

UNIVERSIDADE FEDERAL DE ITAJUBÁ

Pró-Diretoria de Pesquisa e Pós-Graduação

Programa de Pós-Graduação em Engenharia Elétrica

**UMA PROPOSTA DE ARQUITETURA
MULTI-AGENTE PARA OPERAÇÃO
DE SISTEMAS ELÉTRICOS**

ALEXANDRE RASI AOKI

ITAJUBÁ

2003

UNIVERSIDADE FEDERAL DE ITAJUBÁ

Pró-Diretoria de Pesquisa e Pós-Graduação

Programa de Pós-Graduação em Engenharia Elétrica

**UMA PROPOSTA DE ARQUITETURA
MULTI-AGENTE PARA OPERAÇÃO
DE SISTEMAS ELÉTRICOS**

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Itajubá para a obtenção do título de Doutor em Ciências em Engenharia Elétrica.

ALEXANDRE RASI AOKI

Orientador

PROF. DR. GERMANO LAMBERT-TORRES

Co-Orientador

PROF. DR. LUIZ EDUARDO BORGES DA SILVA

Itajubá, Fevereiro de 2003.

*A minha amada esposa
Milena, por todo seu amor
e dedicação sem os quais
não conseguiria alcançar
meus objetivos.*

AGRADECIMENTOS

Primeiramente, agradeço ao meu orientador, Prof. Germano Lambert Torres, por todas as oportunidades a mim oferecidas, mais do que me ajudar e ensinar contribuiu significativamente na formação do Engenheiro que sou hoje.

Ao Prof. Luiz Eduardo pelo incentivo que me ajudou a entender melhor o procedimento científico envolvido nesse trabalho.

Aos amigos, Carlos Henrique Valério de Moraes e Jonas Guedes Borges da Silva, pelo apoio na programação e pela amizade. Obrigado por compartilharem esse sonho comigo!

Aos meus familiares (Vó Isabel, pais, Durval, irmãos e irmãs, sobrinhos e sobrinhas, cunhados e cunhadas, sogro e sogra) que compreenderam todas as minhas ausências e sempre me apoiaram. Em especial para minha Mãe que me deu a base para que tudo isso se tornasse realidade.

Aqueles que não estão mais presentes (Vô Hideo, Vó Maria e Vô Aldo), mas que compartilharam desse sonho comigo.

Às minhas tias japonesas que são uma fonte de força de vontade, otimismo e um grande exemplo em minha vida.

A Deus que iluminou meu caminho e nunca me abandonou.

Por último, mas a mais importante, para minha esposa Milena, que sempre me deu força e dividiu todos os momentos, bons e ruins, desse sonho comigo. Por mim, por nós e pelos nossos futuros filhos!

LISTA DE TABELAS

Tabela 2.1 – Exemplos de tipos de agentes e suas descrições	17
Tabela 4.1 – Hierarquia de restabelecimento.....	73

LISTA DE FIGURAS

Figura 1.1 – Sistema baseado em conhecimento.....	8
Figura 2.1 – Definições de IA.....	12
Figura 2.2 – Conceito geral de agente inteligente.....	14
Figura 2.3 – Arquitetura de um AI utilizada.....	18
Figura 2.4 – Agentes reflexivos simples.....	21
Figura 2.5 – Agentes que acompanham o mundo.....	21
Figura 2.6 – Agentes baseados em metas.....	22
Figura 2.7 – Agentes baseados em satisfação.....	23
Figura 2.8 – Comunicação direta.....	26
Figura 2.9 – Sistema confederado.....	27
Figura 2.10 – Comunicação difundida.....	28
Figura 2.11 – Sistemas que usam quadro negro.....	29
Figura 3.1 – Classes (a) Genérica e (b) Exemplo “Window”.....	37
Figura 3.2 – Arquitetura multi-agente proposta.....	41
Figura 3.3 – Diagrama de Interação da arquitetura multi-agente proposta.....	42
Figura 3.4 – Exemplo de aplicação da arquitetura proposta.....	44
Figura 3.5 – Exemplo de modelo OO para o exemplo.....	45
Figura 3.6 – Classes do modelo OO.....	45
Figura 3.7 – Fluxograma de ações do Agente Distância.....	48
Figura 3.8 – Fluxograma de ações do “Agente Caminho”.....	49
Figura 3.9 – Diagrama de interações do SMA.....	51
Figura 4.1 – Diagrama de classes utilizado.....	54
Figura 4.2 – Classe “Equipamentos”.....	55
Figura 4.3 – Classes: (a) “Equipamentos Gerais”, (b) “Equipamentos de Proteção”, (c) “Equipamentos de Medição” e (d) “Equipamentos de Manobra”.....	56
Figura 4.4 – Classes: (a) “Geradores”, (b) “Transformadores”, (c) “Linhas”, (d) “Barras”, (e) “Alimentadores” e (f) “Capacitores”.....	58
Figura 4.5 – Classes: (a) “Disjuntores”, (b) “Chaves”, (c) “Chaves By-Pass” e (d) “Chaves de Aterramento”.....	59
Figura 4.6 – Sistema multi-agente para subestações elétricas.....	61
Figura 4.7 – Algoritmo de classificação por agente das sugestões.....	71
Figura 4.8 – Algoritmo de eliminação das sugestões iguais em listas diferentes....	72

Figura 4.9 – Algoritmo de eliminação das sugestões iguais e diferentes na lista auxiliar	74
Figura 4.10 – Diagrama de interação do SMA para subestações elétricas	75
Figura 4.11 – Modelo multi-agente para sistemas de distribuição.....	77
Figura 4.12 – Algoritmo de funcionamento do “Agente Restabelecimento”	79
Figura 4.13 – Ordenação da base de dados do sistema de distribuição.....	80
Figura 4.14 – Funções difusas para classificação	81
Figura 4.15 – Fluxograma de ações do algoritmo “Avançado”	82
Figura 4.16 – Algoritmo que cria listas de barras energizadas e desenergizadas ...	83
Figura 4.17 – Algoritmo que cria listas de barras para cada falha	84
Figura 4.18 – Algoritmo que busca soluções para as falhas	85
Figura 4.19 – Algoritmo geral de fluxo de carga.....	86
Figura 4.20 – Algoritmo para cálculo das correntes no sistema.....	88
Figura 4.21 – Algoritmo para cálculo das tensões nas barras	89
Figura 4.22 – Algoritmo para cortar cargas não prioritárias.....	90
Figura 4.23 – Diagrama de interação para o SMA para sistemas de distribuição	92
Figura 4.24 – Sistema integrado para restabelecimento de sistemas elétricos	94
Figura 5.1 – ETD Estuário – Bandeirante Energia S/A	97
Figura 5.2 – Falha em BA-1 - Passo 1	98
Figura 5.3 – Falha em BA-1 - Passo 2	99
Figura 5.4 – Falha em BA-1 - Passo 3	100
Figura 5.5 – Falha em BA-1 - Passo 4	101
Figura 5.6 – Falha em BA-1 - Passo 5	102
Figura 5.7 – Falha em BA-1 - Passo 6	103
Figura 5.8 – Colapso total - Passo 1	104
Figura 5.9 – Colapso total - Passo 2	105
Figura 5.10 – Colapso total - Passo 3	106
Figura 5.11 – Colapso total - Passo 4	107
Figura 5.12 – Colapso total - Passo 5	108
Figura 5.13 – Colapso total - Passo 6	109
Figura 5.14 – Colapso total - Passo 7	110
Figura 5.15 – Falha permanente em transformador - Passo 1	111
Figura 5.16 – Falha permanente em transformador - Passo 2	112
Figura 5.17 – Falha permanente em transformador - Passo 3	113
Figura 5.18 – Falha permanente em transformador - Passo 4	114

Figura 5.19 – Falha permanente em transformador - Passo 5	115
Figura 5.20 – Falha permanente em transformador - Passo 6	116
Figura 5.21 – Falha permanente em transformador - Passo 7	117
Figura 5.22 – Falha permanente em transformador - Passo 8	118
Figura 5.23 – Sistema de distribuição da CEB – Subestação de Taguatinga/DF ...	119
Figura 5.24 – Alimentador 0809	121
Figura 5.25 – Soluções para a falha total no alimentador.....	123
Figura 5.26 – Alimentador 0802	125
Figura 5.27 – Soluções para falha parcial no alimentador	126
Figura 5.28 – Fluxograma de integração do SISCO – Distribuição / SAO.....	128
Figura 5.29 – Fluxograma de Controle de Concorrência do SAO	129
Figura 5.30 – Fluxograma de Controle de Concorrência SMA	130
Figura 5.31 – Fluxograma de integração global dos sistemas	131
Figura B.1 – Divisão das especificações da FIPA	147
Figura C.1 – Janela principal do SISCO.....	156
Figura C.2 – Escolhendo o equipamento para visualizar as medições.....	158
Figura C.3 – Visualizando dados de medições.....	159
Figura C.4 – Visualizando dados de proteções	160
Figura C.5 – Janela de diálogo para alterar o estado das proteções.....	160
Figura C.6 – Janela de diálogo para alterar o estado dos disjuntores	161
Figura C.7 – Gerando o cenário pré-falta.....	162
Figura C.8 – Simulando uma falha	163
Figura C.9 – Fim da falha	164
Figura C.10 – Plano inicial de restabelecimento	165
Figura C.11 – Ajustando as medições durante o restabelecimento	166
Figura D.1 – Janela principal do programa “SISCO – Distribuição”	168
Figura D.2 – Área de cadastro de linhas	169
Figura D.3 – Janela para cadastro de linhas	170
Figura D.4 – Área de cadastro de barras	170
Figura D.5 – Janela para cadastro de barras.....	171
Figura D.6 – Fluxos calculados pelo “Flow”.....	172
Figura D.7 – Tensões e ângulos calculados pelo “Flow”.....	172
Figura D.8 – Sistema exemplo.....	173
Figura D.9 – Resposta da simulação	173
Figura D.10 – Alterando o estado da linha para “Falha”	174

Figura D.11 – Simulando em “Modo Básico”	175
Figura D.12 – Resposta da simulação em “Modo Básico”	175
Figura D.13 – Simulando em “Modo Avançado”	176
Figura D.14 – Resposta da simulação em “Modo Avançado”	177
Figura D.15 – Ajustando os dados para duas falhas simultâneas	178
Figura D.16 – Resposta obtida para a reconfiguração do sistema	178
Figura D.17 – Ajustando os dados para duas falhas simultâneas sobrepostas	179
Figura D.18 – Resposta obtida para a reconfiguração do sistema	179
Figura E.1 – Fluxograma global de operação.....	182
Figura E.2 – Tabela de bairros do banco de dados da RA I – Brasília.....	184
Figura E.3 – Tabela de pontos do banco de dados da RA I – Brasília	184
Figura E.4 – Tabela de ruas do banco de dados da RA I – Brasília	185
Figura E.5 – Tabela de quadras do banco de dados da RA I – Brasília	185
Figura E.6 – Estrutura de relacionamento das tabelas	186
Figura E.7 – Janela principal do programa “Percurso”.....	187
Figura E.8 – Área de cadastro de quadras	188
Figura E.9 – Janela para cadastro de quadras.....	188
Figura E.10 – Área de cadastro de chaves	190
Figura E.11 – Janela para cadastro de chaves	190
Figura E.12 – Dados para busca.....	191
Figura E.13 – Resposta da busca de percurso.....	192
Figura E.14 – Buscando o caminho entre bairros para a solução	193
Figura E.15 – Determinando os melhores caminhos entre bairros	194
Figura E.16 – Melhor caminho para a primeira etapa da solução.....	195
Figura E.17 – Melhor caminho para a segunda etapa da solução	195
Figura E.18 – Melhor caminho para a terceira etapa da solução	196
Figura E.19 – Melhor caminho para segunda solução do passo 2.....	196

LISTA DE ABREVIATURAS E SIGLAS

ACL	<i>Agent Communication Language</i>
AI	Agente Inteligente
AUML	<i>Agent Unified Modeling Language</i>
BC	Base de Conhecimento
COS	Centro de Operação do Sistema
DEC	Desligamento Equivalente de Interrupção pó por Consumo
EMS	<i>Energy Management Systems</i>
ES	<i>Expert System</i> – Sistema Especialista
FEC	Freqüência Equivalente de Interrupção pó por Consumo
FIPA	<i>Foundation for Intelligent Physical Agents</i>
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
KQML	<i>Knowledge Query and Manipulation Language</i>
LT	Linha de Transmissão
POO	Programação Orientada a Objetos
PSO	<i>Particle Swarm Optimization</i>
SCADA	<i>Supervisory Control and Data Acquisition System</i>
SE	Subestação Elétrica
SEP	Sistema Elétrico de Potência
SISCO	Sistema Inteligente de Supervisão e Controle
SMA	Sistema Multi-Agente
UML	<i>Unified Modeling Language</i>

SUMÁRIO

AGRADECIMENTOS	iii
LISTA DE TABELAS	iv
LISTA DE FIGURAS	v
LISTA DE ABREVIATURAS E SIGLAS	ix
RESUMO	xiv
<i>ABSTRACT</i>	xv
1 INTRODUÇÃO	1
1.1 Inteligência Artificial versus Vida Artificial.....	1
1.2 Descrição do Problema a ser Resolvido.....	4
1.3 Objetivos.....	7
1.4 Histórico da Aplicação de Sistemas Inteligentes em Sistemas Elétricos de Potência	8
1.5 Organização desse Trabalho.....	10
2 AGENTES INTELIGENTES	12
2.1 Introdução	12
2.2 Propriedades dos Agentes	14
2.2.1 Autonomia	15
2.2.2 Habilidade Social	15
2.2.3 Reatividade	15
2.2.4 Iniciativa Própria.....	16
2.3 Arquitetura de um Agente Inteligente	16
2.3.1 Sensores.....	18
2.3.2 Atuadores	18
2.3.3 Programa Inteligente	19
2.3.4 Base de Conhecimento	19
2.3.5 Interface de Comunicação	19
2.4 Tipos de Agentes	20
2.4.1 Agentes Reflexivos Simples.....	20
2.4.2 Agentes que Acompanham o Mundo	20
2.4.3 Agentes Baseados em Metas	21
2.4.4 Agentes Baseados em Satisfação	22
2.5 Sistemas Multi-Agente Inteligente	23

2.5.1 Comunicação entre Agentes	26
2.5.2 Interação dos Agentes	29
2.5.3 Coerência e Coordenação dos Agentes	31
2.5.4 Vantagens de Sistemas Multi-Agentes.....	32
2.5.5 Mais Informações sobre Agentes Inteligentes	33
3 A ARQUITETURA MULTI-AGENTE PROPOSTA.....	35
3.1 Programação Orientada a Objetos	36
3.2 A Arquitetura Multi-Agente Proposta	38
3.3 Exemplo de Implementação da Arquitetura Multi-Agente Proposta	43
3.3.1 Modelo Orientado a Objetos	43
3.3.2 Agentes Especializados	47
3.3.3 Diagrama de Interações	50
4 IMPLEMENTAÇÃO DA ARQUITETURA MULTI-AGENTE PROPOSTA EM SISTEMAS ELÉTRICOS	53
4.1 Orientação a Objetos aplicada em Sistemas Elétricos	53
4.1.1 Embasamento Teórico.....	53
4.1.2 A Estrutura de Programação Orientada a Objetos Proposta	53
4.2 Implementação da Arquitetura Proposta para Restabelecimento de Subestações Elétricas	60
4.2.1 Agentes Especializados	62
4.2.1.1 Agente Medidas	62
4.2.1.2 Agente Equipamentos	65
4.2.1.3 Agente Chaves	67
4.2.1.4 Agente Proteções.....	68
4.2.1.5 Agente Planejador.....	70
4.2.2 Diagrama de Interação entre Agentes	75
4.3 Implementação da Arquitetura Proposta para Restabelecimento de Sistemas de Distribuição	76
4.3.1 Agentes Especializados	78
4.3.1.1 Agente Restabelecimento.....	78
4.3.1.2 Agente Manobras	81
4.3.1.3 Agente Fluxo de Carga	86
4.3.1.4 Agente Corte Seletivo de Carga	89
4.3.2 Diagrama de Interação entre Agentes	91

4.4 Uma Proposta de Integração dos Aplicativos para Sistemas de Distribuição e para Subestações Elétricas.....	93
5 TESTES, RESULTADOS E IMPLEMENTAÇÃO	96
5.1 Falha na Barra da Subestação	96
5.2 Colapso Total na Subestação.....	103
5.3 Falha Permanente em Transformador.....	110
5.4 Falha Total em Alimentador de Distribuição	118
5.5 Falha Parcial em Alimentador de Distribuição	124
5.6 Implementação Real	127
6 CONCLUSÕES	133
6.1 Resultados	133
6.2 Desenvolvimentos Futuros.....	138
REFERÊNCIAS	139
ANEXO A PUBLICAÇÕES REALIZADAS	145
A.1 Artigos em Congressos	145
A.2 Capítulos de Livro	146
ANEXO B ESPECIFICAÇÕES DA FIPA.....	147
B.1 Aplicações	148
B.2 Arquitetura Abstrata	148
B.3 Comunicação de Agentes	149
B.3.1 Protocolos de Interação.....	149
B.3.2 Atos Comunicativos.....	151
B.3.3 Conteúdo de Linguagem.....	151
B.4 Gerenciamento de Agentes.....	152
B.5 Transporte de Mensagens de Agentes	153
B.5.1 Representações ACL	153
B.5.2 Representações Envelope	153
B.5.3 Protocolos de Transporte.....	154
ANEXO C IMPLEMENTAÇÃO COMPUTACIONAL DO SISTEMA MULTI-AGENTE PARA O RESTABELECIMENTO DE SUBESTAÇÕES ELÉTRICAS.....	155
C.1 Apresentando um Caso.....	155
C.2 Visualizando e Alterando Estados de Equipamentos	157
C.3 Simulando um Caso.....	161
C.3.1 Gerando o Cenário Pré-falta.....	161
C.3.2 Simulando.....	162

C.3.3 Restabelecendo a Subestação	165
ANEXO D IMPLEMENTAÇÃO COMPUTACIONAL DO SISTEMA MULTI-AGENTE PARA O RESTABELECIMENTO DE SISTEMAS DE DISTRIBUIÇÃO	167
D.1 Cadastrando um Sistema	168
D.2 Simulando um Sistema	171
D.2.1 Simulando um Sistema no Modo Básico	173
D.2.2 Simulando um Sistema no Modo Avançado	175
D.2.3 Simulando Falhas Simultâneas	177
ANEXO E SISTEMA INTELIGENTE DE LOCOMOÇÃO DE VIATURAS PARA ATENDIMENTO NA REDE DE DISTRIBUIÇÃO	180
E.1 Fluxograma Global de Operação	181
E.2 Modelo de Dados	183
E.2.1 Estrutura do Banco de Dados do Distrito Federal	183
E.2.2 Estrutura do Banco de Dados de Chaves	186
E.3 Procedimentos Operacionais	186
E.3.1 Cadastrando uma Cidade	187
E.3.2 Buscando um Percurso	191
E.4 Metodologia de Busca	192

RESUMO

Há uma demanda permanente por novas ferramentas computacionais necessárias para operação, supervisão, análise e planejamento de sistemas elétricos. Devido às características da aplicação, estes *softwares* se tornam maiores e mais complexos, e quando terminados são difíceis de manter e integrar com aplicativos antigos.

O objetivo principal desta tese é propor um novo tipo de arquitetura de sistema multi-agente capaz de auxiliar os operadores de sistemas elétricos no seu trabalho de operação, com enfoque especial para as tarefas de restabelecimento após uma falha.

A arquitetura do sistema multi-agente foi dividida em dois pacotes de agentes. O primeiro é composto por “Agentes de Suporte”, os quais são responsáveis por prestar serviços que auxiliem o processo de tomada de decisões, como: estabelecer e coordenar a comunicação entre os agentes, atualizar o modelo computacional, interface com o mundo real, etc. O segundo pacote é composto por “Agentes Especializados”, que são os detentores do conhecimento específico empregado na análise e solução dos problemas encontrados no domínio de aplicação.

Os principais benefícios alcançados com o modelo multi-agente foram: comportamento cooperativo, processamento concorrente, topologia distribuída e arquitetura aberta.

O comportamento cooperativo é caracterizado por trocas dinâmicas de informação entre os agentes e possibilitando a divisão de tarefas, reduzindo as características hierárquicas do processo de decisão. Já o processamento concorrente melhora o uso da capacidade de *hardware* e diminui o tempo de resposta.

A topologia da arquitetura proposta é distribuída, o que torna possível no processo de abstração a divisão do sistema em um número menor de partes, limitado pelas características funcionais. E ainda, a arquitetura proposta é aberta, permitindo a adição de componentes e a expansão do sistema, como parte de um sistema maior com a criação de novos níveis de análise.

Como um exemplo de aplicação da arquitetura proposta foi implementado na prática em subestações elétricas e em sistemas de distribuição de concessionárias de energia elétrica do Brasil.

ABSTRACT

There is a permanent demand for new computational tools required for power system operation, supervision, analysis and planning. These software's become larger and increasingly complex due to the system's characteristics, and when finally finished they are difficult to maintain and to integrate into old application.

The main objective of this thesis is to consider a new type of multi-agent system's architecture, capable of to assist the power systems operators in their work of operation, with special approach for the restoration tasks after a fault.

The multi-agent system's architecture was divided in two packages of agents. The first one is composed by "Support Agents", which are responsible for doing services that assist the decision-making process, as: to establish and to coordinate the communication between agents, to update the computational model, interface with the real world, etc. The second package is composed by "Specialized Agents", that have the specific knowledge used in the analysis and solution of the problems found in the application domain.

The main benefits obtained with the multi-agent model were: cooperative behavior, competitive processing, distributed topology and open architecture.

The cooperative behavior is characterized by dynamic exchange of information between agents making possible segmentation of tasks, besides allowing the reduction of the hierarchic characteristics of the decision process. The competitive processing improves the use of the hardware capacity and decreases the reply time.

The architecture's topology considered is distributed, what makes possible in the abstraction process the system's division in a lesser number of parts, limited to functional characteristics. The architecture considered is opened, allowing the addition of components and the system's expansion, as part of a bigger system with the creation of new levels of analysis.

As an example of application of the architecture considered it was implemented in real cases like power substations and distribution systems of Brazil's power utility companies.

1 INTRODUÇÃO

1.1 Inteligência Artificial versus Vida Artificial

A área da Inteligência Artificial (IA) busca entender entidades inteligentes, porém ao contrário da Filosofia e da Psicologia, as quais também estão focadas na inteligência, a IA busca reproduzir essas entidades computacionalmente.

A IA foi formalmente iniciada em 1956, quando John McCarthy convenceu Marvin Minsky, Claude Shannon e Nathaniel Rochester a ajudá-lo a organizar um workshop de dois meses com os pesquisadores americanos interessados em automata, redes neurais e no estudo da inteligência. Esse *workshop*, realizado no *Dartmouth College*, resultou na adoção do nome “Inteligência Artificial” para essa área de pesquisa e, sobretudo, serviu para os pesquisadores se conhecerem. Pelos vinte anos seguintes esses pesquisadores dominaram a área [RUSSEL & NORVIG, 1995].

Embora a IA tenha sido formalmente iniciada naquela época já se trabalhava nessa área anteriormente, um bom exemplo disso é o Teste de Turing, proposto por Alan Turing (1950), o qual foi desenvolvido para gerar uma definição operacional satisfatória de inteligência.

Turing definiu o comportamento inteligente como a habilidade de alcançar o desempenho humano em todas as tarefas cognitivas, de forma suficiente para enganar um interrogador. Em grosso modo, o teste que ele propôs era para que um computador fosse interrogado por um humano através de um teletipo, e passaria no teste se o interrogador não pudesse afirmar se havia um computador ou um humano do outro lado.

Os primeiros anos da IA foram cheios de sucesso – de modo limitado, pois os computadores ainda eram muito primitivos. Era comum os pesquisadores da

área fazerem previsões não muito tímidas sobre seu futuro sucesso. Em seguida, a IA enfrentou uma época denominada de “Período Negro”, quando muitos de seus fundos de suporte à pesquisa foram cortados.

Nessa época, a barreira que quase todos os pesquisadores de IA enfrentaram foi que os métodos que eram suficientes para demonstrações em um ou dois exemplos simples falhavam quando empregados em problemas maiores ou reais. Isso ocorreu devido à:

- Falta de conhecimento sobre o domínio de aplicação contido nos programas, por exemplo, programas de tradução de artigos científicos russos para o inglês eram baseados em simples transformações sintáticas e na substituição das palavras através de um dicionário eletrônico;
- Escolha de problemas intratáveis para serem solucionados, por exemplo, acreditava-se que para resolver problemas maiores era só dispor de máquinas mais rápidas e com mais memória;
- Algumas limitações fundamentais nas estruturas básicas usadas para gerar o comportamento inteligente, por exemplo, os neurônios perceptrons podiam aprender qualquer coisa que eles pudessem representar, porém eles podiam representar muito pouco.

Esse período terminou no início da década de 70, quando Edward Feigenbaum e outros pesquisadores em Stanford deram início ao Projeto de Programação Heurística (HPP), o qual iria investigar a extensão de uma nova metodologia, baseada na clara separação do conhecimento (em forma de regras) da componente de raciocínio, para outras áreas do conhecimento humano. E assim iniciou-se o desenvolvimento de sistemas especialistas (*Expert Systems* – ES).

A grande demanda por aplicações de ES em problemas do mundo real causou um aumento concomitante por esquemas de representação de conhecimento, o qual resultou em novas versões da linguagem LISP (*LISt Processor*, criada em 1958, por John McCarthy) e na criação da linguagem PROLOG (*PROgramming in LOGic*, criada em 1973, por Alain Colmaurer).

Na década de 80, a IA se tornou uma área de grande interesse devido ao expressivo número de sistemas especialistas operando com sucesso em empresas. Além disso, as redes neurais voltaram à voga depois de reinventarem o algoritmo de aprendizagem por retro-propagação do erro para mais de duas camadas. Surgiu então uma grande rivalidade entre a IA simbólica (métodos tradicionais) e a IA conexionista (redes neurais).

Da década de 90 para cá, a IA cresceu muito devido ao grande interesse da comunidade científica e da rápida evolução dos sistemas computacionais nessa época. Muitas técnicas surgiram, diversas aplicações em novas áreas foram desenvolvidas e a IA se tornou uma das linhas de pesquisa mais atrativas citada pelos cientistas de outras disciplinas.

A principal deficiência da IA tradicional tem sido assumir que a cognição existe dentro da mente de uma pessoa. Um programa de IA é modelado segundo um paradigma de uma única e isolada pessoa, processando informação dentro do seu cérebro. Já a Vida Artificial, que descende da Cibernética e da Biologia Teórica, estabelece que separar mente e corpo é absurdo, e que para se estudar inteligência é necessário estudar organismos materiais que evoluem e se adaptam a um mundo também material.

A Vida Artificial está mais interessada em sistemas que se auto-organizam para implementar determinado comportamento inteligente, do que especificar *a priori* esse comportamento, como faz a IA tradicional. Esta idéia deu origem a duas linhas de investigação: a primeira, dedica-se a construir robôs cujo comportamento emerge da interação com o ambiente, em vez de ser pré-programado; a segunda, dedica-se a criar modelos de inteligência distribuída, em que populações de agentes simples solucionam problemas reais apenas com informação local.

A Vida Artificial tem dois conceitos principais: primeiro, a inteligência deve estar ligada a um determinado corpo com necessidades específicas num determinado ambiente; segundo, a inteligência é o resultado emergente da interação de populações de componentes que solucionam problemas muito simples.

Há duas vertentes para se implementar essa interação: a primeira, normalmente empregada em Sistemas Multi-Agentes (SMA), onde sub-rotinas autônomas desempenham funções especializadas e trocam informações entre si, sem que haja necessariamente alteração no seu processo de tomada de decisão; e a segunda, chamada de interação social, onde além da troca de informações há o compartilhamento de crenças, regras e dicas, o que tipicamente resulta numa mudança no processo de decisão (KENNEDY & EBERHART, 2001).

