301.0          | 124.5559     |
| CH5F1  | CAA 266R | 1000.0          | 301.0          | 83.8719      |
| F1F3   | CAA 266R | 1000.0          | 301.0          | 21.8533      |
| F1F2   | CAA 266R | 1000.0          | 301.0          | 26.4573      |
| C3CH6  | CAA 266R | 1000.0          | 301.0          | 0.0          |
| S1CH1  | CAA 266R | 10.0            | 301.0          | 284.6946     |

Fonte: Própria do autor.

Figura D.4 – Análise de carregamento dos trechos do alimentador S2\_AL1.

| trecho | condutor  | comprimento (m) | ampacidade (A) | corrente (A) |
|--------|-----------|-----------------|----------------|--------------|
| D1CH8  | Cobre 70R | 1000.0          | 274.0          | 56.323       |
| CH8E1  | Cobre 70R | 1000.0          | 274.0          | 56.323       |
| CH9D1  | Cobre 70R | 1000.0          | 274.0          | 94.2658      |
| E1E2   | Cobre 70R | 1000.0          | 274.0          | 15.0351      |
| E2CH7  | Cobre 70R | 1000.0          | 274.0          | 0.0          |
| D1D2   | Cobre 70R | 1000.0          | 274.0          | 21.1466      |
| D1CH13 | Cobre 70R | 1000.0          | 274.0          | 0.0          |
| E3CH6  | Cobre 70R | 1000.0          | 274.0          | 0.0          |
| S2CH9  | Cobre 70R | 10.0            | 274.0          | 94.2658      |
| E1E3   | Cobre 70R | 1000.0          | 274.0          | 17.4593      |

Fonte: Própria do autor.

## D.2 Análise de queda de tensão dos nós de carga

Figura D.5 – Análise de queda de tensão nos nós de carga do alimentador S1\_AL2

| no de carga | tensao (kV) | potencia aparente (kVA) | potencia ativa (kW) | potencia reativa (kVar) |
|-------------|-------------|-------------------------|---------------------|-------------------------|
| H1          | 13.4297     | 707.1068                | 700.0               | 100.0                   |
| G2          | 13.5913     | 707.1068                | 700.0               | 100.0                   |
| G1          | 13.6881     | 608.2763                | 600.0               | 100.0                   |
| I1          | 13.3172     | 718.0529                | 700.0               | 160.0                   |
| I3          | 13.2616     | 803.99                  | 800.0               | 80.0                    |
| I2          | 13.2783     | 807.7747                | 750.0               | 300.0                   |
| H2          | 13.4212     | 403.1129                | 400.0               | 50.0                    |
| S1          | 13.8        | 0.0                     | 0.0                 | 0.0                     |

Fonte: Própria do autor.

Figura D.6 – Análise de queda de tensão nos nós de carga do alimentador S1\_AL1

| no de carga | tensao (kV) | potencia aparente (kVA) | potencia ativa (kW) | potencia reativa (kVar) |
|-------------|-------------|-------------------------|---------------------|-------------------------|
| F1          | 13.2842     | 824.6211                | 800.0               | 200.0                   |
| F2          | 13.2706     | 608.2763                | 600.0               | 100.0                   |
| F3          | 13.2738     | 502.4938                | 500.0               | 50.0                    |
| S1          | 13.8        | 0.0                     | 0.0                 | 0.0                     |
| A1          | 13.6289     | 509.902                 | 500.0               | 100.0                   |
| A3          | 13.6014     | 406.0788                | 400.0               | 70.0                    |
| A2          | 13.6405     | 304.1381                | 300.0               | 50.0                    |
| B1          | 13.5415     | 605.3098                | 600.0               | 80.0                    |
| B2          | 13.5244     | 642.573                 | 600.0               | 230.0                   |
| C3          | 13.4307     | 509.902                 | 500.0               | 100.0                   |
| C2          | 13.373      | 948.6833                | 900.0               | 300.0                   |
| C1          | 13.4424     | 806.2258                | 800.0               | 100.0                   |

Fonte: Própria do autor.

Figura D.7 – Análise de queda de tensão nos nós de carga do alimentador S2\_AL1

| no de carga | tensao (kV) | potencia aparente (kVA) | potencia ativa (kW) | potencia reativa (kVar) |
|-------------|-------------|-------------------------|---------------------|-------------------------|
| E1          | 13.6444     | 583.0952                | 500.0               | 300.0                   |
| S2          | 13.8        | 0.0                     | 0.0                 | 0.0                     |
| E2          | 13.6344     | 355.1056                | 350.0               | 60.0                    |
| D2          | 13.7176     | 502.4938                | 500.0               | 50.0                    |
| E3          | 13.632      | 412.3106                | 400.0               | 100.0                   |
| D1          | 13.7307     | 412.3106                | 400.0               | 100.0                   |

Fonte: Própria do autor.

## D.3 Análise de carregamento dos transformadores das subestações

Figura D.8 – Análise de carregamento dos transformadores da subestação S1.

| trafo | potencia aparente (MVA) | carregamento (MVA) |
|-------|-------------------------|--------------------|
| S1_T2 | 6.0                     | 5.6894             |
| S1_T1 | 6.0                     | 5.6894             |

Fonte: Própria do autor.

Figura D.9 – Análise de carregamento dos transformadores da subestação S2.

| trafo | potencia aparente (MVA) | carregamento (MVA) |
|-------|-------------------------|--------------------|
| S2_T1 | 5.0                     | 2.2349             |

Fonte: Própria do autor.