Nos Sistemas Multi-Agentes Inteligentes a premissa básica é prover mecanismos para a criação de sistemas inteligentes a partir de entidades de software autônomas, chamadas de agentes, que interagem através de um ambiente compartilhado por todos os membros de uma sociedade, e sobre o qual esses agentes atuam alterando o seu estado.

Um SMA é capaz de prover mecanismos para interação e coordenação dessas entidades, uma vez que cada Agente Inteligente (AI) possui uma certa habilidade e seus próprios objetivos. E ainda, os AIs precisam interagir para resolverem problemas grandes, portanto deve existir um conhecimento social embutido em cada um para permitir que ele conheça os outros AIs inseridos no mesmo ambiente (SYCARA, 1998). Este trabalho utiliza este encaminhamento.

Dentro da segunda linha de investigação destaca-se também a técnica de otimização por enxame de partículas (PSO – *Particle Swarm Optimization*), a qual utiliza uma população de soluções candidatas que evolui para uma solução ótima medida por uma função de avaliação definida pelo usuário. As partículas percorrem o espaço de busca de tal forma que a velocidade de cada partícula é acelerada de forma estocástica a partir de sua melhor posição e a partir da sua melhor posição vizinha (ESMIN, AOKI & LAMBERT-TORRES, 2002).

1.2 Descrição do Problema a ser Resolvido

Quando ocorre um blecaute no sistema elétrico de potência os efeitos na indústria, comércio e até na vida cotidiana da população em geral podem ser muito prejudiciais. Logo após um blecaute, é muito importante que o serviço de

fornecimento de energia elétrica seja restabelecido rapidamente. A tarefa de restabelecimento é bastante complexa e, freqüentemente, tem de ser modificada ou reformulada de acordo com os desdobramentos dos eventos do sistema [ADIBI, 2000].

Em adição, a dificuldade inerente da tarefa de restabelecimento, pode-se também relacionar o estresse causado no corpo técnico da companhia durante uma falta de energia no sistema, pois os operadores devem tomar decisões rápidas, por vezes sem ter uma completa idéia do problema do sistema. Isso tudo leva a soluções não otimizadas ou mesmo a soluções que podem comprometer o próprio sistema (ou algum de seus equipamentos). Dessa forma, o desenvolvimento de ferramentas que auxiliem o operador durante a falta de energia é muito importante, pois fornece os elementos necessários para uma tomada de decisão segura sobre “o que fazer” e “como fazer”.

Nos últimos anos, houve um interesse crescente em práticas de restabelecimento de sistemas elétricos de potência pelas indústrias e concessionárias. Uma interessada em manter o seu processo produtivo sem paradas, e a outra em melhorar seus índices de desempenho e qualidade no fornecimento.

E ainda, cada vez mais os consumidores estão exigentes quanto à continuidade do fornecimento de energia elétrica. Índices de continuidade e qualidade de energia, como DEC e FEC, têm sido cada vez mais fiscalizados e cobrados pelas agências reguladoras. Inclusive, reduções dos valores desses índices estão programadas para ocorrerem ao longo dos anos. É sabido, que ultrapassagens desses valores podem levar as concessionárias ao pagamento de multas.

Por outro lado, a operação de sistemas elétricos de potência envolve tarefas nos Centros de Operação do Sistema, tais como: monitoramento contínuo de grandezas do sistema, manobras, análises, o restabelecimento do sistema após uma falha, entre outras. O operador também dispõe de algumas ferramentas para auxiliá-lo, porém há dificuldades na extração/interpretação dos dados. Aliado a este problema, muitas vezes para uma análise completa é necessário o uso de diversos aplicativos, sendo que estes, via de regra, não estão preparados para trocar dados

ou cooperarem entre si. Isso inviabiliza a análise devido ao tempo necessário para executar todos os programas ou pela complexidade desta ação.

O grau de risco envolvido na operação dos sistemas de distribuição de energia elétrica tem aumentado consideravelmente, pois além do crescimento do número de ramais, os investimentos neste sistema não têm acompanhado o crescimento da carga atendida, fazendo com que os equipamentos trabalhem bem mais próximos de sua capacidade nominal, ou até mesmo em sobrecarga.

Analisando, agora, o problema específico de subestações, quando ocorre uma falha que produz um desligamento de carga ou outra situação de emergência, ela deve ser localizada o mais rápido possível. Esta tarefa pode ser muito demorada devido à natureza complexa de construção e operação das subestações. E ainda, a quantidade de informação analisada é muito grande e até mesmo inspeções visuais podem ser necessárias [ALVES DA SILVA *et al.*, 1996].

Existem na literatura diversos trabalhos de aplicação de Inteligência Artificial (IA) em restabelecimento e operação de SEPs, alguns com resultados expressivos, porém as implementações desses programas em centros de controle ainda são poucas, quando comparadas com o número de aplicações desenvolvidas. Esta discrepância revela a necessidade de se desenvolver interfaces de integração dos sistemas SCADA (*Supervisory Control and Data Acquisition System*) e EMS (*Energy Management System*) com os sistemas inteligentes [BANN *et al.*, 1996].

Há três características principais nas aplicações de IA em SEPs: em primeiro lugar, são trabalhos focados em um tipo de problema do SEP; em segundo lugar, cada qual utiliza um modelo de dados e conhecimento diferente; e em terceiro lugar, utilizam técnicas, inteligentes ou não, muito boas, porém que não possuem uma estrutura de suporte simples para integração com outras técnicas. Essas três características tornam o trabalho de integração desses aplicativos muito difícil.

1.3 Objetivos

O objetivo principal desta tese é desenvolver uma arquitetura de sistema multi-agente inteligente capaz de auxiliar os operadores de sistemas elétricos no seu trabalho de operação, com enfoque especial para as tarefas de restabelecimento após uma falha.

Essa estrutura deve suportar a implementação de uma aplicação para subestações elétricas e outra para sistemas de distribuição que possam funcionar em conjunto colaborando entre si, e ainda, uma futura expansão para sistemas elétricos de potência.

Segundo LAMBERT-TORRES *et al.* (1997) um sistema de apoio à decisão capaz de colaborar com o operador durante a tarefa de operação de sistemas elétricos deve ser capaz de:

- Avaliar o agente da ocorrência;
- Definir a área defeituosa que foi isolada pela atuação da proteção;
- Caracterizar se o defeito é permanente ou transitório;
- Identificar os componentes envolvidos e os afetados;
- Gerar um plano de restabelecimento.

Para tal, essa arquitetura é composta de agentes especializados para cada implementação, os quais trabalham como especialistas em determinados tipos de análises simples/locais, e ainda um conjunto de agentes de suporte os quais colaboram com a interação entre os componentes do SMA.

O comportamento inteligente emerge da interação entre os agentes, a qual possibilita que uma população de entidades autônomas simples resolva problemas complexos, maiores que seus domínios de conhecimento individuais e independente da topologia do sistema elétrico utilizado.

1.4 Histórico da Aplicação de Sistemas Inteligentes em Sistemas Elétricos de Potência

A aplicação de técnicas de Inteligência Artificial em Sistemas Elétricos de Potência iniciou-se com a publicação do trabalho de SAKAGUCHI & MATSUMOTO (1983), ambos da *Mitsubishi Electric Corporation*. Nesse trabalho os autores utilizam um sistema baseado em conhecimento (Figura 1.1), desenvolvido em Interlisp que possuía dezesseis regras, capaz de colaborar com o operador no processo de tomada de decisão numa situação de restabelecimento. Esse sistema é capaz de raciocinar sobre as possíveis alterações topológicas para restabelecer o sistema e as suas conseqüências, em termos de sobrecargas nas linhas.

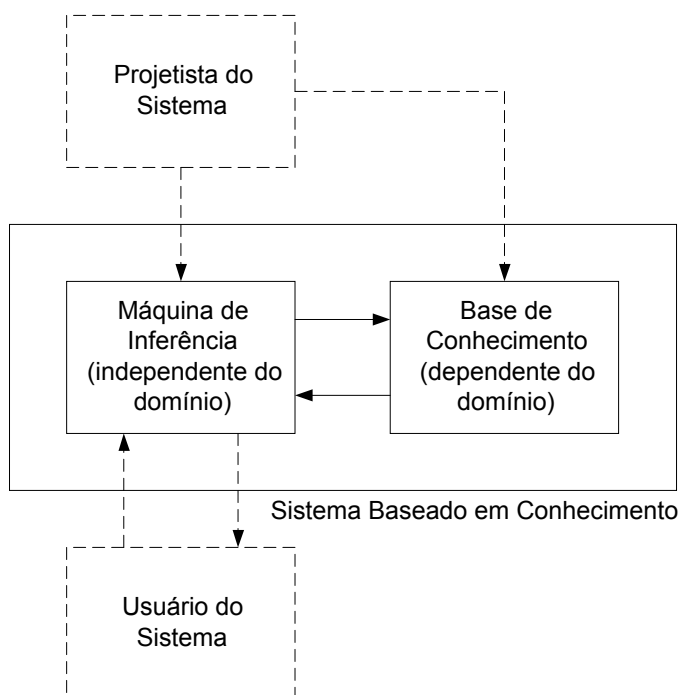


Figura 1.1 – Sistema baseado em conhecimento

As principais vantagens apontadas para esse desenvolvimento foram: a capacidade do sistema baseado em conhecimento de trabalhar com heurísticas, a modularidade do conhecimento e a transparência para o usuário do encadeamento do raciocínio empregado pelo programa. Como principais desvantagens observou-se a lentidão do processo de computação simbólica e a falta de técnicas para checar a consistência da base de conhecimento como um todo.

A aplicação de sistemas inteligentes em sistemas elétricos de potência ganhou bastante prestígio dentro e fora da comunidade científica, desde 1994 há uma conferência internacional em aplicação de sistemas inteligentes em sistemas de potência (*International Conference on Intelligent System Application to Power Systems – ISAP*). O ISAP foi originado da junção de duas conferências denominadas ESAP e ANNPS.

- O ESAP (*The International Conference on Expert System Application to Power Systems*) se dedicava a discussão de tópicos ligados à aplicação da Inteligência Artificial simbólica (sistemas especialistas, sistemas baseados em conhecimento, algoritmos genéticos, etc.) à solução dos problemas dos sistemas elétricos de potência. O ESAP teve 4 edições: Helsinki (1987), Seattle (1989), Yokohama (1991) e Melbourne (1993).
- O ANNPS (*The International Forum on Application of Neural Networks to Power Systems*) discutia assuntos da Inteligência Artificial conexionista (basicamente, redes neurais) que pudessem contribuir para solucionar problemas nos sistemas elétricos de potência. O ANNPS teve 2 edições: Seattle (1991) e Kobe (1993).

Como os assuntos das duas conferências estavam cada vez mais sendo aplicados em conjunto, formando sistemas inteligentes híbridos, decidiu-se montar uma conferência única, onde todos estes assuntos pudessem ser debatidos, daí surgiu o ISAP. Sua primeira edição foi realizada em *Montpellier*, em meados de 1994, sob os auspícios da *Électricité de France* (EDF). Em seguida, aconteceu a segunda edição, em *Orlando*, em janeiro de 1996, que teve como organizadora a *University of Florida*. Em julho de 1997, em *Seul*, foi organizada a terceira edição do ISAP pela *Seoul National University*, com o auxílio da *Korea Electric Power Corporation*.

Em abril de 1999, no Rio de Janeiro, aconteceu a quarta edição do ISAP organizada pela Escola Federal de Engenharia de Itajubá, onde o *Chair* da conferência foi o Prof. Germano Lambert Torres. Em junho de 2001, em Budapest, foi organizada a quinta edição da conferência pela *Technical University of Budapest*,

com o auxílio da *Hungarian Elettrotechnical Association*. Nessa edição da conferência houve pela primeira vez uma sessão para trabalhos utilizando agentes inteligentes, onde foram apresentados os trabalhos de SANZ-BOBI *et al.* (2001), MANGINA, MCARTHUR & MCDONALD (2001) e CHANG *et al.* (2001).

A sexta edição está programada para acontecer em junho de 2003 em *Lemnos* e está sendo organizada pela *National Technical University of Athens*, com o auxílio da *Hellenic Artificial Intelligence Society*.

Os desenvolvimentos na área de Agentes Inteligentes não remontam há muito tempo atrás, com um ápice na década de 1990 [WOOLDRIDGE & JENNINGS, 1995]. Nesse período diversos trabalhos buscaram formalizar e consolidar essa área dentro da chamada Inteligência Artificial Distribuída, como por exemplo o trabalho de PARUNAK (1996) e o de JENNINGS (1996).

A aplicação de sistemas multi-agentes em sistemas elétricos de potência só se consolidou a partir de 2000, com a publicação de trabalhos como os de AMIN (2001) e AZEVEDO, FEIJÓ & COSTA (2000). Vale ressaltar também que o problema de integrar aplicativos inteligentes para sistemas elétricos de potência foi apresentado por BANN *et al.* (1996). Nesse trabalho os autores relataram que a Inteligência Artificial colaborava em muito para resolver diversos problemas de sistemas elétricos, porém havia uma dificuldade na integração dos aplicativos inteligentes existentes devido à utilização de diferentes modelos de dados e processos de inferência e pela falta de meios para elaborar um processo de comunicação externo.

1.5 Organização desse Trabalho

Essa tese está organizada em capítulos conforme apresentado a seguir:

O capítulo 2 apresenta a teoria de agentes inteligentes, bem como de sistemas multi-agentes. Ressaltam-se neste capítulo diversas características

desejáveis que devem ser perseguidas durante a implementação de tais sistemas, bem como seus benefícios.

No terceiro capítulo introduzem-se alguns aspectos teóricos importantes e diretamente relacionados com o desenvolvimento dos aplicativos, são eles: a estrutura de dados e a estrutura dos sistemas multi-agentes.

No capítulo 4 detalha-se a implementação do aplicativo de apoio à decisão para operação de subestações elétricas e para sistemas de distribuição, apresentando os agentes desenvolvidos, a filosofia de operação/restabelecimento utilizada, bem como o processo de integração.

No capítulo 5 apresentam-se alguns exemplos e testes realizados com os aplicativos. Nesses exemplos são explorados aspectos de uso isolado de cada aplicativo, bem como para integração do mesmo.

No último capítulo traçam-se algumas conclusões, e ainda, são apontados alguns desenvolvimentos futuros.

2 AGENTES INTELIGENTES

2.1 Introdução

A Inteligência Artificial (IA) se concentra no entendimento das entidades inteligentes, logo estudar IA é aprender mais sobre nós mesmos. Ao contrário da filosofia e da psicologia, a IA tenta reproduzir essas entidades inteligentes computacionalmente, além de entendê-las.

O estudo da inteligência é muito antigo, há mais de 2000 anos tenta-se entender processos como enxergar, aprender, lembrar, etc. Com o advento dos computadores, no começo da década de 50, pôde-se experimentar essas teorias sobre inteligência. Nessa época a IA se mostrou mais difícil do que se imaginava.

A IA é uma área de grande interesse científico e com muito a se trabalhar ainda. Segundo RUSSEL & NORVIG (1995), as definições de IA podem ser organizadas em quatro categorias, conforme Figura 2.1.



Figura 2.1 – Definições de IA

Dentro dessas quatro categorias há uma grande diversidade de definições que as caracterizam, podendo resumi-las em:

- Sistemas que pensam como humanos – Para se fazer um programa pensar como um humano é necessário utilizar modelos cognitivos de como os humanos pensam. Estes modelos podem ser desenvolvidos através de duas abordagens: alcançar a resposta correta ou raciocinar como humano.
- Sistemas que pensam racionalmente – As leis do pensamento remontam da Grécia antiga, quando o filósofo Aristóteles desenvolveu seus silogismos, que deram início aos estudos da Lógica. Com o desenvolvimento da Lógica Formal no final do século XIX e começo do século XX, obteve-se uma notação precisa para declarações sobre todos os tipos de coisas do mundo e as relações entre elas. Há dois problemas na aplicação da Lógica em IA: primeiro não é fácil transformar conhecimento informal em declarações em termos formais; segundo há uma grande diferença entre estar apto a resolver um problema e resolvê-lo.
- Sistemas que agem como humanos – Nessa área estão os desenvolvimentos de técnicas e aplicações que envolvem as seguintes capacidades: processar linguagem natural, representação do conhecimento para armazenar informações, raciocínio automatizado para usar as informações armazenadas e aprendizado para se adaptar a novas circunstâncias.
- Sistemas que agem racionalmente – Agir racionalmente significa agir de tal maneira a alcançar um objetivo dadas algumas condições. As vantagens dessa abordagem é que ela é mais geral que as leis do pensamento e não é baseada no comportamento humano.

O conceito de Agente Inteligente (AI) surgiu dentro da categoria dos sistemas que agem racionalmente, sendo que um agente é uma entidade de software autônoma capaz de perceber seu ambiente através de sensores e agir neste ambiente através de atuadores (Figura 2.2), para tal é necessário fazer inferências corretas, mas não apenas isso, é necessário também a habilidade de representar

conhecimento, expressar resultados em linguagem natural, aprender e sentir o mundo, etc.

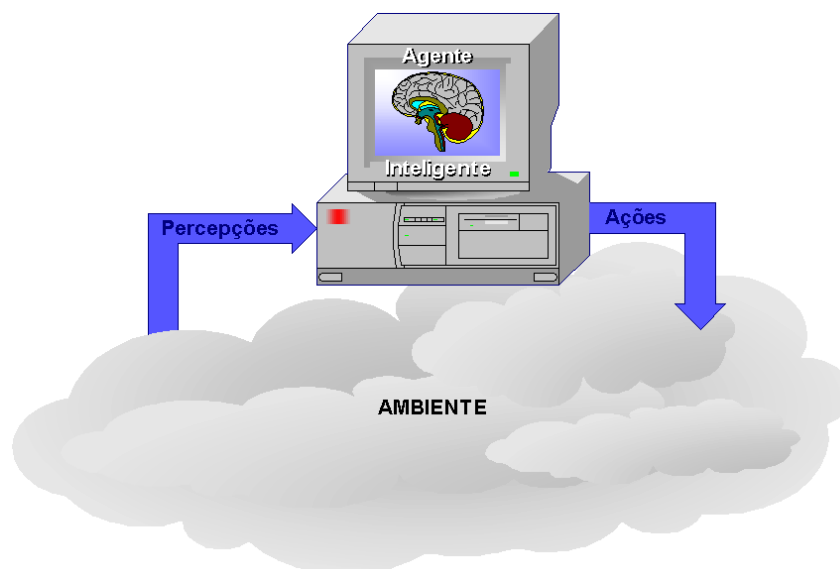


Figura 2.2 – Conceito geral de agente inteligente

O que é racional em um certo momento depende de quatro fatores:

- A medida de desempenho que define o grau de sucesso;
- Tudo o que o agente percebeu anteriormente, chamada de seqüência de percepções;
- O que o agente sabe sobre o ambiente;
- As ações que o agente pode desenvolver.

Não há um conceito definitivo e aceito por toda comunidade científica para agentes inteligentes (AIs), porém a idéia expressada acima traduz a concepção utilizada na implementação de todos os agentes implementados nesse trabalho.

2.2 Propriedades dos Agentes

As principais propriedades apontadas por WOOLDRIDGE & JENNINGS (1995) que são necessárias para caracterizar um AI são: autonomia, habilidade social, reatividade e iniciativa própria.

2.2.1 Autonomia

Um agente é autônomo quando opera sem a intervenção direta de humanos ou outras entidades computacionais, e possui algum tipo de controle sobre suas ações e estados internos.

Esta autonomia, ou poder de controle, sobre suas ações é oriunda do conhecimento concedido e aprendido pelo AI. Dessa forma, podem-se utilizar diversas técnicas da IA para prover o AI de meios para raciocinar sobre suas ações, e ainda, de algoritmos de aprendizagem que permitam o AI aumentar o seu conhecimento através de suas percepções e das ações executadas pelo usuário.

2.2.2 Habilidade Social

A habilidade social confere ao AI meios para interagir com outros AIs, inclusive com humanos, através de uma linguagem de comunicação apropriada. Como acontece em toda sociedade, os AIs devem conversar na mesma linguagem seguindo uma gramática, ou um conjunto de regras de composição de mensagens.

Os elementos envolvidos na comunicação devem possuir o mesmo significado para todos os envolvidos no processo, caso contrário, o transmissor emite uma informação e o receptor entende outra [RUSSEL & NORVIG, 1995].

Mas não só a linguagem compõe essa característica, todo agente possui um mecanismo de tomada de decisão, que geralmente é racional do ponto de vista do indivíduo, porém deve ser implementada uma estrutura que permita que essa estratégia de tomada de decisão englobe as características sociais e do ambiente onde o AI está inserido. Este mecanismo permite ao agente variar o seu grau de sociabilidade de acordo com as suas percepções do ambiente e dos outros agentes [HOGG & JENNINGS, 2001].

2.2.3 Reatividade

Reatividade é a propriedade que garante uma resposta às modificações ocorridas no ambiente em tempo hábil, esta propriedade está muito

relacionada com a propriedade de continuidade temporal, a qual permite ao agente um estado constante de prontidão. Situação semelhante ocorre nos seres humanos, que no momento de ler um jornal, tomar um banho, ou dormir, podem perceber que algo diferente ocorreu como um grande estrondo, ou alguém que chama, dando-se início a um conjunto de ações, que permitem ao Homem tomar as devidas atitudes [ETZIONI & WELD, 1995].

2.2.4 Iniciativa Própria

Um agente não simplesmente age somente em resposta às mudanças em seu ambiente, mas exhibe comportamento baseado em seus propósitos, tomando iniciativa pertinente.

Essa característica confere ao agente a possibilidade de apresentar comportamento intencional, ou seja, comportamento baseado em crenças, objetivos, habilidades, etc. Esse tipo de metodologia compõe uma teoria formal que especifica as propriedades do AI e dá suporte a diversos modelos de agentes encontrados atualmente.

2.3 Arquitetura de um Agente Inteligente

A arquitetura de um agente depende do seu tipo e função, e do dispositivo computacional no qual ele será instalado e irá trabalhar, o qual pode ser um computador pessoal, uma rede ou um hardware dedicado à aplicação.

De uma forma geral, a arquitetura é responsável pelos sinais dos sensores ficarem disponíveis para o AI, e ainda, é responsável por executar o programa inteligente do agente e alimentar os atuadores com as ações escolhidas pelo agente. O programa inteligente deve ser compatível com a arquitetura do sistema computacional no qual este será inserido.

De acordo com RUSSEL & NORVIG (1995), a relação entre agente, programa inteligente e arquitetura é dada por:

AI = Arquitetura + Programa Inteligente

Para a construção do programa inteligente é necessário definir os limites de percepção e ação do agente, assim como ter conhecimento de suas metas e medidas de desempenho, além de conhecer o ambiente no qual ele irá operar. Na Tabela 2.1 são apresentados alguns exemplos.

Tabela 2.1 – Exemplos de tipos de agentes e suas descrições

Tipo de Agente	Percepções	Ações	Objetivos	Ambiente
Médico	Sintomas, respostas do paciente e observações	Perguntas, exames e tratamento	Curar o paciente	Clínica, hospital e paciente
Buscador	Palavras-chave	Buscar, analisar e classificar itens	Achar itens que contém as palavras-chave	Internet
Operador de Sistema Elétrico	Estados de Disjuntores, chaves, proteções, alarmes, etc.	Manobras, supervisão, restabelecimento	Manter o fornecimento de energia	Centro de operações e sistema elétrico

Uma vez definidos o ambiente, o tipo de agente e suas características, começa a se pensar nas suas metas e medidas de desempenho. Essas medidas não fazem parte do corpo do agente, elas são atribuições externas, definidas de acordo com as necessidades do ambiente.

Uma maneira de se definir as medidas é inserir uma análise de um especialista no assunto da aplicação para se quantificar o sucesso das decisões. A análise da atuação do agente face às modificações no ambiente percebidas e a comparação com padrões de qualidade pré-estabelecidos mostra o quão feliz ele conseguiu ser.

Neste trabalho foram empregados agentes com a estrutura apresentada na Figura 2.3.

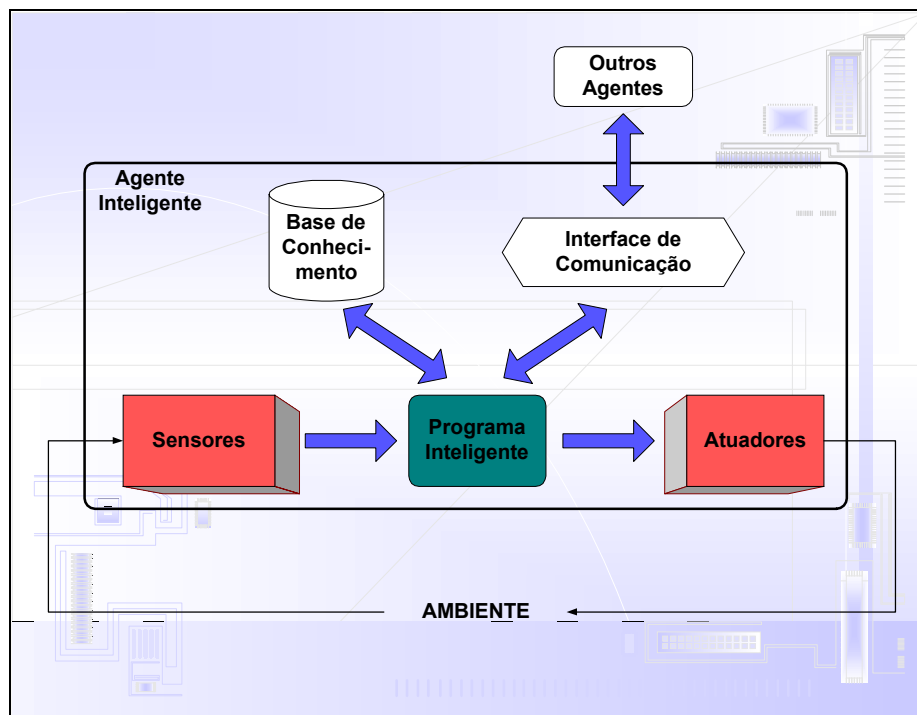


Figura 2.3 – Arquitetura de um AI utilizada

2.3.1 Sensores

Os sensores são responsáveis pela entrada (aquisição) de dados do ambiente para o AI. Nenhum sistema reage ao meio em que se encontra caso não exista uma maneira de saber o que está ocorrendo nesse meio, portanto a necessidade do AI receber informações do ambiente é suportada por este módulo, o qual contém os serviços controladores dos sensores de Hardware (placas de aquisição de dados) e Software (bancos de dados).

2.3.2 Atuadores

Os atuadores são os meios para que o AI possa interagir com o ambiente após receber e processar as informações do meio. O AI pode responder ao ambiente e inclusive modificá-lo através de seus atuadores (placas de entrada/saída).

2.3.3 Programa Inteligente

Um AI deve conhecer o seu mundo e raciocinar sobre seus possíveis cursos de ação. E ainda, pode aceitar novas tarefas e metas, adquirir e atualizar conhecimento baseado nas mudanças do ambiente e se adaptar a essas mudanças. Esse agente precisa ter noções sobre o estado corrente do mundo, saber como inferir informações não visíveis a sua percepção do mundo, entender como o mundo se modifica com o decorrer do tempo, o que ele quer desenvolver e que ações deve tomar em circunstâncias variadas.

Para isto, o programa inteligente contém as regras de produção, ou seja, o método responsável por armazenar o conhecimento de um AI. Pode ser utilizada qualquer técnica da IA para implementar uma certa inteligência no agente, dependendo da aplicação e do grau de autonomia dado ao mesmo.

2.3.4 Base de Conhecimento

O AI tem como núcleo uma Base de Conhecimento (BC), que é um conjunto de representações de fatos sobre o mundo, onde cada representação é chamada de sentença. As sentenças são expressas na chamada linguagem de representação de conhecimento e a aprendizagem se dá através do processo de atualização da BC.

A BC também contém regras para interpretação dos dados provenientes do ambiente, os objetivos que agente deve alcançar e o conhecimento da comunidade de agentes onde ele está inserido.

2.3.5 Interface de Comunicação

A Comunicação entre os agentes é alvo de muitas pesquisas e as linguagens comuns não apresentam características específicas para este fim, o que dificulta de forma significativa os processos de desenvolvimento de Sistemas Multi-Agente (SMA). De uma forma geral, pode-se afirmar que dois agentes podem comunicar-se se possuírem uma linguagem comum ou usarem linguagens inter-traduzíveis.

O processo de comunicação é baseado em protocolos, que são um conjunto de regras que normaliza e controla a troca de mensagens entre as unidades. Para interagir com uma sociedade, o AI deve conhecer o protocolo utilizado por ela.

A interface de comunicação é responsável por prover meios do AI se comunicar com outros agentes. Esse módulo pode ser inerente na linguagem utilizada para desenvolver o AI, ou pode-se utilizar ainda de padrões como Knowledge Query and Manipulation Language – KQML (FININ *et al.*, 1994a).

2.4 Tipos de Agentes

De acordo com a sua estrutura interna, os agentes podem ser distribuídos em quatro grupos distintos [RUSSEL & NORVIG, 1995]: agentes reflexivos simples, agentes que acompanham o mundo, agentes baseados em metas e agentes baseados em satisfação.

2.4.1 Agentes Reflexivos Simples

O agente age de forma reflexiva à verificação de qualquer mudança no estado do ambiente. Uma vez detectada uma variação, ele busca a primeira regra que corresponda a essa descrição de estado para atuar no ambiente, Figura 2.4.

2.4.2 Agentes que Acompanham o Mundo

Nesse tipo de agente, a percepção atual é combinada com o antigo estado anterior, de forma a gerar uma descrição atualizada e seqüencial dos estados. O agente incorpora uma espécie de previsão do que o fato percebido gera no ambiente, de acordo com estados anteriores guardados na memória. Baseado nessa visão e interpretação, ele escolhe a regra a ser utilizada, Figura 2.5.

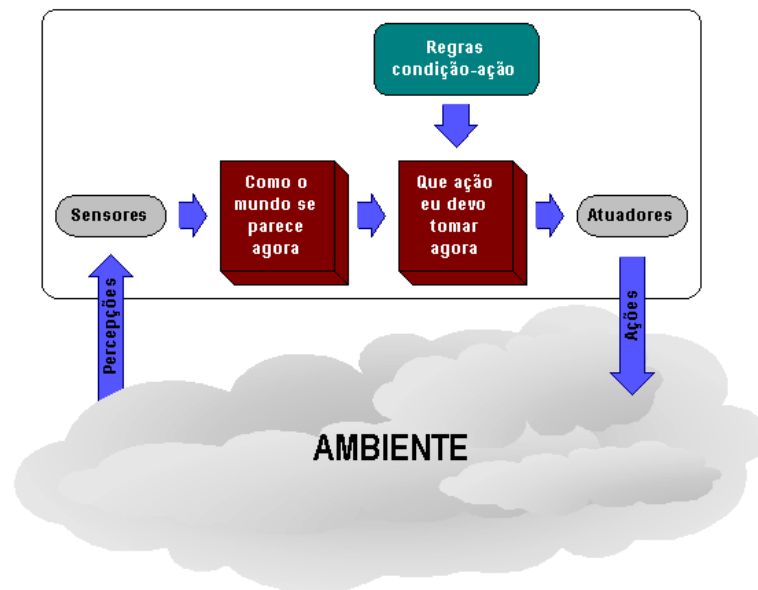


Figura 2.4 – Agentes reflexivos simples

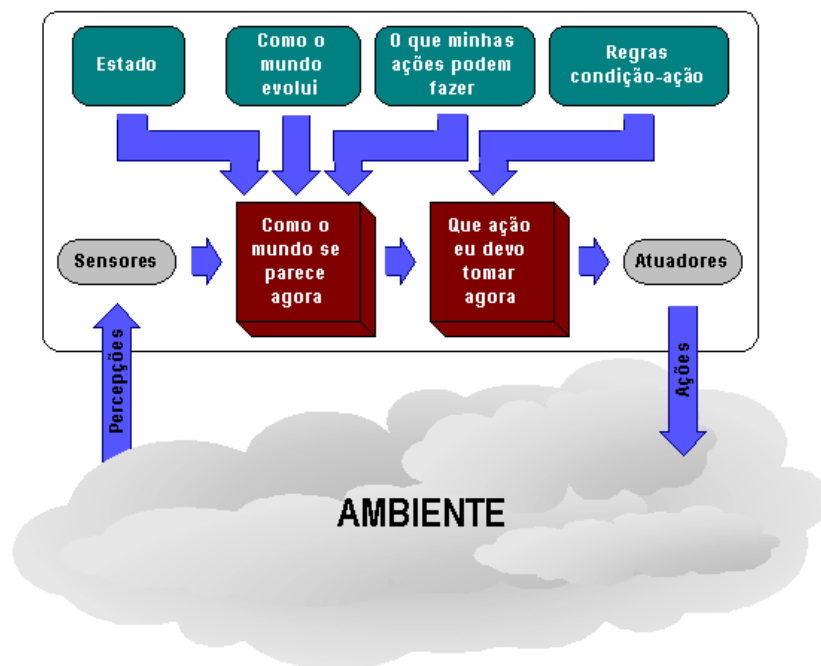


Figura 2.5 – Agentes que acompanham o mundo

2.4.3 Agentes Baseados em Metas

Esse tipo de agente assume que apenas saber o estado atual do ambiente e os estados anteriores não é suficiente para decidir que ação tomar. Para ele, é preciso ter definida uma meta. Ele precisa ter uma noção do resultado que

busca, precisa saber o que precisa fazer, para isso ele armazena algum tipo de meta na memória e, a partir dela, busca uma solução para alcançá-la.

Não há preocupação com o desempenho, e ainda, nesse tipo de programa, o programador já embutiu a ação correta para diversos casos previsíveis. Se houver uma mudança imprevista no ambiente, o agente reescreve uma lista de regras condição-ação, buscando novos destinos e gerando um novo comportamento, Figura 2.6.

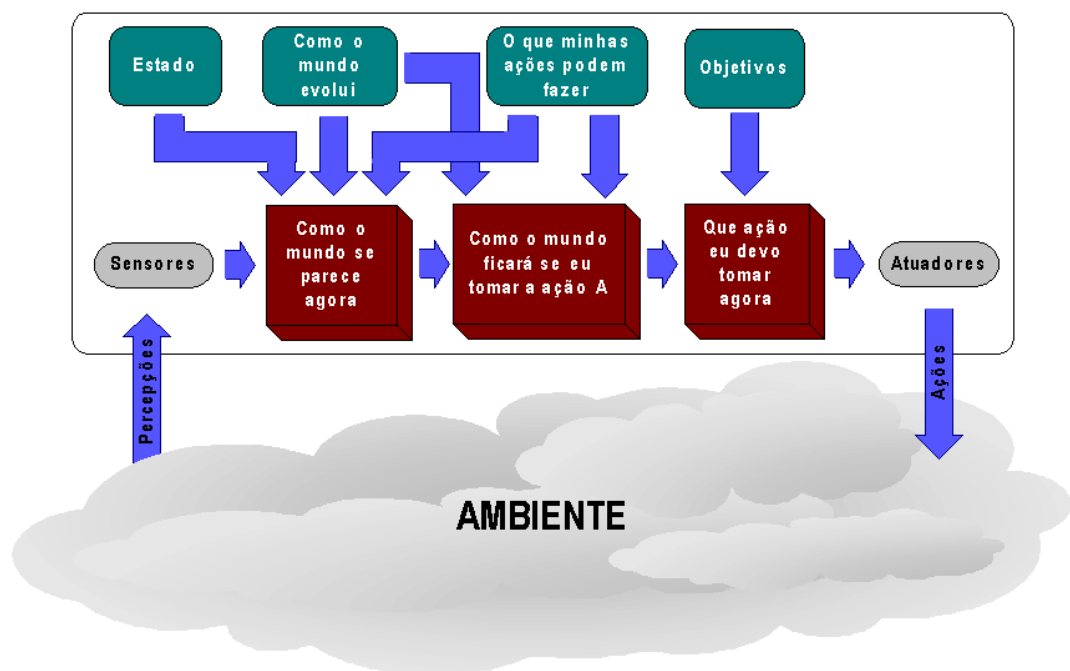


Figura 2.6 – Agentes baseados em metas

2.4.4 Agentes Baseados em Satisfação

São agentes que buscam maximizar o seu grau de satisfação. As decisões tomadas são sempre relacionadas com a satisfação que o agente terá com a sua atuação, Figura 2.7.

Apenas metas não são suficientes para gerar comportamento de qualidade é necessário incluir um sentido de eficiência. O AI busca o caminho de menor custo permitindo decisões em casos onde há conflito de metas e somente uma pode ser alcançada escolhendo o descarte. E também, em situações em que há várias metas que o agente pode buscar, e nenhuma delas pode ser escolhida com

certeza, a satisfação permite a determinação do caminho que leva à probabilidade de maior sucesso.

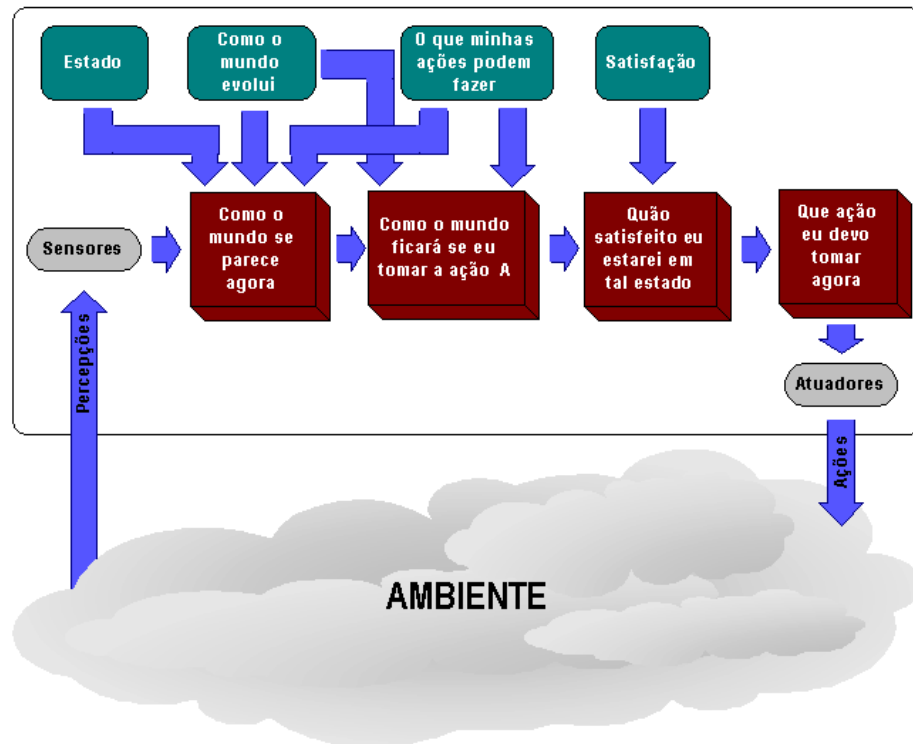


Figura 2.7 – Agentes baseados em satisfação

2.5 Sistemas Multi-Agente Inteligente

Um SMA tem como premissa básica prover mecanismos para a criação de sistemas inteligentes a partir de entidades de software autônomas, chamadas de agentes. Esses agentes interagem através de um ambiente, compartilhado por todos os membros do SMA, e sobre o qual eles atuam alterando o seu estado.

Uma vez que cada Agente Inteligente (AI) possui uma certa habilidade e objetivos próprios, o SMA tem que ser capaz de prover mecanismos para interação e coordenação dessas entidades. Dessa forma, o SMA consegue resolver problemas maiores que os domínios de conhecimento de seus integrantes.

Os SMA podem ser classificados como reativos e cognitivos. Os sistemas reativos são compostos por um grande número de agentes muito simples

que interagem entre si, sem possuir uma representação explícita do estado do ambiente e dos outros agentes, nem de suas ações passadas [FERBER, 1999]. A principal influência neste tipo de SMA vem da entomologia, ciência que estuda o comportamento de insetos. Já os sistemas cognitivos em geral possuem poucos agentes, dado que cada agente comporta-se de maneira racional, possui uma concepção lógica simbólica do mundo e planeja suas estratégias de ação de acordo com um forte mecanismo de inferência lógica. Esse tipo de agente, além de utilizar técnicas de dedução e aprendizado, considera também aspectos característicos da vontade humana, como crença, desejo e intenção [WOOLDRIDGE & JENNINGS, 1995].

Segundo GASSER & BOND (1988) para se obter um comportamento coerente em um SMA, os agentes deveriam não somente compartilhar conhecimento sobre o problema e as soluções, mas também raciocinar sobre o processo de coordenação entre os agentes. HEWITT (1977) afirma que em um SMA não há como se implementar um mecanismo de controle global, nem um conhecimento consistente global, nem um critério de sucesso global, ou até mesmo uma representação global do sistema, dessa forma a tarefa de coordenação se torna bastante difícil.

SYCARA (1998) sintetizou diversos estudos dos mais renomados pesquisadores da área e apontou as principais características de um SMA:

- Cada agente possui informações ou capacidades incompletas para resolver o problema, portanto possui um escopo de atuação limitado;
- Não há controle global no sistema;
- Os dados são descentralizados;
- A computação é assíncrona.

GASSER (1991) apontou as dificuldades inerentes encontradas na implementação de um SMA, elas serão apresentadas abaixo e discutidas nos próximos itens:

- Comunicação – Como habilitar os agentes para se comunicarem? Qual protocolo de comunicação utilizar?
- Interação – Qual linguagem os agentes devem utilizar para interagir uns com os outros e combinar seus esforços?

- Coerência e coordenação – Como garantir que os agentes se coordenarão uns com os outros para alcançar uma solução coerente para o problema que eles estão tentando resolver.

SYCARA 1998 apontou que apesar das dificuldades o interesse na pesquisa e desenvolvimento de SMAs é devido principalmente às propriedades relacionadas abaixo:

- Capacidade de resolver problemas que são muito grandes para serem resolvidos por sistemas centralizados seja por causa de limitações operacionais, ou pelo risco de se ter um desempenho ruim ou falta de confiabilidade por ser um sistema único;
- Permite a interconexão e interoperação de múltiplos aplicativos já existentes;
- Provê soluções para problemas que podem naturalmente ser tratados como uma sociedade de agentes interativos e autônomos;
- Provê soluções que usam fontes de informação espacialmente distribuídas de maneira eficiente, e ainda quando o conhecimento também está distribuído;
- É capaz de melhorar o desempenho nos seguintes aspectos:
 - eficiência computacional devido ao fato de SMAs explorarem a computação concorrente;
 - confiabilidade devido à distribuição das capacidades entre diversos agentes, e a possibilidade de redundância dos agentes;
 - capacidade de ampliação devido à possibilidade de alteração do número de agentes e de suas capacidades;
 - robustez na tolerância de incertezas, uma vez que a informação é distribuída entre os agentes;
 - facilidade de manutenção devido à modularidade, o que torna mais fácil o trabalho de achar e corrigir erros de programação;

- capacidade de sempre responder mesmo que haja anomalias, as quais serão tratadas localmente e não serão propagadas para todo o sistema;
- flexibilidade porque agentes com diferentes habilidades podem cooperar entre si para resolver diversos problemas;
- reuso devido as diferentes habilidades dos agentes que podem ser integrados em outros sistemas para resolver diferentes problemas.

2.5.1 Comunicação entre Agentes

O processo de comunicação habilita os agentes a trocarem informações, e ainda, coordenarem suas ações e cooperarem entre si. Em SMA o problema da comunicação não se restringe somente à definição da linguagem a ser utilizada, mas também a uma estrutura de suporte ao processo de comunicação.

Há basicamente quatro tipos de estruturas para o processo de comunicação:

- Comunicação direta – Envolve estabelecer uma ligação direta com outro agente usando um protocolo do tipo TCP/IP, o qual garanta a segurança do processo de comunicação (Figura 2.8). Neste caso o agente tem que saber onde o outro agente está, ou dispor de um diretório de agentes onde fique registrado o endereço de todos os agentes do sistema [FIPA, 2001a].

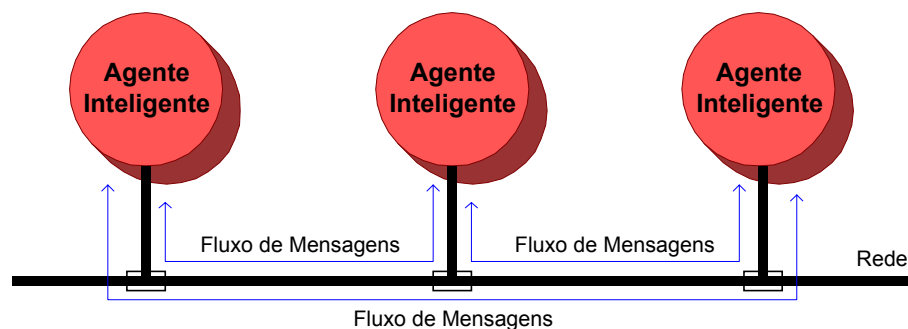


Figura 2.8 – Comunicação direta

- Sistemas confederados – Quando o número de agentes em um sistema fica muito grande o custo computacional envolvido na comunicação direta é proibitivo. Uma alternativa é utilizar sistemas confederados (Figura 2.9), onde os AIs se comunicam através de um Facilitador (agente mediador). O Facilitador é responsável por receber a mensagem do agente do seu sistema e roteá-la até o local adequado.

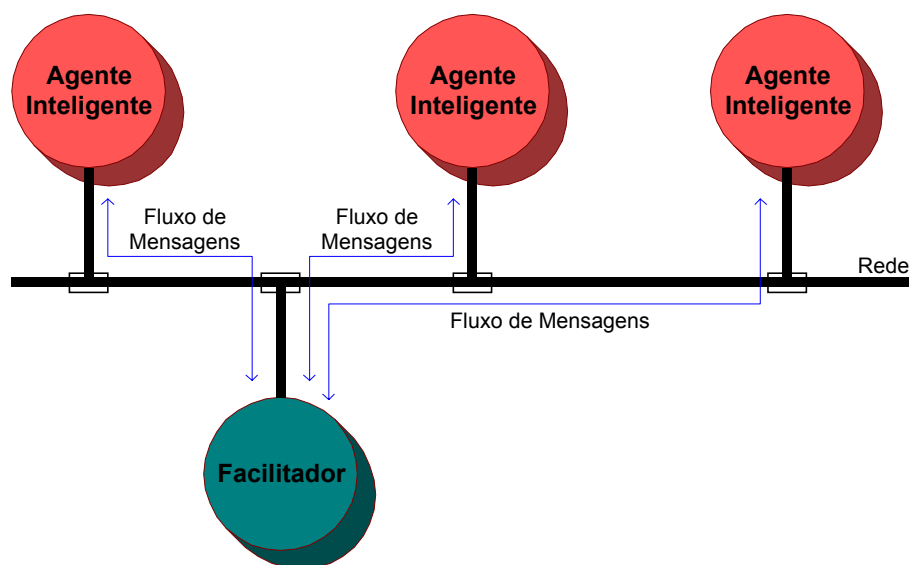


Figura 2.9 – Sistema confederado

- Comunicação difundida (*Broadcast Communication*) – Em situações onde a mensagem tem que ser comunicada a todos os agentes no sistema, ou o remetente não sabe a quem endereçar a mensagem então ele pode difundir a mensagem para todos os agentes (Figura 2.10). Quando a mensagem, ou o número de agentes, for grande haverá um consumo de banda significativo [SMITH, 1980] [DAVIS & SMITH, 1983].

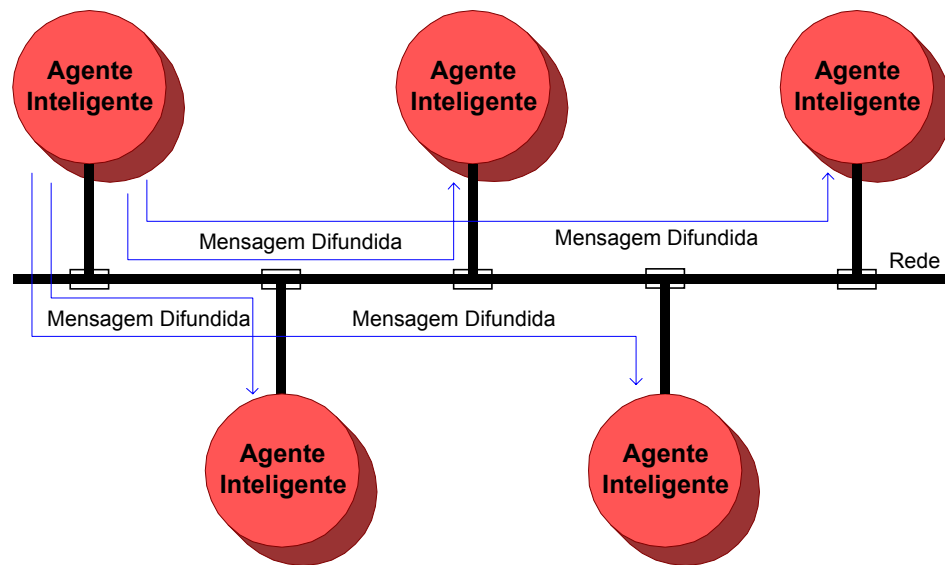


Figura 2.10 – Comunicação difundida

- Sistemas que usam quadro negro (*Blackboard Systems*) – Em IA um quadro negro é uma memória compartilhada, onde os agentes escrevem mensagens, colocam resultados parciais e obtém informações (Figura 2.11). Usualmente essa memória é particionada em diversos níveis de abstração, de tal forma a que um certo agente trabalhando num determinado nível de abstração tenha acesso ao correspondente nível. Deste modo, dados que foram gerados em qualquer nível podem ser comunicados para outros níveis [NII, 1986] [CORKILL, GAALLAGHAR & MURRAY, 1986].

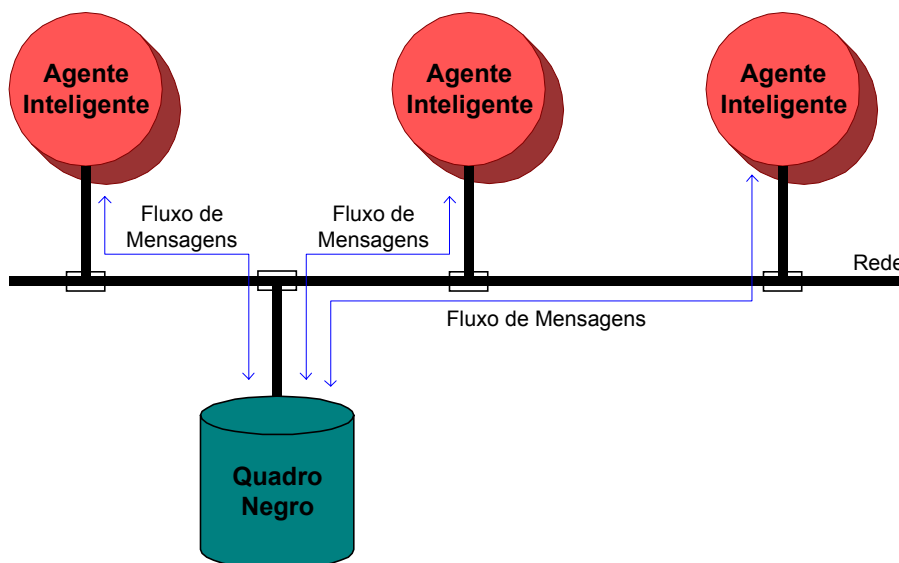


Figura 2.11 – Sistemas que usam quadro negro

2.5.2 Interação dos Agentes

Um segundo aspecto importante que deve ser considerado para implementação de SMA é a interação. GASSER & BOND (1988) definem a interação como sendo um tipo de ação coletiva onde um agente executa uma ação ou toma uma decisão, a qual foi influenciada pela presença ou conhecimento de outro agente. Como os SMA são geralmente baseados em objetivos, muitas interações serão estabelecidas por esses objetivos.

A natureza heterogênea e distribuída dos SMAs faz com que a implementação da interação entre agentes seja uma tarefa difícil [MULLER, 1994]. Logo, é importante utilizar uma linguagem comum e independente dos agentes. Os AIs se comunicam através de uma interface de mensagens comum que é independente da sua estrutura interna de dados e da sua base de conhecimento, portanto a linguagem deve ser capaz de prover meios de trocar dados e conhecimento entre todos os agentes. Sendo assim, cada agente deve ter uma camada lingüística responsável pela semântica independente dele, a qual é capaz de fazer a interface baseada em mensagens entre os agentes.

Na comunidade científica de SMAs o método mais comum de desenvolvimento da camada lingüística é baseado na Teoria dos Atos de Fala (*Speech-Act Theory*) [AUSTIN, 1988]. A Teoria dos Atos de Fala trouxe grandes

contribuições no entendimento das relações entre os estados internos dos agentes e das expressões que eles trocam com outros agentes [COHEN & LEVESQUE, 1995]. Ela concebe a comunicação e a interação como uma estrutura de objetivos para expressões, conhecimento dos AIs e ações planejadas para modificar esse conhecimento, e provê uma descrição unificada da ação e comunicação, onde a comunicação é tratada como uma ação, uma vez que ela traduz as crenças, desejos e intenções dos agentes [SEARLE, 1990].

Uma das linguagens mais comuns usadas pela comunidade científica de SMAs é a KQML, a qual facilita a cooperação e interoperação de alto nível entre agentes [EXTERNAL INTERFACES WORKING GROUP ARPA KSE, 1993]. Através da KQML os agentes podem pesquisar de simples programas e bases de dados até os mais sofisticados sistemas baseados em conhecimento e se comunicar com eles através do uso de performativas classificadas nas seguintes categorias básicas [FININ *et al.*, 1994b]:

- Performativas básicas de consulta: *evaluate, ask-if, ask-all, ...*
- Performativas de consulta multi-resposta: *stream-in, stream-all, stream-about, ...*
- Performativas de resposta: *reply, sorry, error, ...*
- Performativas genéricas de informação: *tell, achieve, cancel, ...*
- Performativas geradoras: *standby, ready, next, ...*
- Performativas de definição de capacidades: *advertise, subscribe, monitor, ...*
- Performativas de rede: *register, unregister, broadcast, ...*

Uma descrição geral semântica de uma performativa KQML é constituída de [LABROU & FININ, 1994]:

- Uma descrição da linguagem natural do significado intuitivo da performativa;
- Uma expressão na nossa lógica que descreve o ato em questão, esta é uma representação formal da descrição da linguagem natural;
- Pré-condições que indicam o estado necessário para um agente enviar uma performativa e para o outro aceitá-la e processá-la;

- Pós-condições que descrevem os estados dos agentes depois do envio da performativa (agente remetente) e do aceite da mesma (agente destinatário);
- Condições complementares para o agente remetente que indicará o estado final do mesmo, depois de possivelmente uma conversa ter sido realizada e da intenção sugerida pela performativa ter sido satisfeita;
- Qualquer comentário em linguagem natural que for cabível para melhorar o entendimento da performativa.

KQML inclui também uma arquitetura para plataformas de comunicação entre agentes baseada na presença de facilitadores, que são agentes que tem conhecimento sobre quais AIs estão acessíveis e quais são as habilidades de cada um. Isto permite a construção de uma sociedade aberta, ou seja, agentes entram e saem da sociedade bastando para isso avisar o facilitador. Finalmente, KQML induz a um conjunto de protocolos, que são seqüências típicas de mensagens que são normalmente trocadas entre agentes a partir de uma interação iniciada por algum agente pelo envio de algum tipo específico de mensagem.

A especificação da *Foundation for Intelligent Physical Agents* (FIPA) para sua estrutura de mensagens FIPA ACL [FIPA, 2001b], intencionalmente baseada em KQML, define explicitamente um protocolo de interação como um ambiente multi-agente compartilhado contendo atos comunicativos [FIPA, 2001c]. A especificação adota uma linguagem semântica para representar o conteúdo da mensagem, onde as proposições são expressas em uma lógica de atitudes mentais e ações, formalizada em uma linguagem modal de primeira ordem com identidade [JENNINGS, 1993].

2.5.3 Coerência e Coordenação dos Agentes

Por fim, outro aspecto importante na implementação de SMAs é garantir ao sistema coerência e coordenação. GASSER & BOND (1988) diferenciam ambas referenciando coerência quando se trata de quão bem o sistema inteiro se comporta enquanto resolve um problema, ou seja examinando o comportamento do

sistema como um todo. Já a coordenação é a propriedade da interação entre um conjunto de agentes desenvolvendo uma ação coletiva.

As três principais razões propostas por JENNINGS (1996) pelas quais as ações de múltiplos agentes devem ser coordenadas são:

- Há dependências entre as ações dos agentes, as quais podem ocorrer quando os objetivos individuais dos agentes são relacionados;
- Há uma necessidade de se encontrar restrições globais quando a solução em desenvolvimento pelo SMA deve satisfazer a certas condições para ser considerada de sucesso;
- Nenhum agente individual tem competência, conhecimento, informações suficientes para resolver o problema inteiro.

A coordenação das ações dos agentes possibilita o desenvolvimento da cooperação, a qual permite que uma comunidade de agentes especializados use suas capacidades para resolver problemas grandes, porém para isso haverá um custo de comunicação a mais [BIGUS & BIGUS, 1997].

2.5.4 Vantagens de Sistemas Multi-Agentes

A idéia de SMA oferece uma maneira de flexibilizar as restrições de controle seqüencial, centralizado e planejado para desenvolver sistemas que são descentralizados, distribuídos e concorrentes. DURFEE & MONTGOMERY (1989) observam que muitas aplicações são distribuídas, algumas espacialmente distribuídas e outras funcionalmente distribuídas. Nestes casos as vantagens de se usar sistemas baseados em agentes autônomos são:

- Tolerância a falhas – Os agentes são inerentemente mecanismos distribuídos e sendo assim um sistema baseado em agentes autônomos não falha se um ou mais de seus componentes falham [HATVANY, 1984];
- Software modular– Para se implementar um SMA é necessário particionar o problema, de forma que cada agente possa se filiar ao sistema, começar a operar junto a outros agentes, ou

deixar o sistema sem afetar a operação do sistema [PARUNAK, 1996];

- Sistemas autoconfiguráveis – Uma população de agentes pode se reconfigurar de acordo com a sua execução. Essa vantagem é muito importante em sistemas que podem mudar de configuração, pois os agentes mais apropriados a cada configuração podem ser acionados somente quando necessário [STEELS, 1990];
- Custo reduzido de software – Conforme o software se torna mais modular o custo de desenvolvimento e a sua complexidade é reduzida;
- Custo reduzido de hardware – Como os agentes podem se distribuir numa rede, a capacidade computacional deve ser dividida entre todas as máquinas que estarão executando agentes, ou seja, pode se usar máquinas com menos poder de processamento;
- Resolução de problemas mais rápida – Esta vantagem é devida ao fato do paralelismo existente no processamento em SMA;
- Comunicação reduzida – Pode-se construir uma sociedade de agentes onde apenas se transmite soluções parciais entre os agentes;
- Sistemas flexíveis – Pode-se construir uma sociedade de agentes com capacidades específicas que se unem para resolver problemas específicos de acordo com as características do problema.

2.5.5 Mais Informações sobre Agentes Inteligentes

Há uma grande quantidade de informações sobre AI e SMAs espalhadas pela Internet, isto se deve ao fato desses temas terem atingido um grande nível de reconhecimento na comunidade científica internacional. Algumas organizações foram criadas com o propósito de criar padrões para AIs, como por exemplo a FIPA e a *Agent Society*.

A FIPA possui um conjunto de especificações que estão relacionadas no Anexo B, essas especificações abordam os seguintes temas:

- Aplicações;
- Arquitetura Abstrata;
- Comunicação de Agentes:
 - Protocolos de Interação;
 - Atos Comunicativos;
 - Conteúdo de Linguagem;
- Gerenciamento de Agentes;
- Transporte de Mensagens de Agentes:
 - Representações ACL;
 - Representações Envelope;
 - Protocolos de Transporte.

Há uma diversidade de programas para desenvolvimento de agentes disponível na Internet, muitos deles se encontram relacionados na página da Rede de Excelência Européia de Computação Baseada em Agentes, disponível em <http://www.agentlink.org/resources/agent-software.html>.

3 A ARQUITETURA MULTI-AGENTE PROPOSTA

O desenvolvimento de programas computacionais atualmente já foi difundido em todas as áreas, uma vez que a maioria delas já está utilizando o computador como ferramenta de trabalho. Porém, a tarefa de desenvolvimento de software envolve mais do que implementações em linhas de código-fonte, manuais de usuários, sistemas de ajuda, etc.

Um software de qualidade precisa ser desenvolvido em uma arquitetura robusta e que aceite modificações; é necessário dispor de bons programadores, bem como de ferramentas de programação adequadas; é fundamental estar atento às necessidades do usuário, para que se possa focar de maneira precisa o trabalho de desenvolvimento; e ainda, é imprescindível empregar modelos que descrevam o sistema de todas as formas [BOOCH, RUMBAUGH & JACOBSON, 2000].

A estratégia de formação de modelos é empregada em todas as áreas da engenharia, e está baseada em quatro princípios básicos: (a) a escolha dos modelos influencia a maneira como um determinado problema é resolvido; (b) os modelos podem ser expressos em diferentes níveis de precisão; (c) os melhores modelos estão relacionados à realidade; e (d) qualquer sistema não-trivial será melhor investigado por meio de um conjunto de modelos. Com a modelagem é possível:

- Visualizar o sistema como ele é ou como se deseja que ele seja;
- Especificar a estrutura ou comportamento de um sistema;
- Direcionar o trabalho de construção do sistema;
- Documentar as decisões tomadas.

O trabalho de desenvolvimento da arquitetura multi-agente proposta iniciou-se com a busca de uma técnica robusta para a estruturação dos dados. Essa

técnica teria que ser capaz de atender às necessidades dinâmicas e estáticas do sistema em análise, e ainda prover o SMA de meios fáceis para a manutenção da sua base de dados. Para tal, foi empregado a Programação Orientada a Objetos (POO).

3.1 Programação Orientada a Objetos

A POO é um tipo de programação que busca modelar de forma fiel a topologia física do sistema no mundo real. Uma vez que os ambientes do mundo real são na maioria naturalmente distribuídos, a modelagem desse ambiente deve, conseqüentemente, ser distribuída. No paradigma de orientação a objetos, o mundo é visto como uma coleção de objetos interagindo com outros para alcançar um comportamento significativo [PANDIT, SOMAN & KHAPARDE, 2000].

A POO permite alcançar diversas características de um software de qualidade, como por exemplo:

- Habilidade para cumprir as tarefas de forma correta, como estiverem definidas em suas especificações;
- Habilidade para reagir de forma apropriada quando se encontrar sob condições anormais;
- Facilidade de adaptação a mudanças de especificação;
- Capacidade de reaproveitamento para aplicações em diversos processos de desenvolvimento;
- Compatibilidade, que é a facilidade de combinar elementos de software com outros, e que requer padronização de formato de arquivos, de estrutura de dados e de interface;
- Eficiência, que é a habilidade de gerenciamento dos recursos do sistema, como os dispositivos de comunicação e o espaço ocupado na memória;
- Facilidade de transferência de produtos de software para vários ambientes de software e hardware;
- Facilidade de uso, para pessoas de diversas formações e capacidades terem acesso ao uso do software para resolverem problemas;

- Habilidade para realizar testes e detectar falhas;
- Habilidade de proteger seus componentes, como programas e dados, contra acesso não autorizado e modificações [MEYER, 1997].

O uso da orientação a objetos muda os paradigmas da computação tradicional, pois troca o enfoque a funções por uma modelagem de classes e seus objetos relacionados. Uma classe é uma abstração da programação que captura atributos e comportamentos importantes para um conjunto de objetos similares. Já um objeto é uma instância de uma classe.

Graficamente, uma classe é representada por um retângulo, geralmente incluindo seu nome, atributos e operações, conforme Figura 3.1.

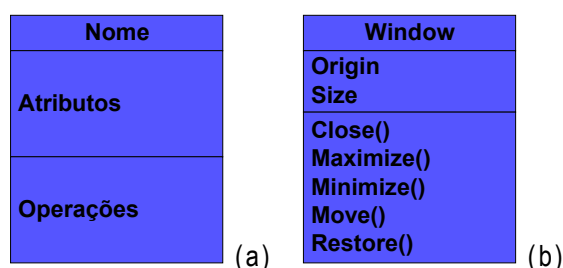


Figura 3.1 – Classes (a) Genérica e (b) Exemplo “Window”

A modelagem, segundo o paradigma da orientação a objetos, se aproveita dos benefícios que uma abordagem hierárquica pode oferecer. Segundo PANDIT, SOMAN & KHAPARDE (2001) a abordagem hierárquica prevê que os sistemas são compostos de apenas alguns tipos de subsistemas em várias combinações e arranjos, isso possibilita a redução de padrões.

A abstração empregada no processo de modelagem sugere o uso de relações entre objetos, destacando-se: associação, generalização e agregação.

A generalização ou herança identifica as similaridades entre uma classe e as versões mais refinadas dela. A classe da qual se parte é chamada classe base ou pai, e as classes refinadas a partir dela são chamadas subclasses ou classes filhas. Os atributos e métodos da classe base são herdados pela subclasse [BOOCH, 1999].

A associação é uma relação que descreve um conjunto de ligações com semântica e estrutura comuns, do mesmo modo que uma classe descreve um grupo de objetos com características comuns. As associações apresentam características bidirecionais, mas não necessariamente precisam ser implementadas em ambas direções. Uma associação ocorre quando uma classe apresenta interdependência consigo própria ou com outra classe, se possui uma ou mais instâncias originadas ou associadas a uma ou mais instâncias da outra.

Já a agregação é uma forma de relacionamento entre elementos de um modelo, que serve para expressar que cada objeto de um determinado tipo é uma combinação de outros objetos [MEYER, 1997].

3.2 A Arquitetura Multi-Agente Proposta

Para o desenvolvimento de um sistema multi-agente é necessário limitar a área de abrangência do domínio de aplicação, pois um SMA que opere muito bem numa certa aplicação, ou classe de aplicações, pode não ter o mesmo desempenho em outras áreas. A arquitetura proposta nesse trabalho visa a aplicação em sistemas reais que possuem as seguintes características principais:

- São geograficamente dispersos;
- Possuem natureza combinatória, ou utiliza heurísticas na sua análise;
- Possuem um certo grau de automação;
- Abrangem diversas áreas de conhecimento para tomada de decisões.

Um primeiro passo no desenvolvimento de uma arquitetura de um SMA é identificar os agentes. É comum nesse tipo de análise começar observando o sistema de fora para dentro, ou seja, de uma visão macro para as particularidades, estabelecendo uma lista de objetivos, os quais devem ser satisfeitos pelo SMA. Como um SMA é composto por diversas entidades que irão interagir entre si para alcançar esses objetivos, busca-se identificar os objetivos de cada agente, bem como os serviços que ele provê. Em seguida, dividem-se esses agentes em

diferentes categorias, dessa forma pode-se identificar os tipos de agentes empregados na aplicação.

A abordagem empregada na arquitetura proposta prevê duas categorias de agentes que são:

- **Agentes de Suporte:** responsáveis por prestar serviços que auxiliem o processo de tomada de decisões, como: estabelecer e coordenar a comunicação entre os agentes, manter um modelo computacional do mundo real atualizado, interface com o mundo real, entre outros.
- **Agentes Especializados:** detentores do conhecimento específico empregado na análise e solução dos problemas encontrados no domínio de aplicação.

Na categoria de agentes de suporte, considerando que o sistema físico real possua um certo grau de automação e as informações estarão contidas em algum tipo de base de dados num certo formato de apresentação optou-se por empregar dois agentes que são:

- **Agente de Interface:** responsável por acessar a base de dados automatizada, logo esse agente é responsável pela continuidade temporal do SMA gerando informações consistentes e atualizadas para os outros AIs. Neste agente está descrita a metodologia de interpretação dos dados, filtragem de dados irrelevantes e duplicados, minimizando assim o processo de comunicação e as tarefas de pré-processamento quando possível.
- **Agente Modelo:** contém o modelo orientado a objeto do sistema físico real e continuamente compara esse modelo com as informações providas pelo Agente de Interface atualizando-o. Além disso, o Agente Modelo grava um histórico dos eventos acontecidos para possibilitar análises cronológicas. E ainda, nesse AI está descrito qual informação deve ser enviada para qual agente.

Uma segunda tarefa é a especificação das conversações estabelecidas entre os agentes. Nesse caso, deve-se primeiro estabelecer qual tipo de comunicação será empregado, por exemplo: comunicação direta, difundida, etc. Na estrutura multi-agente desenvolvida, considerou-se a característica dos sistemas reais serem geograficamente dispersos e da possibilidade de expansão do SMA, logo optou-se pela metodologia de sistemas confederados, a qual inclui mais um agente na categoria de suporte, que é:

- **Agente de Comunicação:** responsável pela interconexão dos agentes, ele funciona como um centro de comunicação: todas as mensagens enviadas por qualquer AI passam por ele e são direcionadas para o respectivo destinatário. E ainda, avisa aos integrantes do SMA quais agentes estão ativos.

Pode-se considerar que uma conversação entre AIs está sempre relacionada com um plano para atingir um objetivo, portanto deve-se especificar qual informação deve ser trocada e com quem, pois um fluxo de informação muito grande pode deteriorar a capacidade de comunicação do meio empregado, o que resultaria numa perda de reatividade do SMA. Deve-se lembrar que nessas conversações são trocadas informações parciais do processo de tomada de decisão, logo conversações desnecessárias podem prejudicar o processo de tomada de decisões.

A definição dos agentes especializados está muito relacionada com o problema e com a forma que o desenvolvedor analisa e soluciona o mesmo. Por exemplo, supondo que o domínio de aplicação fosse o jogo de xadrez poder-se-ia adotar duas abordagens: a primeira, considerando cada peça como um AI que disputaria no SMA pela oportunidade de se movimentar, mais um agente responsável pelo controle dessa concorrência; a segunda, considerando cada tipo de peça (peões, cavalos, etc.) como um agente que dentro de sua classe escolhe o melhor candidato a se movimentar, mais um AI responsável por definir uma estratégia de jogo.

Então, a estrutura proposta nesse trabalho tem a seguinte configuração, Figura 3.2.

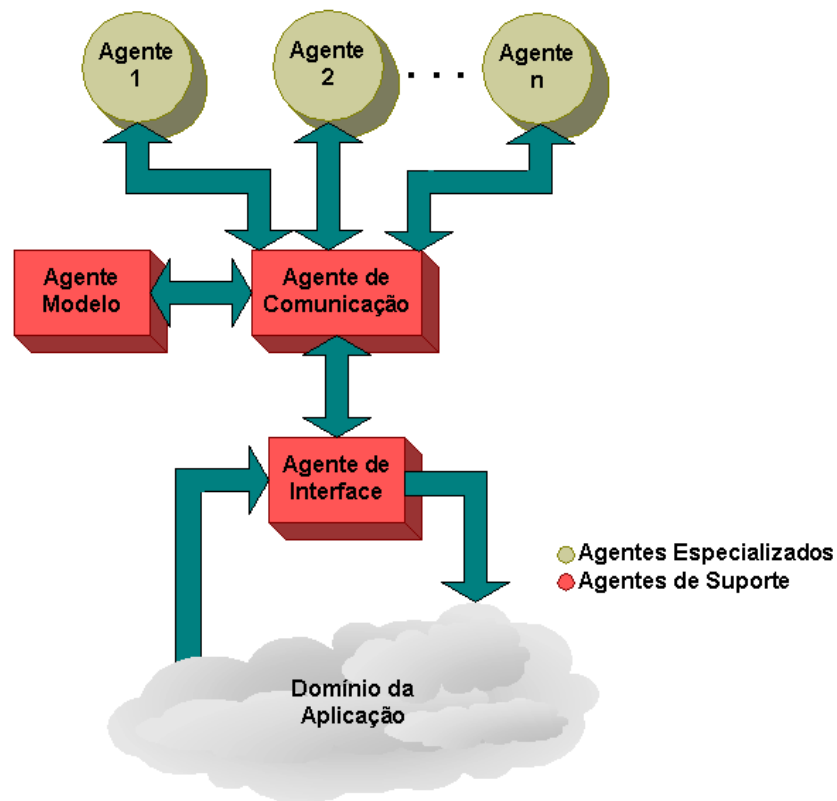


Figura 3.2 – Arquitetura multi-agente proposta

O desenvolvimento de aplicações de sucesso usando tecnologia de agentes requer técnicas que permitam reduzir o risco inerente em qualquer tecnologia nova. Duas possibilidades para colaborar nesse caso são:

- Apresentar a nova tecnologia como uma extensão de um método novo e confiável;
- Prover ferramentas de engenharia para dar suporte aos métodos já comprovados de desenvolvimento de aplicações.

A Linguagem Unificada de Modelagem (UML – *Unified Modeling Language*) unifica e formaliza diversos métodos e abordagens de engenharia de software baseada em orientação a objetos, permitindo o desenvolvimento de grandes aplicações industriais provendo ferramentas de suporte para análise, especificação e projeto do software de forma estruturada e documentada.

A concepção de agente pode ser encarada como um próximo passo para os objetos da POO, sendo assim pode-se utilizar a estrutura da UML para descrever os softwares baseados em Tecnologia de Agentes. Dessa forma

desenvolveu-se a *Agent UML (AUML)*, que constitui uma adaptação da UML para o desenvolvimento de SMA.

Dentre as diversas ferramentas providas pela AUML a que se faz necessária no contexto desse trabalho é o diagrama de interação entre agentes, o qual explicita a seqüência de ações dentro do SMA, bem como a colaboração entre eles.

O diagrama de interação da arquitetura multi-agente proposta nesse trabalho está apresentado na Figura 3.3.

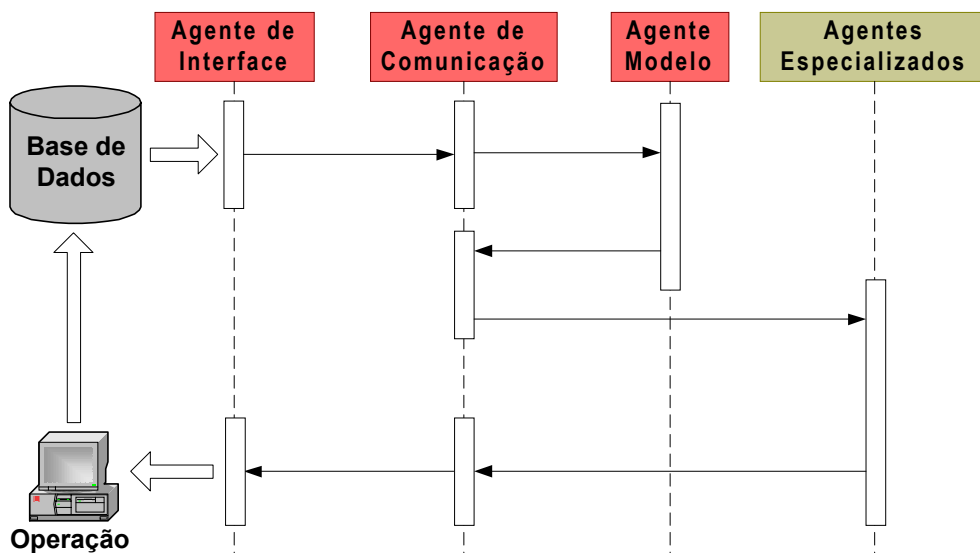


Figura 3.3 – Diagrama de Interação da arquitetura multi-agente proposta

Neste diagrama também são explicitadas todas as comunicações existentes entre os agentes. O ciclo de análise inicia com o “Agente de Interface” que ao perceber uma alteração na base de dados do sistema automatizado faz uma aquisição dos mesmos, interpreta-os e informa o “Agente Modelo” via “Agente de Comunicação”.

O “Agente Modelo” então atualiza o modelo orientado a objetos e envia simultaneamente os novos dados para os “Agentes Especializados”, através do “Agente de Comunicação”, os quais são responsáveis pelas análises dos mesmos e pela sugestão de ações de operação.

Um plano com as ações sugeridas pelos “Agentes Especializados” é então enviado para o “Agente de Comunicação” que o repassa para o “Agente de Interface”, o qual irá apresentá-lo para o usuário, ficando a cargo do mesmo aceitar ou não as ações sugeridas e promover as alterações no sistema real físico.

Após o usuário receber o plano de ações cabe a ele decidir adotá-las, e assim que alguma ação for tomada, ou que houver uma modificação na base de dados do sistema o SMA entra em operação novamente.

3.3 Exemplo de Implementação da Arquitetura Multi-Agente Proposta

É comum hoje em dia nas grandes cidades quando se precisa de um táxi ligar para uma central de atendimento e solicitar um veículo. A central opera diversos taxistas simultaneamente e com o avanço das tecnologias de localização via satélite pode-se precisar a posição destes e então determinar qual o melhor veículo para atender o chamado, Figura 3.4.

Essa tomada de decisão pode levar em consideração diversos fatores, por exemplo:

- A distância entre os veículos e o local de chamada;
- Possíveis rotas para esses veículos;
- Condições de trânsito e tempo estimado de espera.

3.3.1 Modelo Orientado a Objetos

Toda aplicação deve primeiramente ser modelada para dar suporte ao SMA. Nesse caso pode-se imaginar um modelo OO composto por uma classe inicial denominada “Ponto”, a qual representa a posição de qualquer entroncamento existente entre duas ou mais quadras.

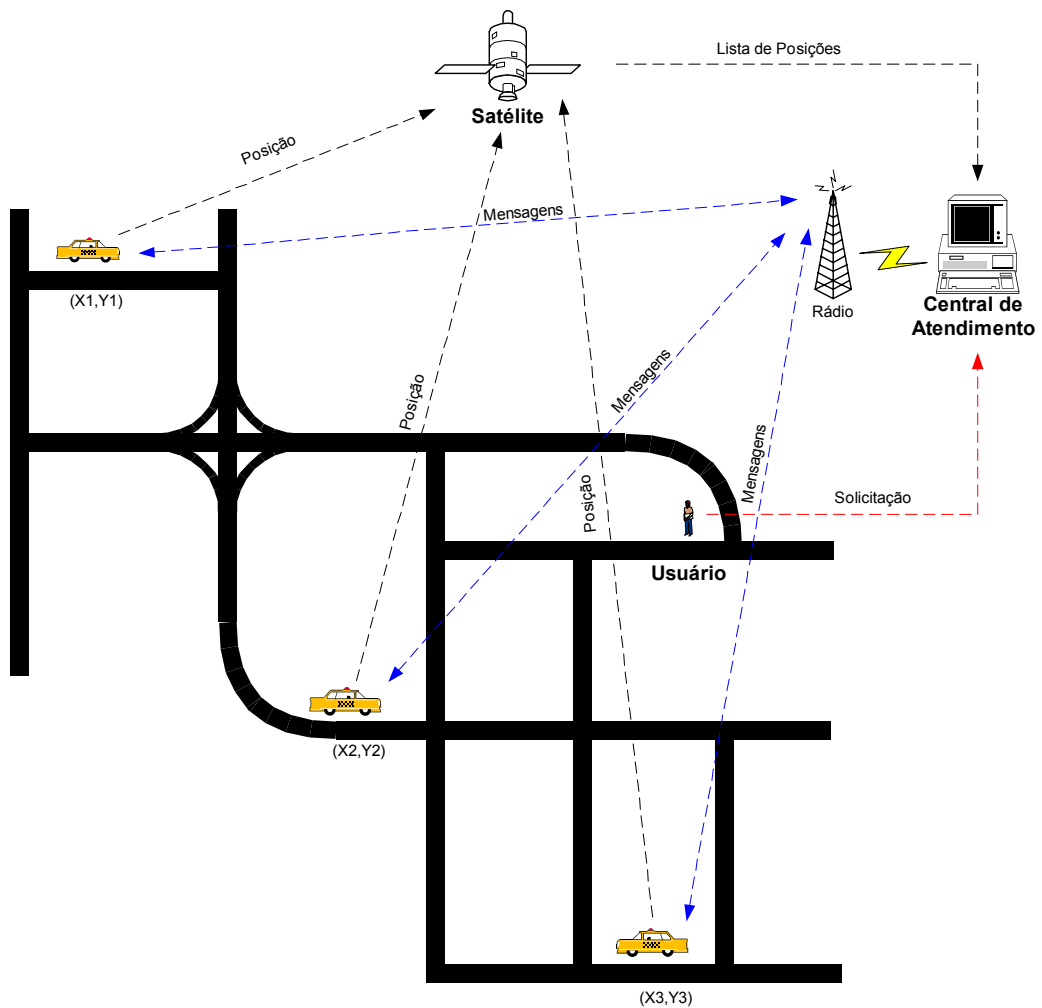


Figura 3.4 – Exemplo de aplicação da arquitetura proposta

Em seguida, tem-se uma classe denominada “Quadra”, que representa o menor trecho de uma rua; outra classe chamada “Rua”, que representa um agrupamento de quadras que possuem o mesmo nome; e ainda, uma classe “Bairro” que é formado por um conjunto de ruas e quadras e uma classe “Cidade” formada por um conjunto de bairros, Figura 3.5.

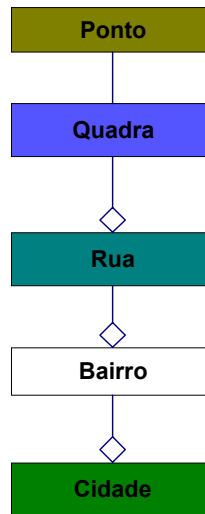


Figura 3.5 – Exemplo de modelo OO para o exemplo

Cada Classe possui um conjunto de atributos e se relaciona conforme Figura 3.6:

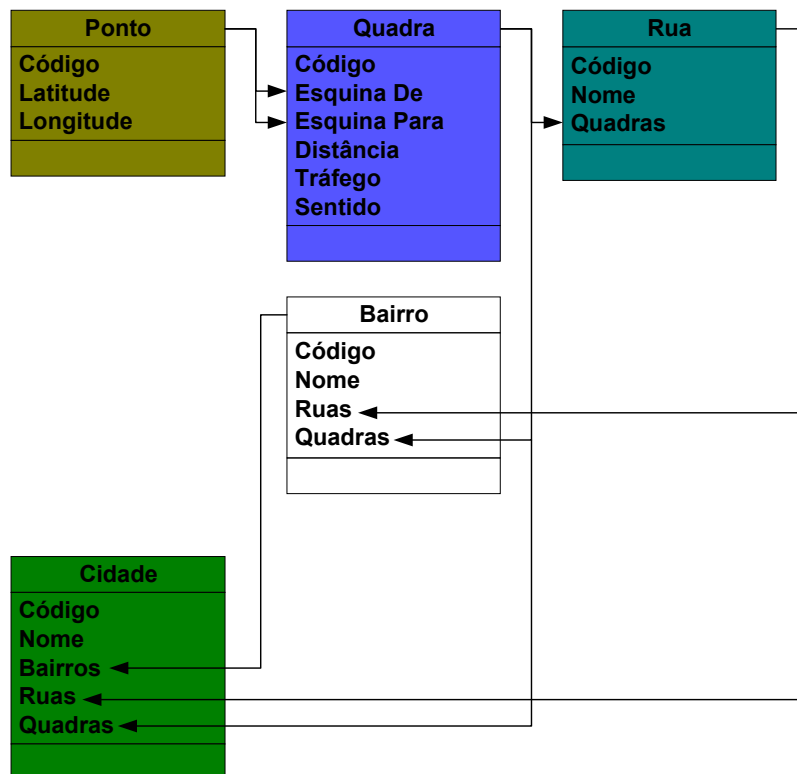


Figura 3.6 – Classes do modelo OO

As classes apresentadas na Figura 3.5 têm os seguintes atributos:

- Ponto:
 - Código: código do ponto;
 - Latitude: coordenada de latitude do ponto;
 - Longitude: coordenada de longitude do ponto;
- Quadra:
 - Código: código da quadra;
 - Esquina De: código do ponto da esquina de origem da quadra;
 - Esquina Para: código do ponto da esquina de destino da quadra;
 - Distância: distância em metros da quadra;
 - Tráfego: condições de tráfego na quadra;
 - Sentido: 1 – Mão única (De → Para); 2 – Mão dupla;
- Rua:
 - Código: código da rua;
 - Nome: nome da rua;
 - Quadras: conjunto de quadras que forma a rua;
- Bairro:
 - Código: código do bairro;
 - Nome: nome do bairro;
 - Ruas: conjunto de ruas que forma o bairro;
 - Quadras: conjunto de quadras que forma o bairro e que não foram contempladas na lista de ruas anterior;
- Cidade:
 - Código: código da cidade;
 - Nome: nome da cidade;
 - Bairros: conjunto de bairros que forma a cidade
 - Ruas: conjunto de ruas que forma a cidade e que não foram contempladas na lista de bairros anterior;
 - Quadras: conjunto de quadras que forma a cidade e que não foram contempladas na lista de ruas anterior.

3.3.2 Agentes Especializados

Para se determinar quais serão os agentes especializados é preciso elaborar a forma como será o processo de tomada de decisão. Nesse exemplo serão utilizados os seguintes AIs:

- **Agente Distância:** responsável por desenvolver um cálculo inicial da distância entre todos os carros disponíveis e o local da solicitação.
- **Agente Caminho:** determina possíveis caminhos para os melhores carros alcançarem o local da solicitação.
- **Agente Trânsito:** realiza uma ponderação na distância do caminho de acordo com as condições de tráfego e calcula o tempo necessário para atender a solicitação.

O fluxograma de ações do “Agente Distância” é apresentado na Figura 3.7. Em primeiro lugar, o agente irá receber a localização da solicitação vinda da Central de Atendimento. Essa localização está no formato: (Rua, Quadra).

Em seguida, o agente irá realizar a leitura da lista de posições de todos os veículos providas pelo sistema de localização via satélite. Essas posições estão no formato: (Latitude, Longitude).

O próximo passo é calcular a distância entre todos os carros e o local da solicitação. Para isso, é necessário adotar um dos dois pontos (“Esquina De” ou “Esquina Para”) da quadra como referência e utilizar uma metodologia de cálculo de distância esférica, que leva em consideração que os dois pontos estão dispostos sobre uma superfície esférica.

Como alguns carros estão perto do local e outros muito longe, no quarto passo pode-se utilizar uma função de avaliação para selecionar os melhores candidatos. Por exemplo, pode-se utilizar a menor distância calculada como parâmetro, todos os veículos que estiverem num raio de até duas vezes esta distância serão selecionados.

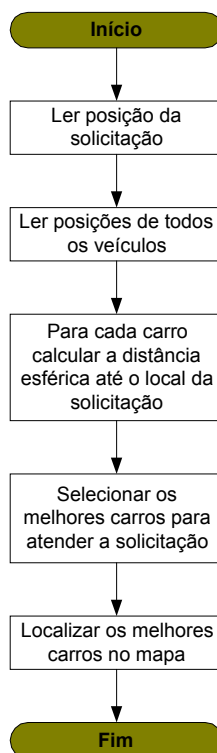


Figura 3.7 – Fluxograma de ações do Agente Distância

No último passo, os melhores veículos serão posicionados no mapa da cidade, ou seja, suas coordenadas passarão de (Latitude, Longitude) para (Rua, Quadra). Esse passo é necessário para se iniciar o procedimento de busca de rotas realizado pelo Agente Caminho.

O “Agente Caminho” pode ser um sistema especialista capaz de buscar rotas entre dois pontos. Podem-se implementar diversos tipos de métodos de busca, como por exemplo, busca em amplitude ou em profundidade, devendo-se observar o desempenho computacional dos mesmos.

Em VALIQUETTE, LAMBERT-TORRES & MUKHEDKAR (1991) é utilizado um método de busca em amplitude que segue o fluxograma da Figura 3.8. Esse algoritmo permite buscar diversos caminhos através do ajuste do fator de busca, o qual permite que o sistema especialista continue buscando mesmo depois de ter encontrado um primeiro caminho.

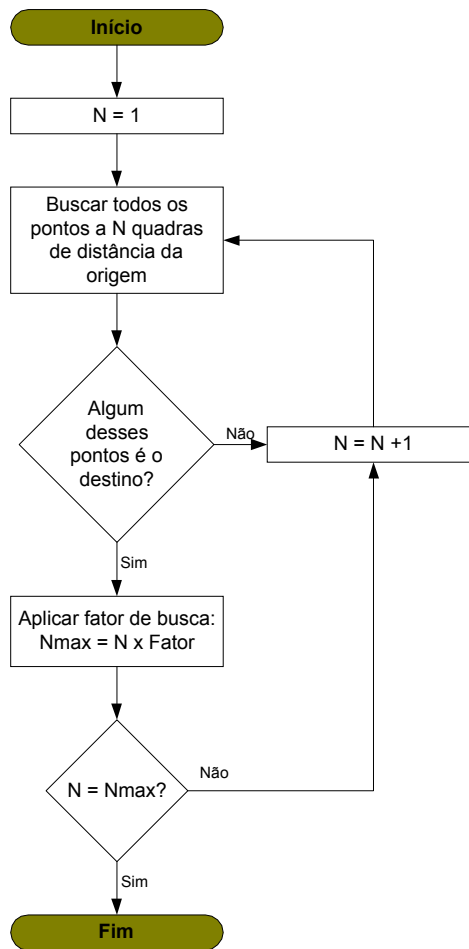


Figura 3.8 – Fluxograma de ações do “Agente Caminho”

Esse procedimento é repetido para cada veículo da lista de melhores candidatos fornecida pelo “Agente Distância”.

O “Agente Trânsito” provê experiência sobre as condições de tráfego e a distância a ser percorrida. Por exemplo, em São Paulo (capital) já se dispõe de informações *on-line* do trânsito nas principais vias da cidade, conforme o trânsito aumenta em um determinado trecho do percurso a velocidade média dos veículos diminui, sendo assim eles demoram mais tempo para percorrer tal trecho. Para efeito de análise esse fato é o mesmo que manter a velocidade média constante e aumentar a distância do trecho de forma que o tempo também aumente proporcionalmente.

Supondo apenas três condições de tráfego: normal, intenso (duas vezes mais que o normal) e parado (congestionado), pode-se utilizar um conjunto de regras de produção do tipo SE-ENTÃO como mostrado abaixo:

SE tráfego = normal ENTÃO distância nova = distância
SE tráfego = intenso ENTÃO distância nova = 2 * distância
SE tráfego = parado ENTÃO distância nova = 10 * distância

Essa análise é feita para cada rota obtida para os melhores veículos. Poder-se-ia também utilizar mais fatores nessa análise, por exemplo: a existência de semáforos, obstáculos, escolas, obras, etc., e ainda, desenvolver um sistema de conjuntos difusos permitindo assim o uso de variáveis lingüísticas enriquecendo o processo de tomada de decisão como um todo.

Após analisar as condições de trânsito o agente calcula a distância total de cada rota levando em consideração as distâncias novas obtidas na análise anterior, e então o tempo estimado para atender a solicitação é obtido fazendo:

$$\text{Tempo estimado} = \text{Distância total} / \text{Velocidade média}$$

É montada uma lista por ordem de tempo estimado crescente dos melhores veículos para atender à solicitação que é devolvida para a Central de Atendimento que escolherá um veículo e o informará via rádio da solicitação.

Esse exemplo foi apresentado dentro do âmbito desse trabalho de doutorado e foi implementado para a Companhia Energética de Brasília, considerando-se ao invés de táxis, viaturas da companhia, e ainda, ao invés de uma pessoa solicitando um táxi tem-se uma chave a ser manobrada. Mais detalhes da implementação computacional desse exemplo estão contidos no Anexo E.

3.3.3 Diagrama de Interações

Nesse exemplo, o diagrama de interações do SMA está apresentado na Figura 3.9.

O ciclo de análise inicia com o “Agente de Interface” que ao perceber uma alteração na base de dados do sistema automatizado faz uma aquisição dos mesmos, interpreta-os e informa o “Agente Modelo” via “Agente de Comunicação”.

O “Agente Modelo” então atualiza o modelo orientado a objetos e envia simultaneamente os novos dados para o “Agente Distância”, através do “Agente de Comunicação”, o qual calcula a distância entre os veículos e o local de solicitação e seleciona os melhores candidatos.

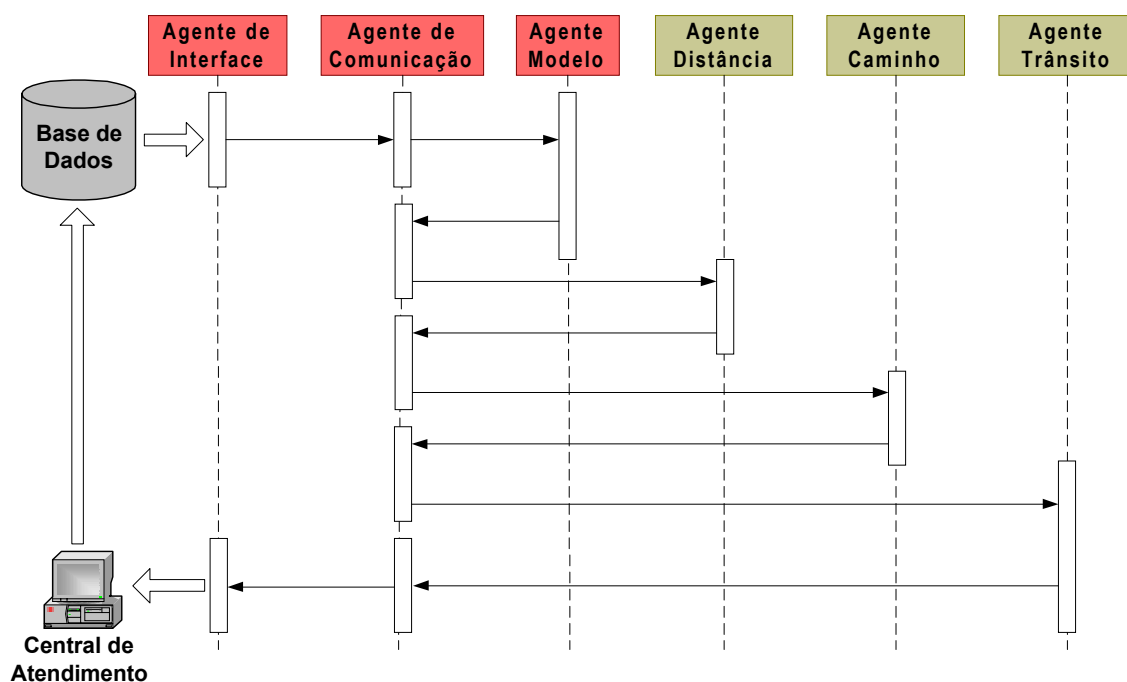


Figura 3.9 – Diagrama de interações do SMA

A lista dos melhores candidatos é passada do “Agente Distância” para o “Agente Caminho”, novamente através do “Agente de Comunicação”, o qual busca rotas para o atendimento da solicitação.

Em seguida, a lista dos melhores candidatos com as suas possíveis rotas é passada do “Agente Caminho” para o “Agente Trânsito” por intermédio do “Agente de Comunicação”. O “Agente Trânsito” analisa as condições de trânsito e estima o tempo para atendimento da solicitação.

Uma lista por ordem de tempo estimado crescente dos melhores veículos para atender à solicitação é então enviada para o “Agente de Comunicação” que a repassa para o “Agente de Interface”, o qual irá apresentá-la para o usuário, ficando a cargo do mesmo escolher o veículo e informá-lo pelo rádio.

4 IMPLEMENTAÇÃO DA ARQUITETURA MULTI-AGENTE PROPOSTA EM SISTEMAS ELÉTRICOS

4.1 Orientação a Objetos aplicada em Sistemas Elétricos

4.1.1 Embasamento Teórico

A identificação das classes e objetos é um processo interativo e pode ser classificado como uma análise do domínio da aplicação. No caso de sistemas elétricos ZHU & LUBKEMAN (1997) apresentam um modelo dos equipamentos do SEP bem sucedido, o qual serviu de base para os desenvolvimentos apresentados a seguir. A abordagem utilizada prevê o estabelecimento de uma classe para cada tipo de equipamento, que seja de interesse para a aplicação.

Segundo PANDIT, SOMAN & KHAPARDE (2000) o uso de programação orientada a objetos em aplicativos para o SEP se deve aos seguintes fatos: os sistemas a serem modelados são de larga escala; o volume de dados a ser manuseado é muito grande; e os requisitos de visualização (interface) do sistema claro, explicativo e objetivo.

Empregando a abordagem hierárquica o SEP é visto como um sistema complexo composto de subsistemas inter-relacionados que possuem dentro de si outros subsistemas. Dessa forma, a abstração busca isolar os mais elementares componentes do sistema [PANDIT, SOMAN & KHAPARDE, 2001].

4.1.2 A Estrutura de Programação Orientada a Objetos Proposta

Para a análise de sistemas elétricos foram definidas as classes apresentadas na Figura 4.1.

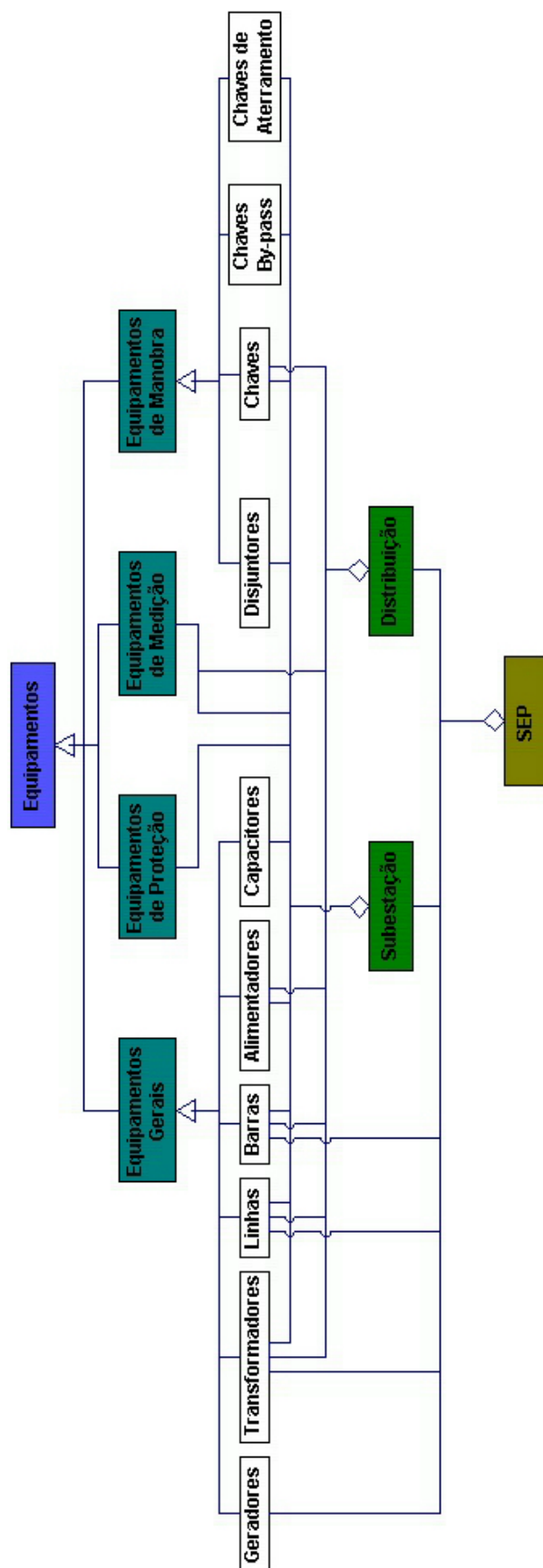


Figura 4.1 – Diagrama de classes utilizado

A qualidade do processo de abstração é fator crucial para o perfeito funcionamento da estrutura computacional, pois a POO enfoca a preocupação com a estrutura dos objetos. Apesar dos SEPs caracterizarem um ambiente dinâmico, a estrutura dos objetos permanece estável enquanto mudam seus detalhes de utilização e seus requisitos. Dessa forma, o modelo permanece preservado, atuando e aguardando nova atualização do estado do mundo.

A classe “Equipamentos” é o ponto de partida para o diagrama de classes do sistema, Figura 4.2. Ela define que todos os elementos fundamentais para o monitoramento, proteção e operação de SEPs são representados a partir dessa entidade no modelo computacional.

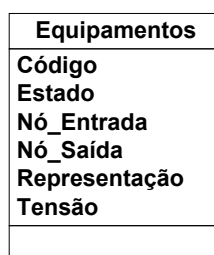


Figura 4.2 – Classe “Equipamentos”

A partir da definição da classe “Equipamentos”, utilizando as relações de associação, herança e agregação são construídas as demais classes. Os atributos e operações da classe “Equipamentos” são:

- **Código:** Código do equipamento;
- **Estado:** Verdadeiro ou falso, que conforme o caso pode representar: energizado ou desenergizado, atuado ou não atuado, aberto ou fechado, etc.;
- **Nó_Entrada:** Código do nó de entrada do equipamento;
- **Nó_Saída:** Código do nó de saída do equipamento;
- **Representação:** Figura que representa o equipamento.

Como essa classe representa o pai de todas as outras classes, esses atributos são herdados por todas as demais classes do modelo. Algumas das classes filhas podem possuir exatamente os mesmos atributos e operações da classe pai, mas algumas sofrem modificações. À medida que as classes vão se especializando,

elas podem receber a inclusão de atributos e métodos adicionais que definam suas características particulares.

De acordo com a abstração empregada, o sistema elétrico de potência é dividido em quatro classes principais ou quatro tipos de dispositivos. São classes filhas de “Equipamentos” as classes “Equipamentos Gerais”, “Equipamentos de Proteção”, “Equipamentos de Medição” e “Equipamentos de Manobra”, Figura 4.3. Algumas características a mais foram adicionadas nessas classes, conforme mostrado abaixo:

- Equipamentos Gerais:
 - Tensão: Classe de tensão do equipamento.
- Equipamentos de Proteção:
 - Função: Tipo de função de proteção desenvolvida pelo equipamento: sobrecorrente, subtensão, etc.
- Equipamentos de Medição:
 - Medida: Tipo de grandeza medida pelo equipamento;
 - Valor: Valor da medida realizada pelo equipamento;
 - Limite_Inferior: mínimo valor permitido da grandeza medida;
 - Limite_Superior: Máximo valor permitido da grandeza medida.

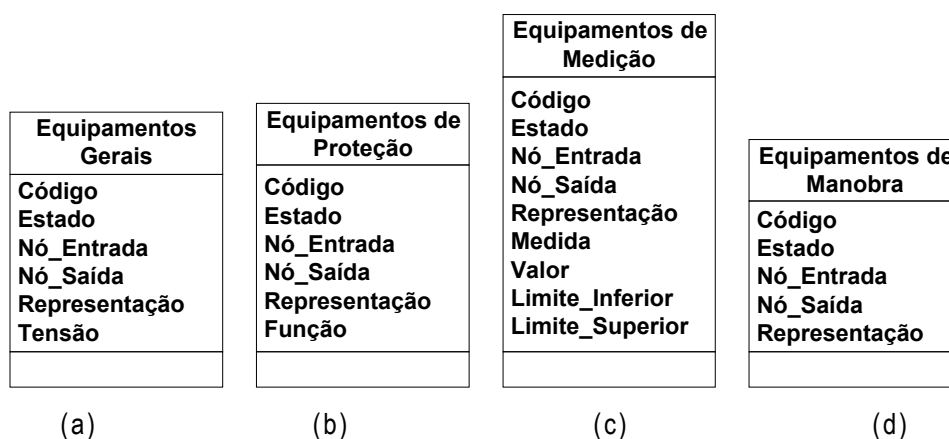


Figura 4.3 – Classes: (a) “Equipamentos Gerais”, (b) “Equipamentos de Proteção”, (c) “Equipamentos de Medição” e (d) “Equipamentos de Manobra”

Como os equipamentos de medição usam transformadores de potencial e de corrente como interface com o sistema elétrico seu nó de entrada é igual ao nó de saída.

Da classe “Equipamentos Gerais” pode-se derivar as seguintes classes: “Geradores”, “Transformadores”, “Linhas”, “Barras”, “Alimentadores” e “Capacitores”, Figura 4.4. As quais possuem as seguintes características adicionais:

- Geradores:
 - Capacidade: Capacidade nominal em MVA do gerador.
- Transformadores:
 - Nó_Saída_2: Código do nó de saída do terciário (se houver);
 - Capacidade: Capacidade nominal em MVA do transformador;
 - Tensão_2: Classe de tensão do secundário;
 - Tensão_3: Classe de tensão do terciário (se houver);
 - Enrolamento_1: Tipo do enrolamento primário;
 - Enrolamento_2: Tipo do enrolamento secundário;
 - Enrolamento_3: Tipo do enrolamento terciário (se houver).
- Linhas:
 - Cabo: Código do cabo utilizado no trecho de linha;
 - R: Resistência do cabo em Ω/km ;
 - X: Reatância do cabo em Ω/km ;
 - Corrente: Corrente conduzida pela linha em A;
 - Capacidade: Capacidade de condução de corrente em A;
 - Distância: Tamanho do trecho de linha em km.
- Barras:
 - Entradas: Número de ligações na entrada;
 - Saídas: Número de ligações na saída;
 - Prioridade: Prioridade de energização;
 - Capacidade: Capacidade de condução de corrente em A;
 - Carga_P: Carga em potência ativa – kW;
 - Carga_Q: Carga em potência reativa – kVAR;

- OutLinhas: Coleção de linhas conectadas a barra pelo nó de entrada;
- InLinhas: Coleção de linhas conectadas a barra pelo nó de saída.
- Alimentadores:
 - Prioridade: Prioridade de energização.
- Capacitores:
 - Capacidade: Capacidade nominal em MVar.



Figura 4.4 – Classes: (a) “Geradores”, (b) “Transformadores”, (c) “Linhas”, (d) “Barras”, (e) “Alimentadores” e (f) “Capacitores”

Da classe “Equipamentos de Manobra” pode-se derivar as seguintes classes: “Disjuntores”, “Chaves”, “Chaves By-Pass” e “Chaves de Aterramento”, Figura 4.5. As quais possuem as seguintes características adicionais:

- Disjuntores:
 - Capacidade: Capacidade nominal em kA do disjuntor.
- Chaves:
 - Tipo: Tipo da chave: NA, NF, de socorro externo e de socorro interno.

Disjuntores	Chaves	Chaves By-Pass	Chaves de Aterramento
Código	Código	Código	Código
Estado	Estado	Estado	Estado
Nó_Entrada	Nó_Entrada	Nó_Entrada	Nó_Entrada
Nó_Saída	Nó_Saída	Nó_Saída	Nó_Saída
Representação	Representação	Representação	Representação
Capacidade	Tipo		

(a) (b) (c) (d)

Figura 4.5 – Classes: (a) “Disjuntores”, (b) “Chaves”, (c) “Chaves By-Pass” e (d) “Chaves de Aterramento”

As classes “Equipamentos de Proteção” e “Equipamentos de Medição” não têm filhos, e englobam os objetos responsáveis por medir, monitorar e proteger o sistema. A classe “Equipamentos de Proteção” representa os relés que monitoram e protegem o SEP. Já a classe “Equipamentos de Medição” representa os voltímetros, amperímetros, e demais medidores.

A relação de Agregação permite a introdução do sentido parte-todo, permitindo que as classes do sistema se associem com outras e formem uma nova classe que representa o conjunto de todas as outras. Assim é formada a classe “Subestação” que representa o agrupamento de: “Transformadores”, “Linhas”, “Barras”, “Alimentadores”, “Capacitores”, “Equipamentos de Proteção”, “Equipamentos de Medição”, “Disjuntores”, “Chaves”, “Chaves By-Pass” e “Chaves de Aterramento”. É claro que existem mais equipamentos em uma subestação elétrica, mas eles não foram considerados por não serem prioritários para operação da mesma.

A classe “Distribuição” é formada pelo agrupamento de: “Transformadores”, “Linhas”, “Barras”, “Alimentadores”, “Equipamentos de Medição” e “Chaves”. Da mesma forma alguns equipamentos foram ignorados, porém podem ser adicionados facilmente ao modelo caso algum aplicativo necessite.

Já a classe “SEP” é formada pelo agrupamento de: “Geradores”, “Linhas”, “Barras”, “Subestação” e “Distribuição”. É importante ressaltar que novas agregações podem ser feitas e novas classes podem ser inseridas facilmente no modelo orientado a objetos, permitindo a integração de novos aplicativos que necessitem de dados que ainda não estão disponíveis nessa modelagem.

4.2 Implementação da Arquitetura Proposta para Restabelecimento de Subestações Elétricas

O restabelecimento da configuração normal de operação de uma Subestação Elétrica (SE), após uma contingência, é feito atualmente através da intervenção do operador e segue critérios operacionais pré-estabelecidos por estudos de engenharia. As linhas mestras do restabelecimento incluem o raciocínio de identificação da necessidade de ações de restabelecimento, de validação das medições, de diagnóstico e de estruturação do plano de manobras de restabelecimento.

A concepção e o desenvolvimento de um sistema inteligente capaz de restabelecer uma SE passa necessariamente pela análise e avaliação dos paradigmas anteriormente estabelecidos e pela definição da estratégia de resolução do problema.

O objetivo perseguido pelo sistema inteligente é o de restabelecer o maior número possível de componentes da SE. A partir da análise dos dados de entrada, identificam-se as causas do problema, os componentes envolvidos e os afetados. A seguir, após uma análise criteriosa baseada no conhecimento e uma avaliação do conjunto de condições associadas à topologia e às condições operativas da SE, identificam-se os componentes que devem e os que não devem

participar do restabelecimento, definindo quando e como deve ser a sua reenergização, dentro da seqüência de restabelecimento. Dessa forma, definem-se a melhor configuração e a seqüência de chaveamento para se atingir a condição operativa ótima pós-contingência.

O sistema multi-agente desenvolvido para controle e supervisão de subestações elétricas está apresentado na Figura 4.6, os detalhes de sua implementação computacional podem ser vistos no Anexo C.

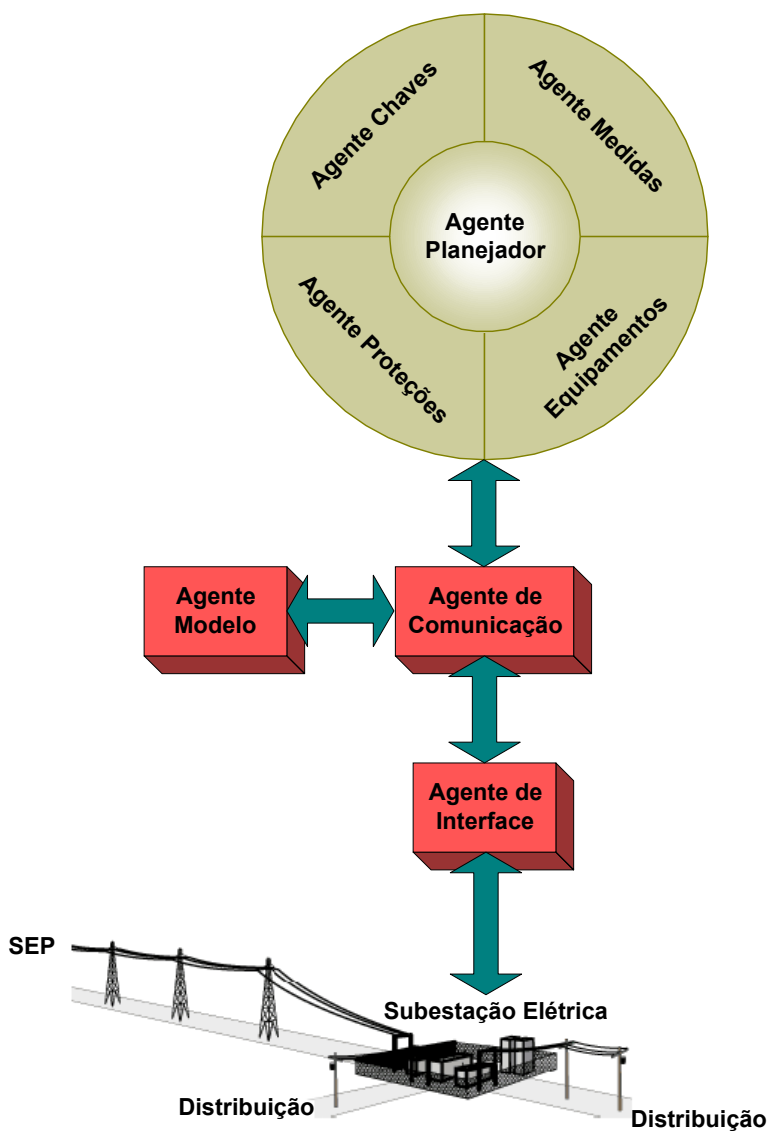


Figura 4.6 – Sistema multi-agente para subestações elétricas

Esta aplicação desenvolvida a partir da arquitetura proposta, tem como objetivo auxiliar o restabelecimento de uma SE através da inferência de um plano de manobras.

Esse plano de manobras utiliza uma visão constante e completa do estado operacional dos componentes e dos equipamentos que a constituem, dispondo de meios para avaliar o agente da ocorrência, delimitar a área defeituosa isolada pela atuação da proteção, caracterizar se o defeito é permanente ou transitório e identificar os componentes envolvidos e os afetados.

O SMA gera uma lista de ações necessárias e suficientes para normalizar operativamente a SE após desligamentos parciais ou totais de seus componentes, reintegrando-a, de forma estabilizada, novamente ao sistema elétrico.

4.2.1 Agentes Especializados

O sistema multi-agente desenvolvido para restabelecimento de subestações elétricas está apresentado na Figura 4.6. Este modelo contém um pacote de agentes especializados em subestações composto por: “Agente Medidas”, “Agente Equipamentos”, “Agente Chaves”, “Agente Proteções” e “Agente Planejador”.

4.2.1.1 Agente Medidas

O “Agente Medidas” possui um algoritmo de monitoramento dos sinais de interesse, como a tensão e a corrente verificadas pelos medidores presentes na planta, buscando detectar possibilidades de defeito ou oscilações que afetem a estabilidade do sistema.

De acordo com os valores encontrados, o agente sugere ações que devolvam o equilíbrio ao sistema, evitando danos aos equipamentos e, em caso de falta, auxiliando no processo de elaboração do plano de restabelecimento.

Se a medição for de tensão há duas possibilidades de lógica de atuação de acordo com a classe do equipamento:

- Linhas: caso não haja tensão deve-se fechar o disjuntor da linha, ou então ajustar o TAP do transformador de acordo com o faixa de valor da medida.
- Barras: se a medida estiver dentro da faixa admitida e algum disjuntor de saída estiver aberto deve-se reenergizar o mesmo, ou então ajustar o TAP do transformador ou chavear o banco de capacitores de acordo com a faixa de valor da medida.

Foi utilizado um conjunto de regras de produção do tipo SE-ENTÃO no agente que é empregado para cada medidor existente no modelo orientado a objeto da subestação, como mostrado abaixo:

```

Medidor ← Medidor em análise
DJ1 ← (Buscar disjuntor ligado ao Medidor)
Equip ← (Buscar equipamento ligado ao DJ1)
SE Medida do Medidor = Tensão ENTÃO
  SE Classe do Equip = Linhas ENTÃO
    Equip_Aux ← (Buscar equipamento ligado ao Medidor)
    SE Valor do Medidor ≤ Limite_Inferior do Medidor E Valor _
do Medidor ≠ 0 ENTÃO
      Enviar sugestão: "Tensão na linha Equip abaixo da faixa _
admitida. Aumentar TAP do transformador Equip_Aux"
    SENÃO SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
      Enviar sugestão: "Tensão na linha Equip acima da faixa _
admitida. Diminuir TAP do transformador Equip_Aux"
    FIM SE
  FIM SE
  SE Classe do Equip = Barras ENTÃO
    PARA i = 1 até Número de saídas do Equip
      DJ[i] ← (Buscar disjuntor ligado ao Equip)
      Equip_Aux ← (Buscar equipamento ligado ao DJ[i])
      SE Classe do Equip_Aux = Capacitores THEN DJ2 ← DJ[i]
      i ← i + 1
    FIM PARA
    DJ3 ← (Buscar disjuntor ligado ao Equip e a um equipamento _
da Classe = Barras)
    SE Valor do Medidor ≤ Limite_Inferior do Medidor E Valor do _
Medidor ≠ 0 ENTÃO
      SE Estado de DJ2 = 1 ENTÃO
        Enviar sugestão: "Tensão na barra Equip abaixo da _
faixa admitida. Aumentar TAP do transformador Equip_Aux"
      SENÃO
        Enviar sugestão: "Tensão na barra Equip abaixo da _
faixa admitida. Aumentar TAP do transformador Equip_Aux _
ou fechar disjuntor DJ2"
      FIM SE
    SENÃO SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
      SE Estado de DJ2 = 0 ENTÃO

```

```

    Enviar sugestão: "Tensão na barra Equip acima da faixa _
    admitida. Diminuir TAP do transformador Equip_Aux"
SENÃO
    Enviar sugestão: "Tensão na barra Equip acima da faixa _
    admitida. Diminuir TAP do transformador Equip_Aux ou _
    abrir disjuntor DJ2"
FIM SE
SENÃO SE Valor do Medidor = 0 ENTÃO
    PARA i = 1 até Número de saídas do Equip
        SE Estado de DJ[i] = 1 ENTÃO Enviar sugestão: "Barra _
        desenergizada. Abrir disjuntor DJ[i]"
        i ← i + 1
    FIM PARA
SENÃO SE Estado de DJ3 = 0 ENTÃO
    PARA i = 1 até Número de saídas do Equip
        SE Estado de DJ[i] = 0 ENTÃO Enviar sugestão: "Tensão _
        na barra normal. Fechar disjuntor DJ[i]"
        i ← i + 1
    FIM PARA
SENÃO SE Estado de DJ3 = 1 ENTÃO
    Equip_Aux1 ← (Buscar barra ligada ao DJ3 e diferente de _
    Equip)
    Equip_Aux2 ← (Buscar disjuntor ligado ao Nó_Entrada de _
    Equip_Aux1)
    Equip_Aux3 ← (Buscar transformador ligado ao Equip_Aux2)
    Medidor1 ← (Buscar medidor de corrente ligado ao Equip_Aux1)
    PARA i = 1 até Número de saídas do Equip
        Medidor2 ← (Buscar medidor de corrente ligado ao _
        Alimentador[i])
        Corrente ← Valor do Medidor1 + Valor do Medidor2 antes _
        da falha
        Potência ←  $\sqrt{3}$  * Valor do Medidor * Corrente
        SE Estado de DJ[i] = 0 E (Corrente < Capacidade de _
        Equip_Aux OU Potência < Capacidade do Equip_Aux3) ENTÃO
            Enviar sugestão: "Tensão na barra normal. Fechar _
            disjuntor DJ[i]"
            i ← i + 1
        SENÃO
            i ← Número de saídas do Equip
    FIM SE
FIM PARA
FIM SE
FIM SE
SENÃO SE Medida do Medidor = Corrente ENTÃO
    SE Classe do Equip = Linhas ENTÃO
        SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
            Enviar sugestão: "Corrente na linha Equip acima da faixa _
            admitida. Abrir disjuntor DJ1"
        FIM SE
    FIM SE
SE Classe do Equip = Barras ENTÃO
        SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
            Enviar sugestão: "Corrente na barra Equip acima da faixa _
            admitida. Abrir disjuntor DJ1"
        FIM SE
    FIM SE
FIM SE

```



```

Equip_Aux ← (Buscar equipamento ligado ao Medidor)
SE Classe do Equip_Aux = Capacitores ENTÃO
    SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
        Enviar sugestão: "Corrente no banco de capacitores Equip _
        acima da faixa admitida. Abrir disjuntor DJ1"
    FIM SE
FIM SE
SE Classe do Equip = Alimentadores ENTÃO
    SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
        Enviar sugestão: "Corrente no alimentador Equip acima da _
        faixa admitida. Abrir disjuntor DJ1"
    FIM SE
FIM SE
FIM SE

```

4.2.1.2 Agente Equipamentos

O "Agente Equipamentos" encapsula um programa de restauração que age em função do equipamento afetado de acordo com uma filosofia definida. Esse programa verifica as condições dos transformadores, barramentos, linhas em busca de defeitos nesses equipamentos, essa busca é feita através de consulta ao estado de alguns relés de proteção específicos.

No caso dos transformadores se houver ausência de defeito interno nos equipamentos, o agente solicita a reenergização dos mesmos. Agora, caso seja encontrado algum defeito interno, o agente bloqueia o processo de restauração e sugere o acionamento da equipe de manutenção.

Para a barra, caso a proteção diferencial tenha atuado deve-se desligar o disjuntor entre as barras e todos os disjuntores ligados à barra. Se for outra proteção pode-se tentar o restabelecimento solicitando o fechamento dos disjuntores abertos.

A análise para as linhas leva em consideração a proteção de sobrecorrente, caso esta tenha atuado as ações de restabelecimento devem ser impedidas, já se for outro tipo de proteção pode-se tentar reenergizar a linha.

Foi utilizado um conjunto de regras de produção do tipo SE-ENTÃO no agente que é empregado para cada equipamento existente no modelo orientado a objeto da subestação, como mostrado abaixo:

```

Equip ← Equipamento em análise
SE Classe do Equip = Transformadores ENTÃO
    Rele1 ← (Buscar estado da proteção 63 do Equip)
    Rele2 ← (Buscar estado da proteção 63C do Equip)
    Rele3 ← (Buscar estado da proteção 63VS do Equip)
    DJ1 ← (Buscar disjuntor ligado ao Equip)
    DJ2 ← (Buscar disjuntor ligado ao Equip)
    SE Nó_Saída2 do Equip ≠ NULL ENTÃO
        DJ3 ← (Buscar disjuntor ligado Equip)
    FIM SE
    SE Rele1 = 1 OU Rele2 = 1 OU Rele3 = 1 ENTÃO
        SE Nó_Saída2 ≠ NULL ENTÃO
            Enviar sugestão: "Defeito interno no transformador Equip.
            Chamar equipe de manutenção. Abrir disjuntor DJ1. Abrir
            disjuntor DJ2. Abrir disjuntor DJ3"
        SENÃO
            Enviar sugestão: "Defeito interno no transformador Equip.
            Chamar equipe de manutenção. Abrir disjuntor DJ1. Abrir
            disjuntor DJ2"
        FIM SE
    SENÃO
        SE Nó_Saída2 ≠ NULL ENTÃO
            Enviar sugestão: "Não há defeito interno no transformador
            Equip. Fechar disjuntor DJ1. Fechar disjuntor DJ2. Fechar
            disjuntor DJ3"
        SENÃO
            Enviar sugestão: "Não há defeito interno no transformador
            Equip. Fechar disjuntor DJ1. Fechar disjuntor DJ2"
        FIM SE
    FIM SE
SENÃO SE Classe do Equip = Barras ENTÃO
    Rele1 ← (Buscar estado da proteção 87 do Equip)
    SE Rele1 = 1 ENTÃO
        PARA i = 1 até Número de Entradas do Equip
            DJ[i] ← (Buscar disjuntor ligado ao Equip)
            Enviar sugestão: "Existe corrente diferencial na barra
            Equip. Abrir disjuntor DJ[i]"
            i ← i + 1
        FIM PARA
        PARA i = 1 até Número de Saídas do Equip
            DJ[i] ← (Buscar disjuntor ligado ao Equip)
            Enviar sugestão: "Existe corrente diferencial na barra
            Equip. Abrir disjuntor DJ[i]"
            i ← i + 1
        FIM PARA
    SENÃO
        PARA i = 1 até Número de entradas do Equip
            DJ[i] ← (Buscar disjuntor ligado ao Equip)
            SE Estado de DJ[i] = 0 ENTÃO Enviar sugestão: "Não há
            corrente diferencial na barra Equip. Fechar disjuntor DJ[i]"
            i ← i + 1
        FIM PARA
        PARA i = 1 até Número de saídas do Equip
            DJ[i] ← (Buscar disjuntor ligado ao Equip)
            SE Estado de DJ[i] = 0 ENTÃO Enviar sugestão: "Não há
            corrente diferencial na barra Equip. Fechar disjuntor DJ[i]"

```

```

        i ← i + 1
    FIM PARA
  FIM SE
SENÃO SE Classe do Equip = Linhas ENTÃO
  Rele1 ← (Buscar estado da proteção 50/51 do Equip)
  DJ1 ← (Buscar disjuntor ligado ao Equip)
  SE Rele1 = 1 ENTÃO
    Enviar sugestão: "Existe sobrecorrente na linha Equip. Abrir _
    disjuntor DJ1"
  SENÃO
    Enviar sugestão: "Não há sobrecorrente na linha Equip. Fechar _
    disjuntor DJ1"
  FIM SE
FIM SE

```

4.2.1.3 Agente Chaves

O "Agente Chaves" possui experiência na filosofia de restauração em função das chaves atuadas e de sua localização no SEP, sendo que de acordo com essa base de conhecimento ele define as ações a serem executadas nas chaves do sistema.

Nesse caso, implementou-se um algoritmo de análise para os disjuntores da SE, os quais segundo sua funcionalidade foram divididos em três grupos:

- Disjuntores de barra para barra: deverão fechar quando uma de suas barras estiver energizada e a outra não, e ainda, não deve haver sugestões para fechar o disjuntor de entrada da barra desenergizada.
- Disjuntores dos bancos de capacitores: deverão fechar quando a tensão da barra for menor ou igual a um valor mínimo definido pelo operador, e ainda, deverão abrir quando a tensão da barra for maior ou igual a um valor máximo também definido pelo operador.
- Disjuntores dos alimentadores, transformadores e linhas: deverão fechar quando houver energia em um dos seus dois lados.

Foi utilizado um conjunto de regras de produção do tipo SE-ENTÃO no agente que é empregado para cada chave existente no modelo orientado a objeto da subestação, como mostrado abaixo:

```

DJ ← Disjuntor em análise
Equip1 ← (Buscar equipamento ligado ao Nó_Entrada de DJ)
Equip2 ← (Buscar equipamento ligado ao Nó_Saída de DJ)
SE Classe do Equip1 = Barras E Classe do Equip2 = Barras ENTÃO
  SE Estado do Equip1 = 1 E Estado do Equip2 = 0 ENTÃO
    DJ_Aux ← (Buscar disjuntor ligado ao Equip2)
  SENÃO SE Estado do Equip1 = 0 E Estado do Equip2 = 1 ENTÃO
    DJ_Aux ← (Buscar disjuntor ligado ao Equip1)
  FIM SE
  SE Estado atual de DJ_Aux = 0 E Houve sugestões para fechar _
  DJ_Aux no passo anterior ENTÃO
    Enviar sugestão: "Barra Equip1 desenergizada. Fechar _
    Disjuntor entre as barras DJ"
  FIM SE
SENÃO SE Classe do Equip1 = Barras E Classe do Equip2 = _
  Capacitores ENTÃO
  DJ_Aux ← (Buscar disjuntor ligado ao Nó_Entrada do Equip1)
  Medidor ← (Buscar medidor de tensão ligado ao DJ_Aux)
  SE Valor do Medidor ≤ Limite_Inferior do Medidor ENTÃO
    Enviar sugestão: "Tensão na barra Barra abaixo da faixa _
    admitida. Fechar disjuntor DJ"
  SENÃO SE Valor do Medidor ≥ Limite_Superior do Medidor ENTÃO
    Enviar sugestão: "Tensão na barra Barra acima da faixa _
    admitida. Abrir disjuntor DJ"
  FIM SE
SENÃO
  SE Estado de DJ = 0 ENTÃO
    SE Estado do Equip1 = 1 OU Estado do Equip2 = 1 ENTÃO
      Enviar sugestão: "Disjuntor DJ aberto com um terminal _
      energizado. Fechar disjuntor DJ"
    FIM SE
  FIM SE
FIM SE

```

4.2.1.4 Agente Proteções

Quando um distúrbio ocorre no SEP, muitos alarmes e relés de proteção são ativados e percebidos pelo "Agente Proteções", o qual possui informação sobre as lógicas de atuação dos relés, bem como sobre o ajuste dos mesmos. Uma vez que um relé tenha atuado, o "Agente Proteções" verifica as ligações deste e a lógica referente ao tipo do componente, após consultar sua base de dados ele elabora e envia ao "Agente Planejador" uma sugestão de ações para a correção do problema.

Foi utilizado um conjunto de regras de produção do tipo SE-ENTÃO no agente que é empregado para cada proteção existente no modelo orientado a objeto da subestação, como mostrado abaixo:

```

Rele ← Proteção em análise
SE (Função do Rele = 50/51 OU Função do Rele = 51) E Estado do _
Rele = 1 ENTÃO
    Medidor ← (Buscar medição de corrente ligada ao Rele)
    DJ1 ← (Buscar disjuntor ligada ao Rele)
    SE Valor do Medidor > Limite_Superior do Medidor ENTÃO
        Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ1"
    SENÃO
        Enviar sugestão: "Proteção Rele é incoerente com valor da _
        medição Medidor. Fechar disjuntor DJ1"
    FIM SE
SENÃO SE (Função do Rele = 50/51N OU Função do Rele = 51N) E Estado _
do Rele = 1 ENTÃO
    Medidor ← (Buscar medição de corrente de neutro ligada ao Rele)
    DJ1 ← (Buscar disjuntor ligado ao Rele)
    DJ2 ← (Buscar disjuntor ligado ao Rele)
    SE Valor do Medidor > Limite_Superior do Medidor ENTÃO
        SE DJ1 = NULL ENTÃO
            Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ2"
        SENÃO
            Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ1"
        FIM SE
    SENÃO
        SE DJ1 = NULL ENTÃO
            Enviar sugestão: "Proteção Rele é incoerente com valor da _
            medição Medidor. Fechar disjuntor DJ2"
        SENÃO
            Enviar sugestão: "Proteção Rele é incoerente com valor da _
            medição Medidor. Fechar disjuntor DJ1"
        FIM SE
    FIM SE
SENÃO SE Função do Rele = 27 E Estado do Rele = 1 ENTÃO
    Medidor ← (Buscar medição de tensão ligada ao Rele)
    DJ1 ← (Buscar disjuntor ligado ao Nó_Entrada do Rele)
    DJ2 ← (Buscar disjuntor ligado ao Nó_Saída do Rele)
    SE Valor do Medidor < Limite_Inferior do Medidor ENTÃO
        SE DJ1 = NULL ENTÃO
            Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ2"
        SENÃO
            Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ1"
        FIM SE
    SENÃO
        SE DJ1 = NULL ENTÃO
            Enviar sugestão: "Proteção Rele é incoerente com valor da _
            medição Medidor. Fechar disjuntor DJ2"
        SENÃO
            Enviar sugestão: "Proteção Rele é incoerente com valor da _
            medição Medidor. Fechar disjuntor DJ1"
        FIM SE
    FIM SE
SENÃO SE (Função do Rele = 63 OU Função do Rele = 63C OU _

```

```

Função do Rele = VS) E Estado do Rele = 1 ENTÃO
  DJ1 ← (Buscar disjuntor ligado ao Rele)
  Equip ← (Buscar equipamento ligado ao Rele)
  DJ2 ← (Buscar disjuntor ligado ao Nó_Saída do Equip)
  DJ3 ← (Buscar disjuntor ligado ao Nó_Saída2 do Equip)
SE DJ3 = NULL ENTÃO
  Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ1. _
  Abrir disjuntor DJ2."
SENÃO
  Enviar sugestão: "Proteção Rele atuou. Abrir disjuntor DJ1. _
  Abrir disjuntor DJ2. Abrir disjuntor DJ3"
FIM SE
SENÃO SE Função do Rele = 79 E Estado do Rele = 1 ENTÃO
  DJ1 ← (Buscar disjuntor ligado ao Rele)
  Equip ← (Buscar equipamento ligado ao DJ)
SE Estado do Equip = 1 ENTÃO
  SE Contador_Rele ≠ NULL E Contador_Rele < 3 ENTÃO
    Enviar sugestão: "Proteção Rele atuou. Fechar disjuntor DJ1"
    Contador_Rele = Contador_Rele + 1
  SENÃO
    Contador_Rele = 0
  FIM SE
FIM SE
FIM SE

```

4.2.1.5 Agente Planejador

O "Agente Planejador" é responsável por determinar quais ações devem ser implementadas para solucionar as tarefas operativas necessárias. Este agente contém um algoritmo de planejamento que analisa a informação de todos os agentes provendo um plano ordenado de ações para o usuário.

Primeiramente, é montada uma lista de sugestões para cada agente, Figura 4.7. Nesse algoritmo de classificação das sugestões por agente é atribuído um peso, onde o mais importante é o "Agente Proteções", em seguida o "Agente Equipamentos", em terceiro lugar o "Agente Medidas" e o menos importante é o "Agente Chaves".

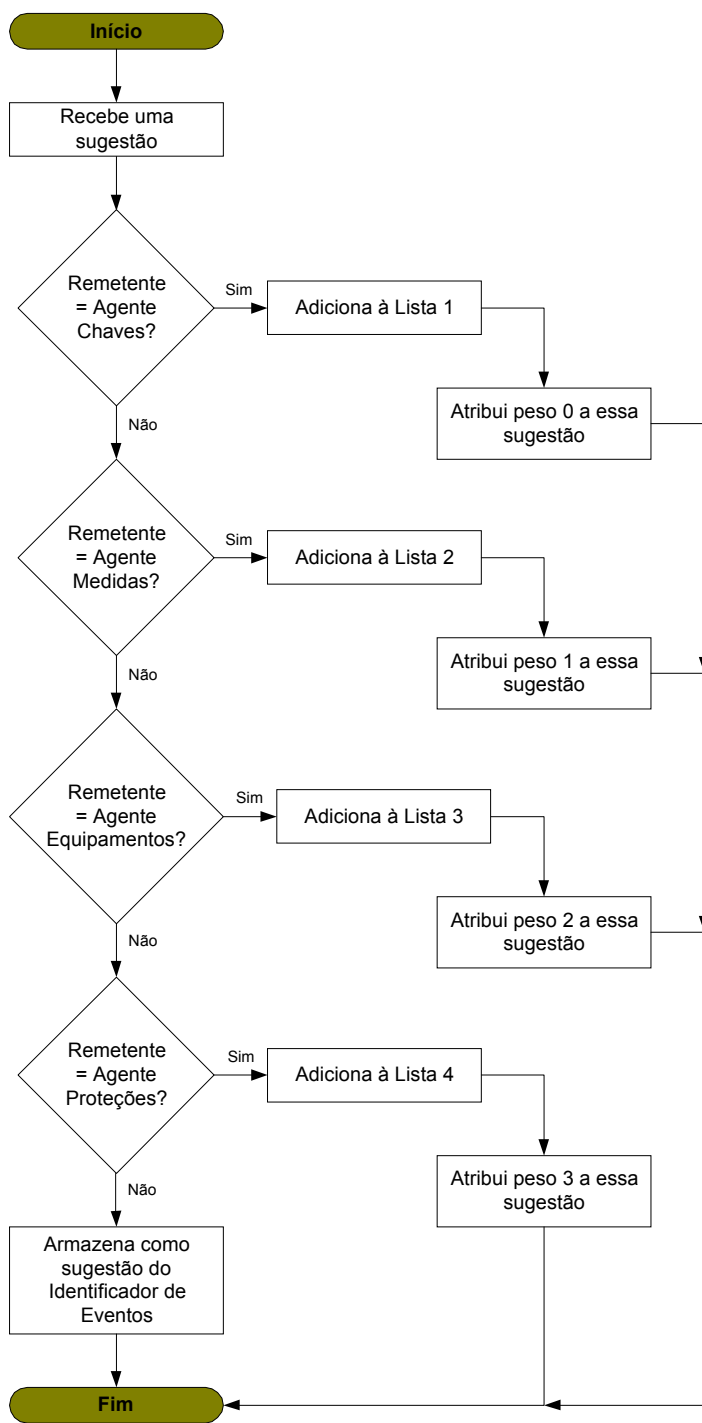


Figura 4.7 – Algoritmo de classificação por agente das sugestões

Quando todos os agentes enviaram suas sugestões, são eliminadas as sugestões iguais que estejam contidas em listas diferentes, de acordo com o seu peso, Figura 4.8.

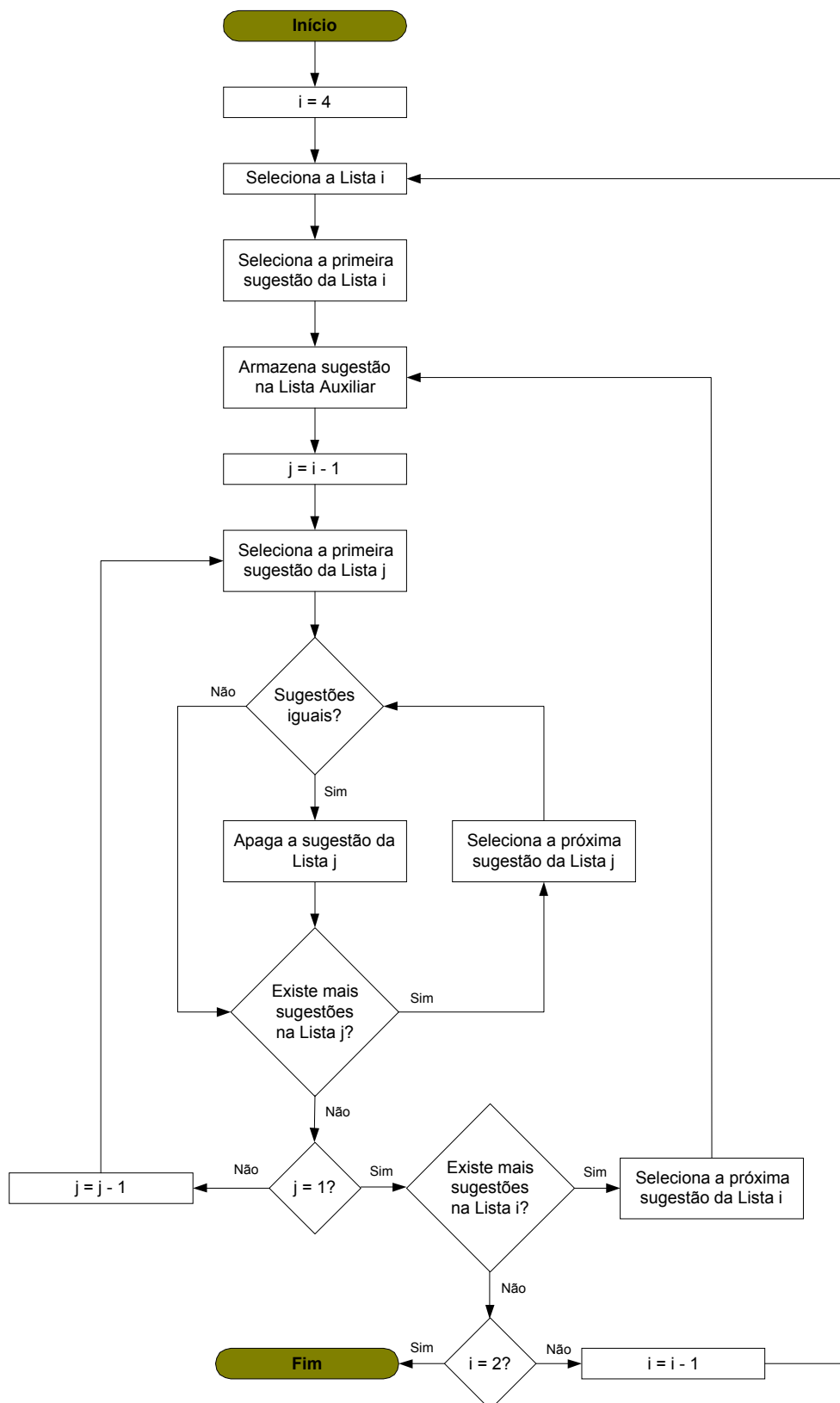


Figura 4.8 – Algoritmo de eliminação das sugestões iguais em listas diferentes

Para isso o algoritmo inicia com a lista de sugestões do “Agente Proteções”, a qual possui o maior peso, e compara cada uma das suas sugestões com as demais sugestões contidas nas outras listas, sempre seguindo a ordem dos pesos dos agentes. Todas as sugestões que passarem por essa etapa vão para uma lista auxiliar.

De posse da lista auxiliar, o “Agente Planejador” vai identificar se ainda há sugestões iguais nessa lista ou analisar quando houver sugestões antagônicas, a fim de eliminá-las e montar a lista final de sugestões, Figura 4.9.

Uma vez que a lista final contém apenas as sugestões válidas é necessário ordená-las, para isso é utilizada a tabela de hierarquia de restabelecimento mostrada abaixo:

Tabela 4.1 – Hierarquia de restabelecimento

INÍCIO	
Fontes	LTs
Transformação	Transformadores e Autotransformadores
Carga	LTs e Alimentadores
Controle	Banco de Capacitores, Reatores e Síncronos
FIM	

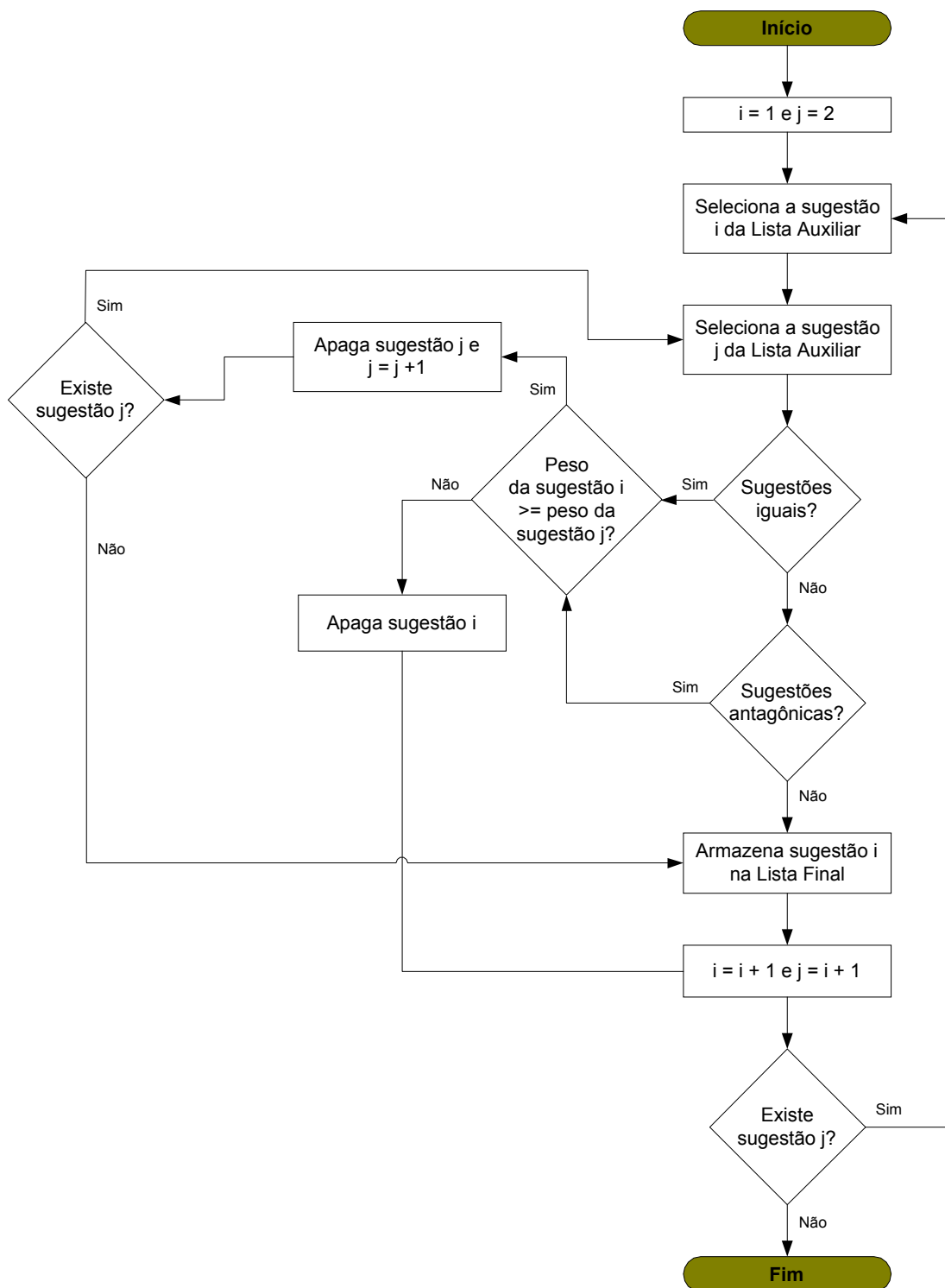


Figura 4.9 – Algoritmo de eliminação das sugestões iguais e diferentes na lista auxiliar

4.2.2 Diagrama de Interação entre Agentes

O diagrama de interação do modelo de SMA para aplicação em subestações elétricas é apresentado na Figura 4.10.

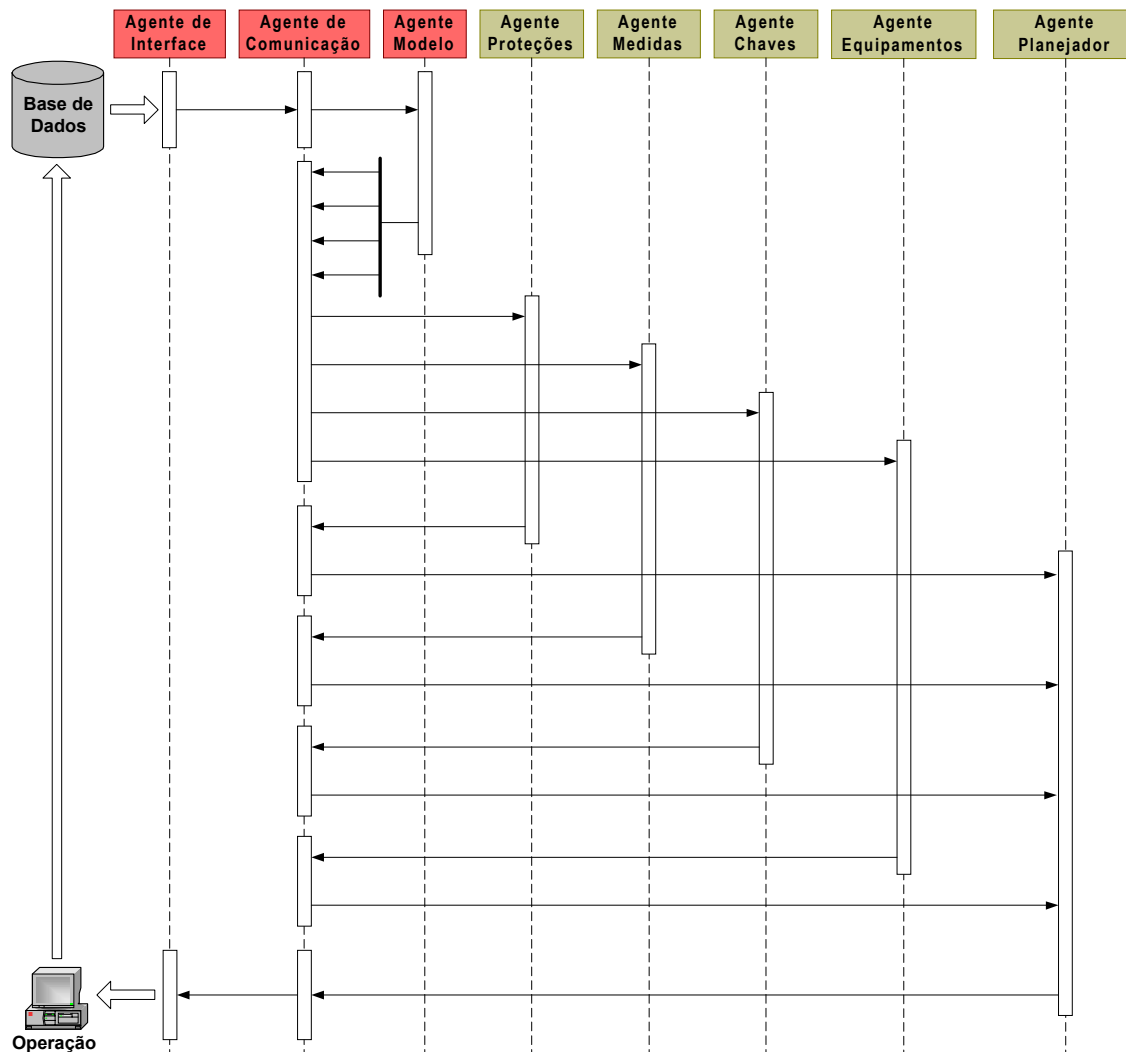


Figura 4.10 – Diagrama de interação do SMA para subestações elétricas

Neste diagrama são explicitadas todas as comunicações existentes entre os agentes. O ciclo de análise inicia com o “Agente de Interface” que ao perceber uma alteração na base de dados do sistema SCADA faz uma coleta dos dados, interpreta-os e informa o “Agente Modelo” via “Agente de Comunicação”.

O “Agente Modelo” então atualiza o modelo orientado a objetos e envia simultaneamente os novos dados para os agentes “Proteções”, “Medidas”,

“Chaves” e “Equipamentos”, os quais são responsáveis pelas análises dos mesmos e pela sugestão de ações de operação.

O processamento desses agentes se dá de forma paralela, ou seja, todos os agentes analisam seus dados ao mesmo tempo, otimizando o uso dos recursos computacionais empregados no seu desenvolvimento. Todas as sugestões de ações de operação providas dos agentes “Proteções”, “Medidas”, “Chaves” e “Equipamentos” são enviadas para o “Agente Planejador” via “Agente de Comunicação” de forma assíncrona.

Quando o “Agente Planejador” recebe todas as ações sugeridas por todos os agentes, ele organiza um plano de ações de operação, ordenando-as e eliminando ambigüidades.

Esse plano é então enviado para o “Agente de Interface”, via “Agente de Comunicações”, o qual irá apresentá-lo para o operador, ficando a cargo do mesmo aceitar ou não as ações sugeridas e implementá-las.

4.3 Implementação da Arquitetura Proposta para Restabelecimento de Sistemas de Distribuição

O sistema multi-agente desenvolvido para restabelecimento de sistemas de distribuição está apresentado na Figura 4.11, os detalhes de sua implementação computacional podem ser vistos no Anexo D.

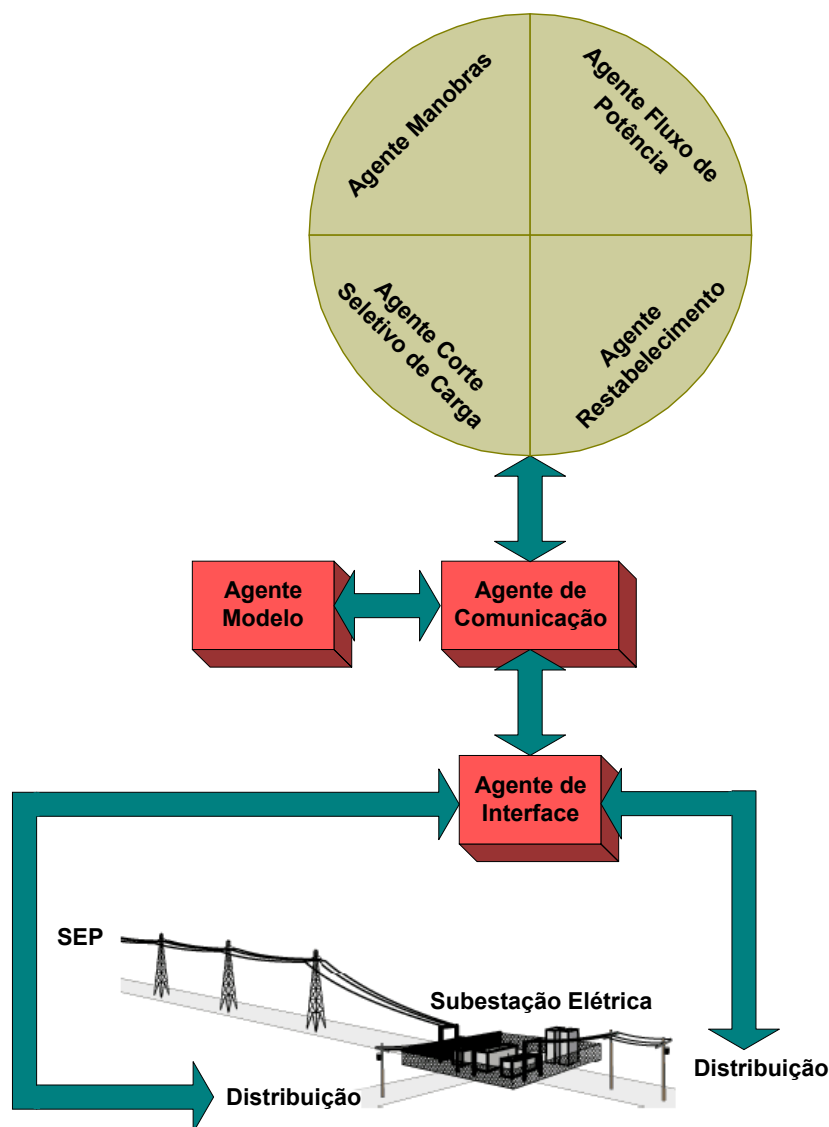


Figura 4.11 – Modelo multi-agente para sistemas de distribuição

O objetivo do restabelecimento é recompor um sistema de distribuição radial de forma eficiente e rápida, reintegrando-o ao sistema elétrico o mais próximo possível da configuração em que se encontrava antes da perturbação, excluindo apenas os componentes defeituosos ou afetados por faltas permanentes, e aqueles que poderiam levar novamente o sistema de distribuição a uma nova contingência.

As influências nas condições operativas do sistema de distribuição ao longo do processo de recomposição de um sistema radial devem ser avaliadas a cada manobra de restabelecimento de forma a validar o que já foi feito e liberar a execução das ações subseqüentes.

4.3.1 Agentes Especializados

O sistema multi-agente desenvolvido para restabelecimento de sistemas de distribuição contém um pacote de agentes especializados em sistemas de distribuição composto por: “Agente Manobras”, “Agente Corte Seletivo de Carga”, “Agente Fluxo de Carga” e o “Agente Restabelecimento”; os quais são responsáveis pelo raciocínio das ações a serem sugeridas ao operador de acordo com a especialidade de cada um.

4.3.1.1 Agente Restabelecimento

O “Agente Restabelecimento” é o responsável por analisar todas alternativas de manobras, bem como suas conseqüências, classificá-las e então apresentar para o operador um plano para restabelecer o sistema de distribuição. O algoritmo de funcionamento do agente está apresentado na Figura 4.12.

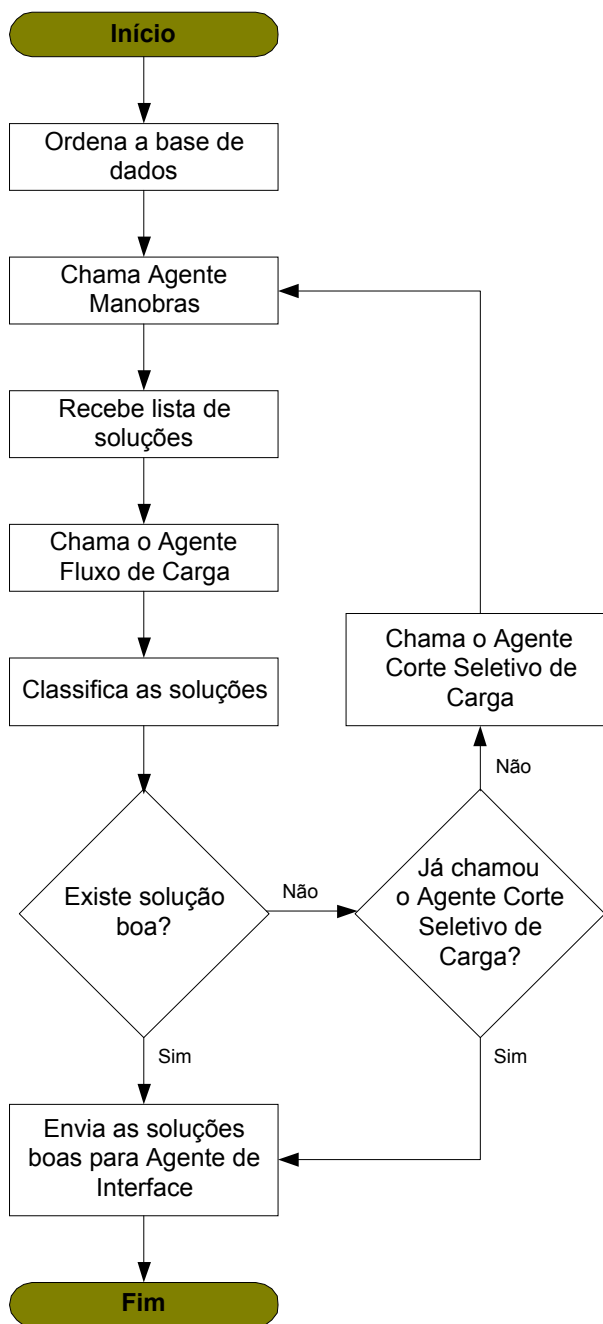


Figura 4.12 – Algoritmo de funcionamento do “Agente Restabelecimento”

O procedimento de ordenação da base de dados do sistema de distribuição em análise consiste do preenchimento da lista de linhas de saída (OutLinhas) e da lista de linhas de entrada (InLinhas) de cada uma das barras. Uma linha de saída é aquela que está conectada a barra pelo seu nó de entrada, já a linha de entrada estará conectada a barra pelo seu nó de saída, Figura 4.13.

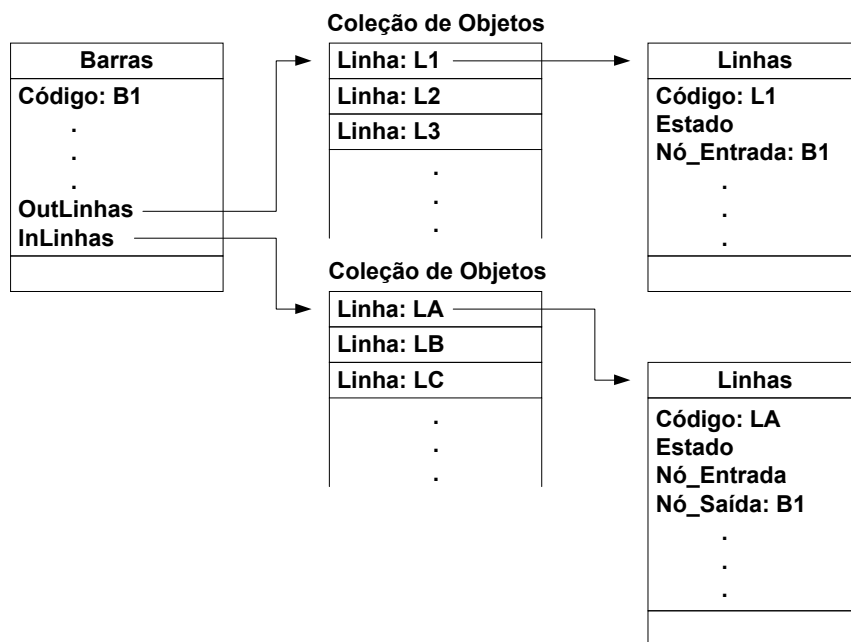


Figura 4.13 – Ordenação da base de dados do sistema de distribuição

O procedimento de classificação das soluções utiliza o conjunto de regras abaixo, bem como as funções difusas apresentadas na Figura 4.14:

SE Manobras = Pouca E Sobrecarga = Leve E Sobretensão = Leve E Subtensão = Leve ENTÃO Nota = Boa
SE Manobras = Pouca E Sobrecarga = Média E Sobretensão = Leve E Subtensão = Leve ENTÃO Nota = Boa
SE Manobras = Pouca E Sobrecarga = Leve E Sobretensão = Média E Subtensão = Leve ENTÃO Nota = Boa
SE Manobras = Pouca E Sobrecarga = Leve E Sobretensão = Leve E Subtensão = Média ENTÃO Nota = Boa
SE Manobras = Pouca E Sobrecarga = Média E Sobretensão = Média E Subtensão = Leve ENTÃO Nota = Média
SE Manobras = Pouca E Sobrecarga = Média E Sobretensão = Leve E Subtensão = Média ENTÃO Nota = Média
SE Manobras = Pouca E Sobrecarga = Leve E Sobretensão = Média E Subtensão = Média ENTÃO Nota = Média
SE Manobras = Média E Sobrecarga = Leve E Sobretensão = Leve E Subtensão = Leve ENTÃO Nota = Média
SE Manobras = Média E Sobrecarga = Média E Sobretensão = Leve E Subtensão = Leve ENTÃO Nota = Média
SE Manobras = Média E Sobrecarga = Leve E Sobretensão = Média E Subtensão = Leve ENTÃO Nota = Média
SE Manobras = Média E Sobrecarga = Leve E Sobretensão = Leve E Subtensão = Média ENTÃO Nota = Média
SE Manobras = Muita OU Sobrecarga = Pesada OU Sobretensão = Pesada OU Subtensão = Pesada ENTÃO Nota = Ruim

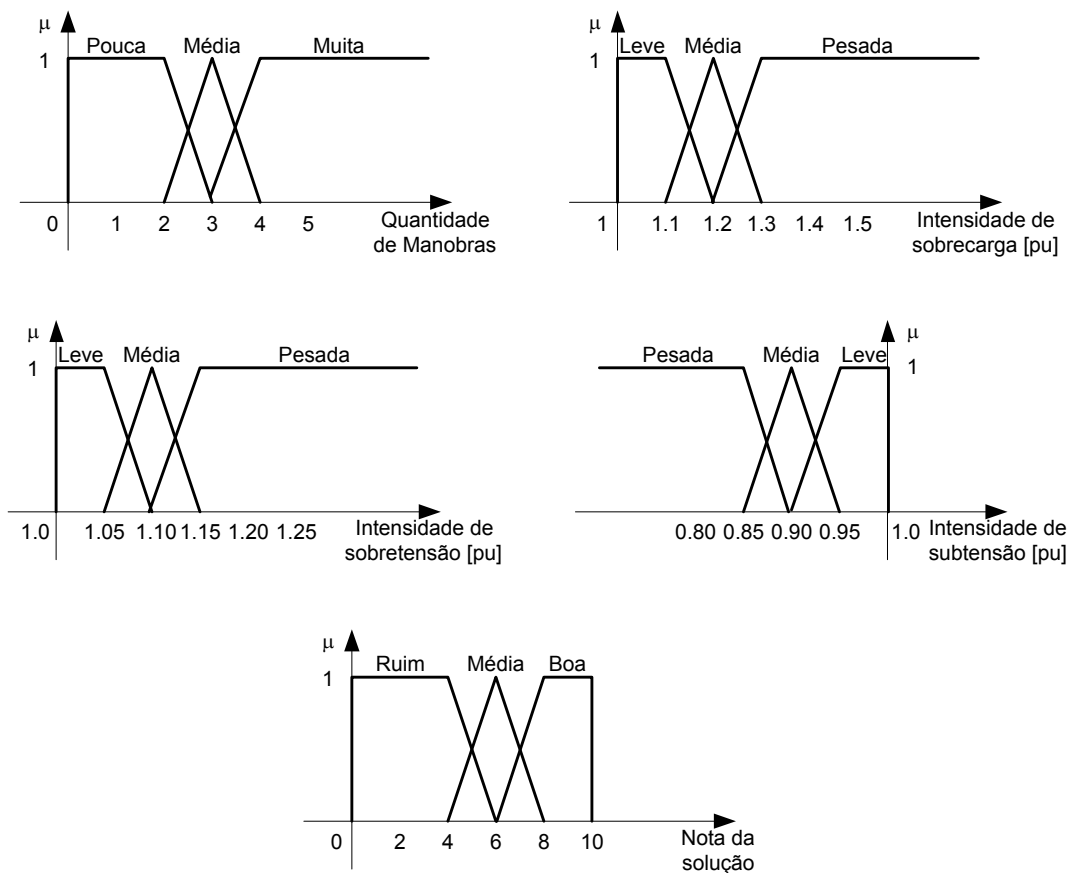


Figura 4.14 – Funções difusas para classificação

Esse procedimento inicia-se com a *fuzzificação* das variáveis envolvidas, em seguida são avaliadas as regras e então é feita a *defuzzificação* pelo método do centróide.

Se “Agente Planejador” avaliar alguma solução como boa ele envia para o usuário, senão ele tenta uma manobra para cortar cargas não prioritárias chamando o “Agente Corte Seletivo de Carga”.

4.3.1.2 Agente Manobras

O “Agente Manobras” é o responsável por realizar uma análise topológica do sistema após a falha e determinar quais chaves de socorro podem ser manobradas de tal forma a restabelecer o fornecimento de energia.

Foram implementados dois algoritmos diferentes, sendo o primeiro denominado “Avançado”, pois necessita das seguintes informações para o seu processamento: barra origem, barra destino e fator de busca; já o segundo foi denominado “Básico” por precisar somente da barra destino.

O algoritmo “Avançado” é constituído por um sistema especialista que busca caminhos viáveis para a energia chegar a todas as barras desenergizadas (Figura 4.15). O método de busca em amplitude empregado é o mesmo apresentado por VALIQUETTE, LAMBERT-TORRES & MUKHEDKAR (1991).

Esse algoritmo permite buscar diversos caminhos através do ajuste do fator de busca, o qual permite que o sistema especialista continue processando mesmo depois de ter encontrado um primeiro caminho.

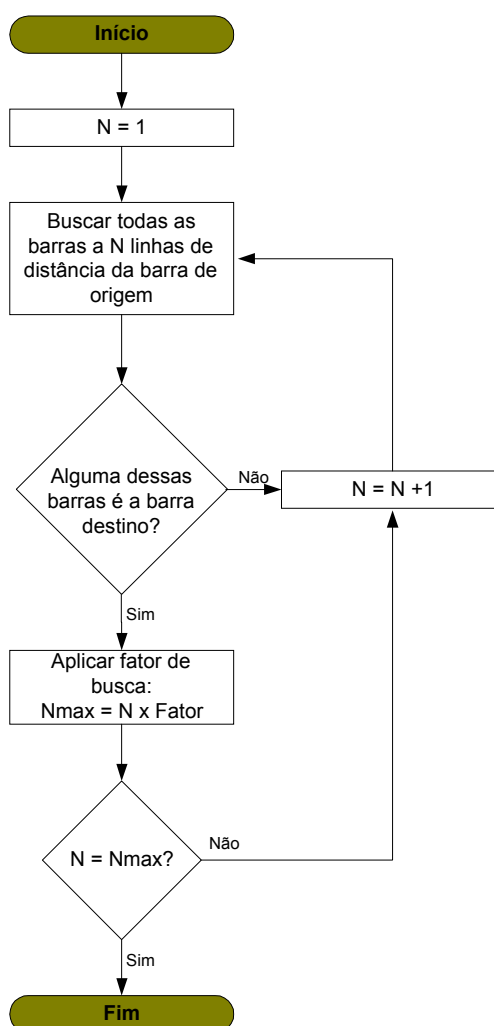


Figura 4.15 – Fluxograma de ações do algoritmo “Avançado”

Já o algoritmo “Básico” inicia-se separando para cada circuito alimentador uma lista de barras energizadas e outra de barras desenergizadas, conforme algoritmo apresentado na Figura 4.16.

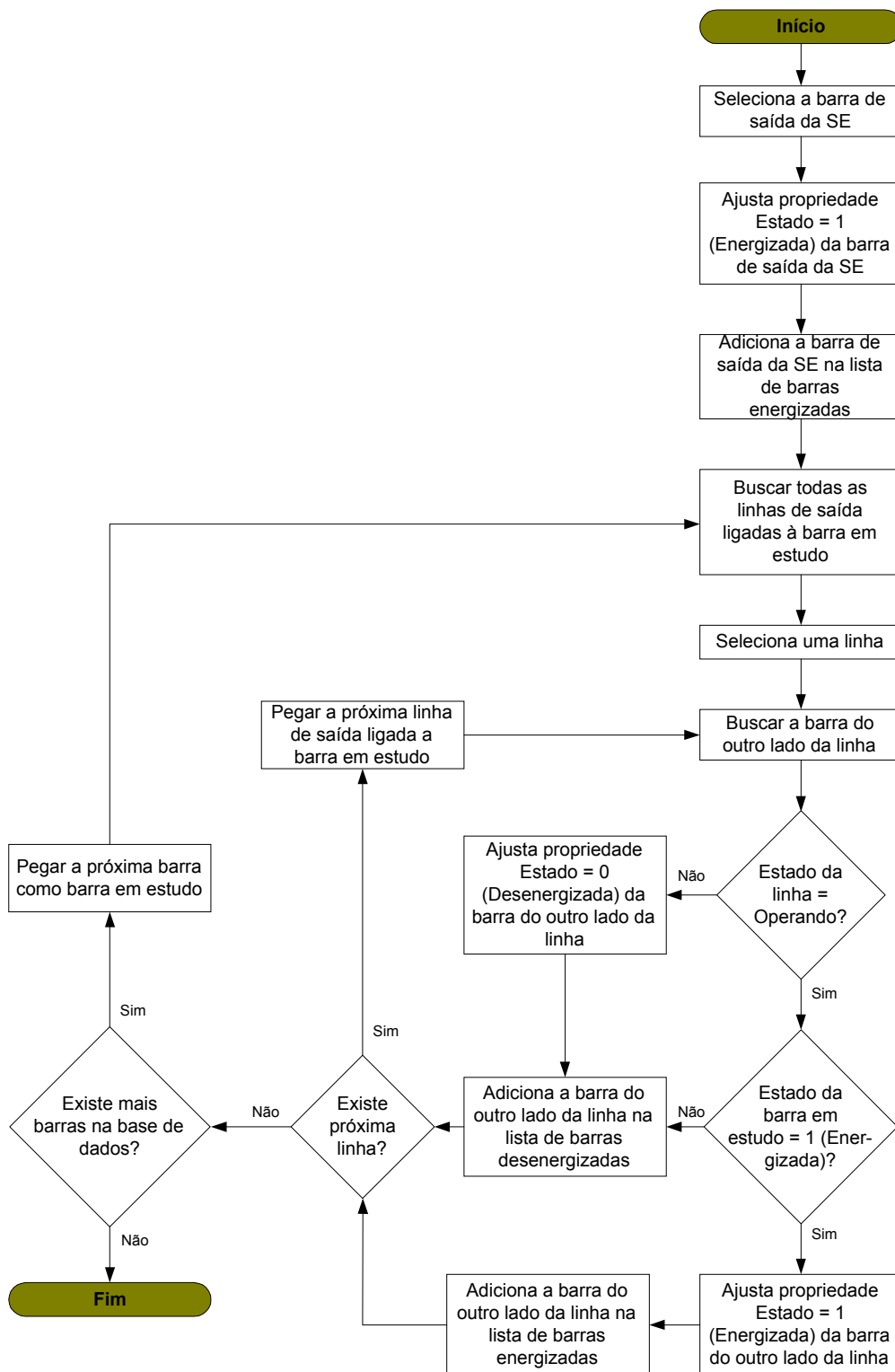


Figura 4.16 – Algoritmo que cria listas de barras energizadas e desenergizadas

Uma vez identificado o que está funcionando e o que não está, é preciso determinar dentro do conjunto de barras desenergizadas quantas e quais são as falhas, ou seja, a partir da lista de barras desenergizadas cria-se listas de barras em falha através do algoritmo visto na Figura 4.17.

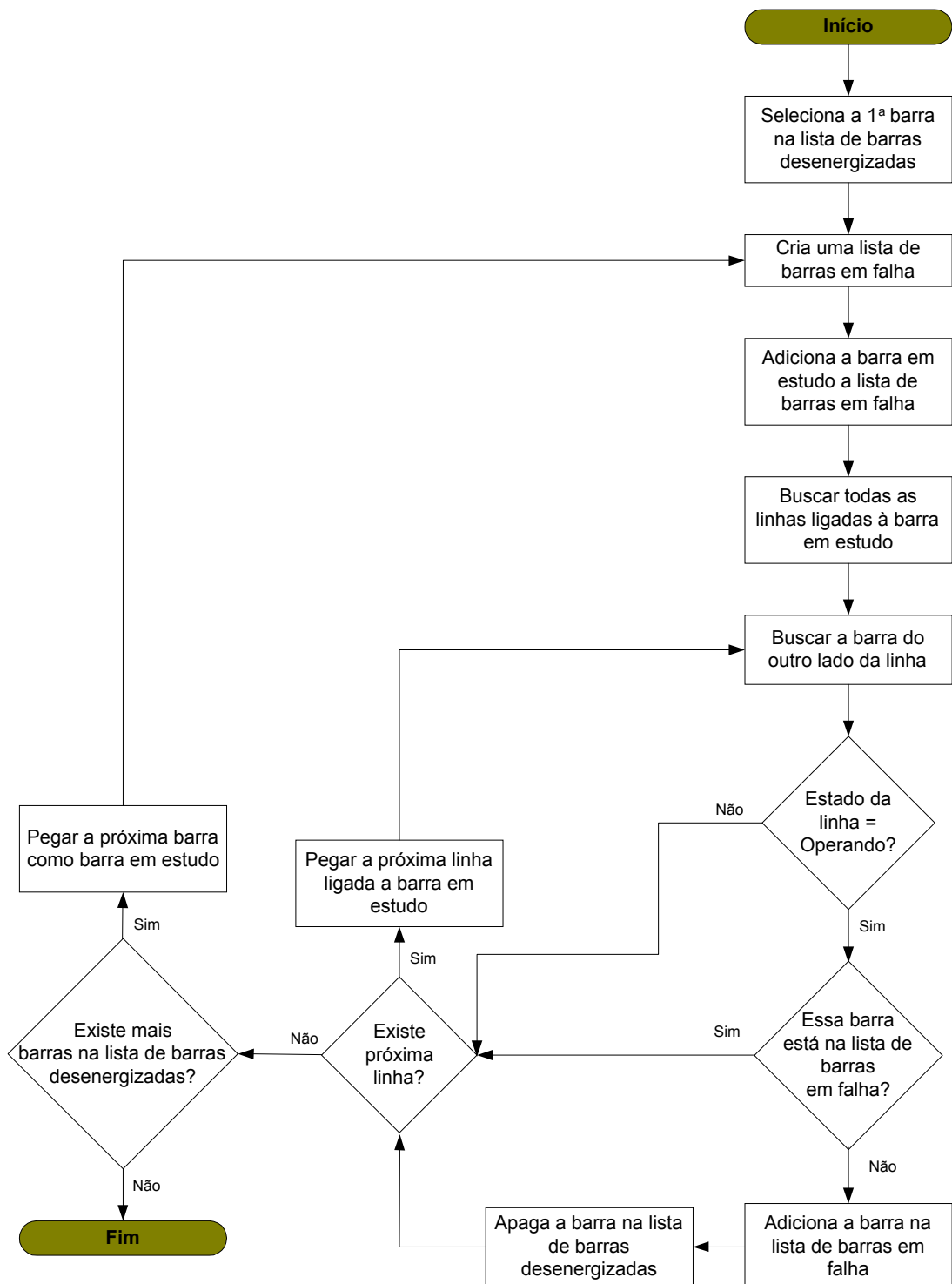


Figura 4.17 – Algoritmo que cria listas de barras para cada falha

Cada lista de barras em falha constitui uma falha ocorrida no sistema de distribuição. Para cada lista será feito um levantamento das linhas que possuem chaves de socorro e se elas estão aptas a operar, conforme Figura 4.18. Cada chave apta é uma possível solução para a falha.

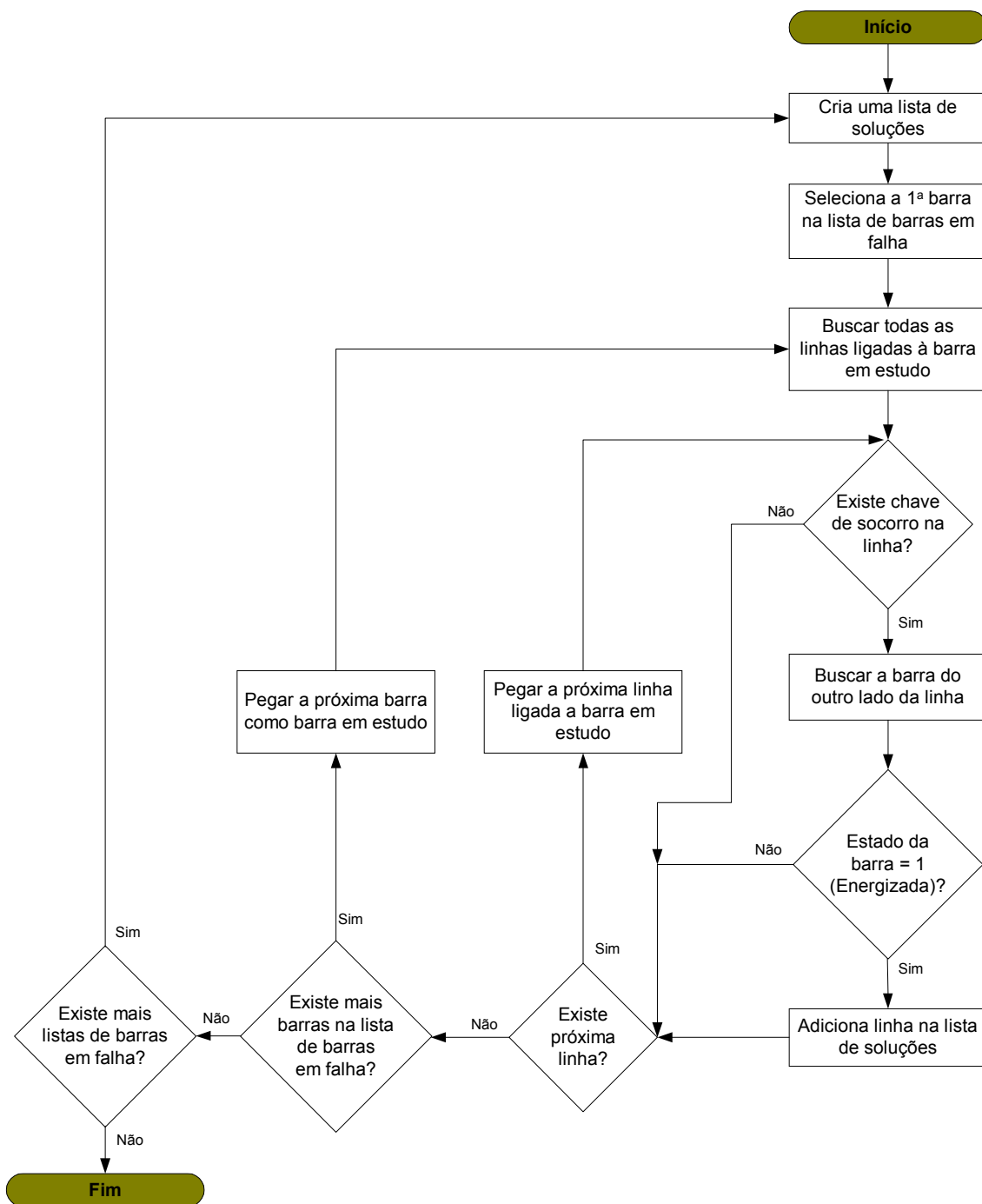


Figura 4.18 – Algoritmo que busca soluções para as falhas

4.3.1.3 Agente Fluxo de Carga

O “Agente Fluxo de Carga”, não é propriamente inteligente, pois não tem um programa com Inteligência Artificial sendo executado internamente, e sim, um algoritmo numérico de cálculo de fluxo de carga responsável por fornecer dados da situação do sistema em uma certa configuração, pré ou pós-falha, conforme Figura 4.19.

Para cada falha serão analisadas todas as soluções possíveis, sendo que para cada caso serão armazenadas as características de sobrecargas, sobretensões e subtensões ocasionadas no sistema.

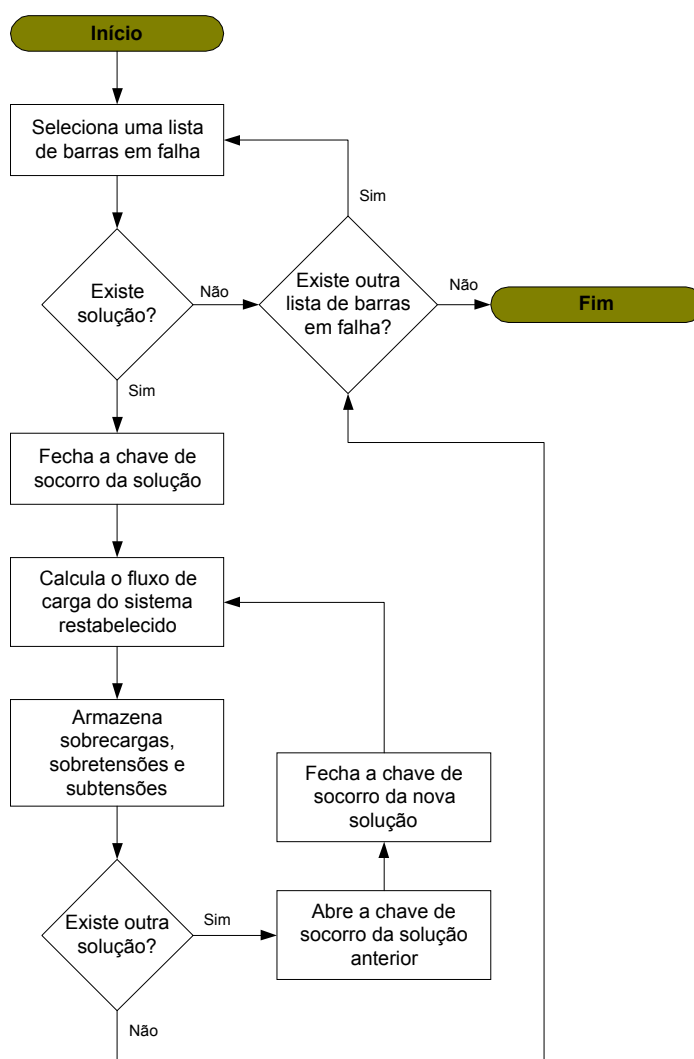


Figura 4.19 – Algoritmo geral de fluxo de carga

O algoritmo utilizado para calcular as correntes nas linhas está apresentado na Figura 4.20.

Primeiramente, é montada uma base de dados com as barras em falha, as quais serão restabelecidas, mais as barras do sistema que existe do outro lado da chave de socorro. Em seguida todas as tensões nas barras são passadas para $1\angle 0^\circ$ pu, e ainda, todas as grandezas de carga ativa e reativa das barras são convertidas para pu, para isso serão calculadas as potências aparentes de cada barra que possuir carga.

O próximo passo é calcular todas as correntes no sistema para essa tensão ($1\angle 0^\circ$ pu). Em seguida são recalculadas todas as tensões nas barras, de acordo com as correntes calculadas, Figura 4.21. Esse processo é repetido até que uma condição de erro seja satisfeita e só funciona para sistemas de distribuição radiais.

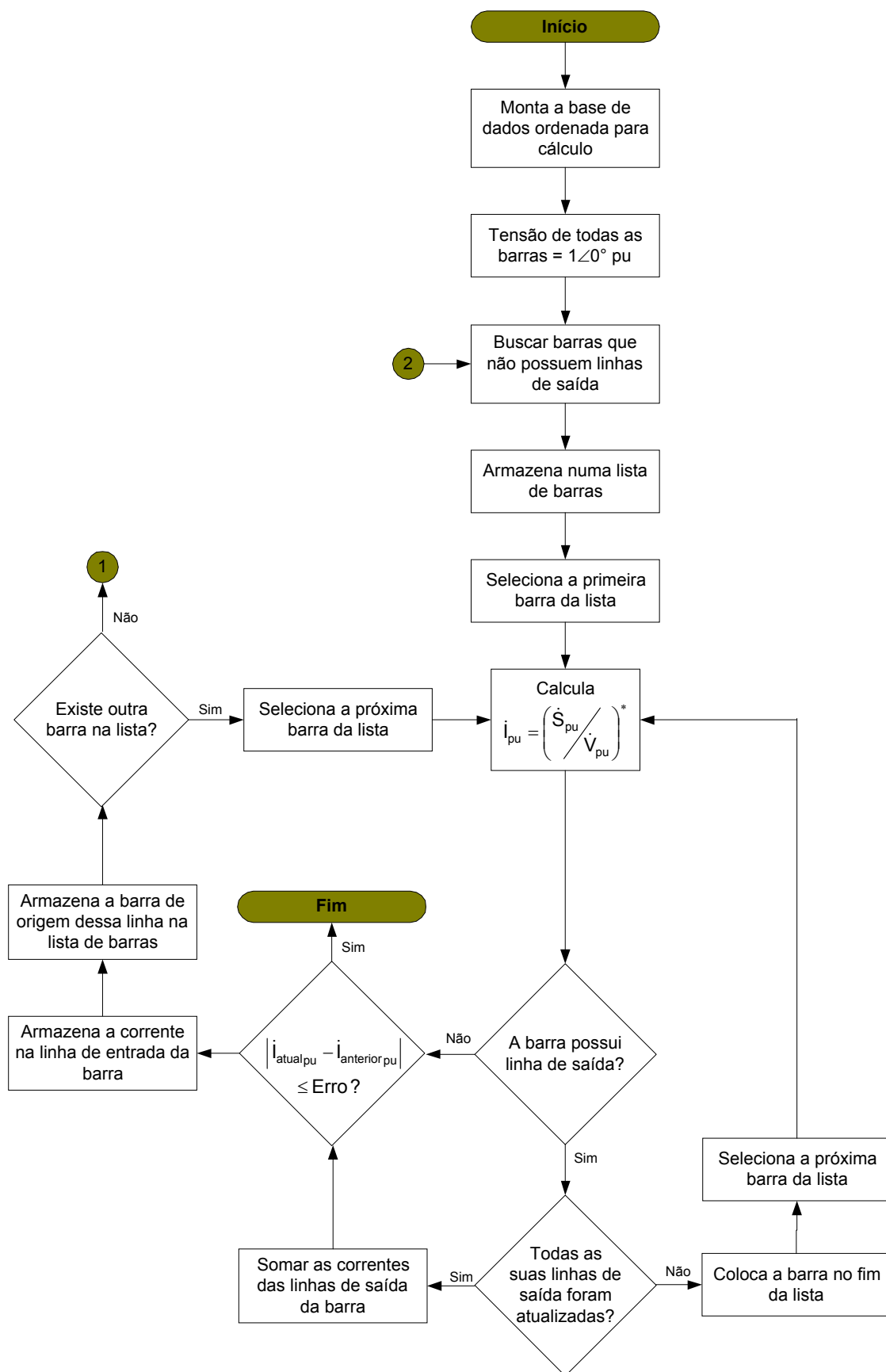


Figura 4.20 – Algoritmo para cálculo das correntes no sistema

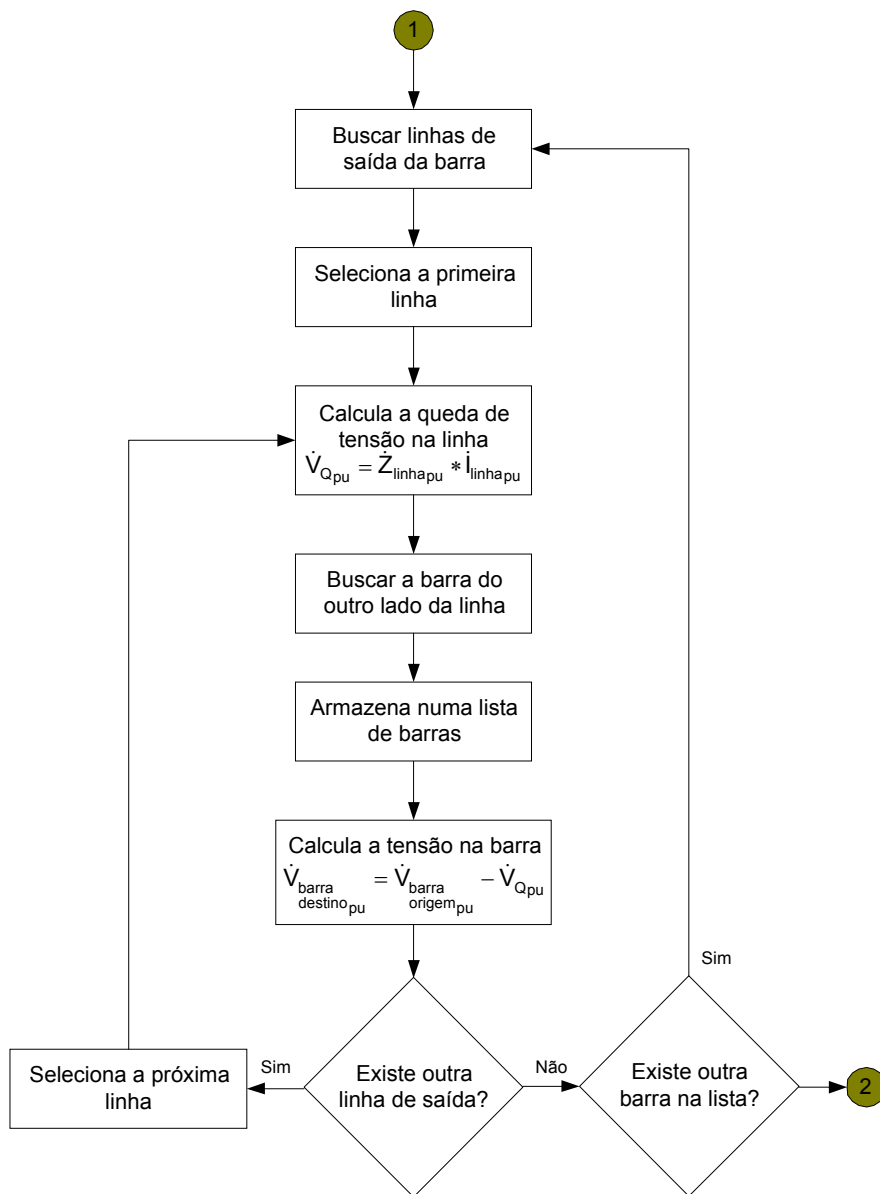


Figura 4.21 – Algoritmo para cálculo das tensões nas barras

4.3.1.4 Agente Corte Seletivo de Carga

O “Agente Corte Seletivo de Carga” provê experiência para colaborar no restabelecimento e na manutenção da estabilidade do sistema. Quando todas as manobras sugeridas impuserem ao sistema uma condição operativa insegura será necessário planejar de novo as ações de restabelecimento levando em consideração o não atendimento de cargas não prioritárias. O algoritmo que corta as cargas não prioritárias da análise está apresentado na Figura 4.22.

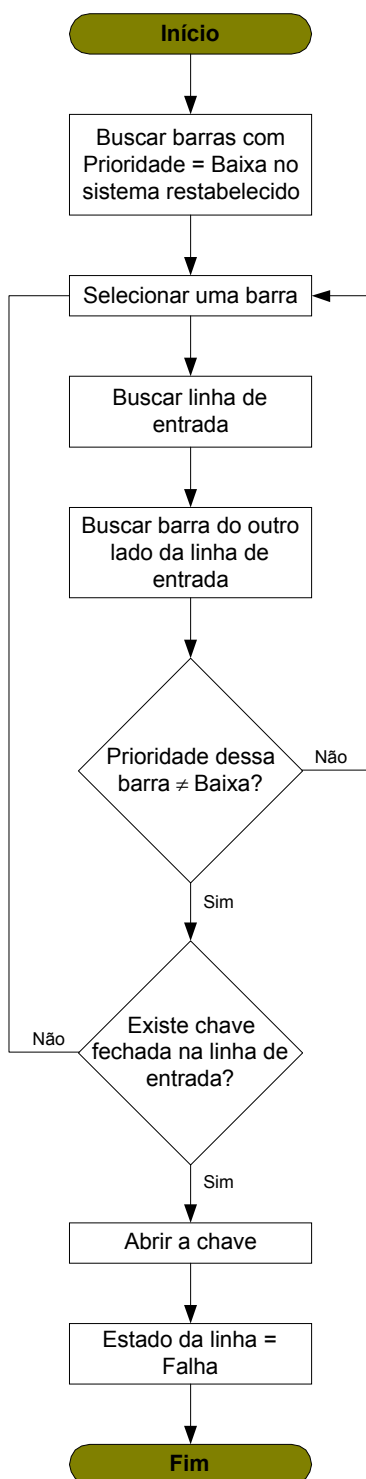


Figura 4.22 – Algoritmo para cortar cargas não prioritárias

Nesse algoritmo, primeiro monta-se uma lista com as barras de baixa prioridade, em seguida busca-se quais dessas barras estão na fronteira com áreas de prioridade diferentes. Isso porque geralmente nessas fronteiras encontram-se as chaves que permitem manobrar o sistema de distribuição.

Uma vez encontrada essa chave, ela é aberta e a linha onde ela está localizada passa a ser considerada em falha.

4.3.2 Diagrama de Interação entre Agentes

O diagrama de interação do modelo de SMA para aplicação em sistemas de distribuição é apresentado na Figura 4.23.

Neste diagrama são explicitadas todas as comunicações existentes entre os agentes. O ciclo de análise inicia com o “Agente de Interface” que ao perceber uma alteração na base de dados do sistema SCADA faz uma coleta dos dados, interpreta-os e informa o “Agente Modelo” via “Agente de Comunicação”.

O “Agente Modelo” então atualiza o modelo orientado a objetos e envia os novos dados para o “Agente Restabelecimento”, o qual é responsável por gerenciar o procedimento de busca da solução e avaliá-la.

O “Agente Restabelecimento” solicita ao “Agente Manobras” que determine as soluções possíveis. Para cada solução encontrada pelo “Agente Manobras” o “Agente Restabelecimento” irá solicitar ao “Agente Fluxo de Carga” para determinar todas as grandezas do sistema considerando aquela manobra efetivada. Dessa forma, pode-se analisar o estado do sistema de distribuição posterior à tomada de decisão.

Todas as manobras que levarem a estados inseguros do sistema serão descartadas, ou mal classificadas. Caso não haja manobra que não leve a estados inseguros, o “Agente Restabelecimento” solicita ao “Agente Corte Seletivo de Carga” que indique quais barras não são prioritárias para a reenergização e qual chave deve ser manobrada para eliminá-las da análise.

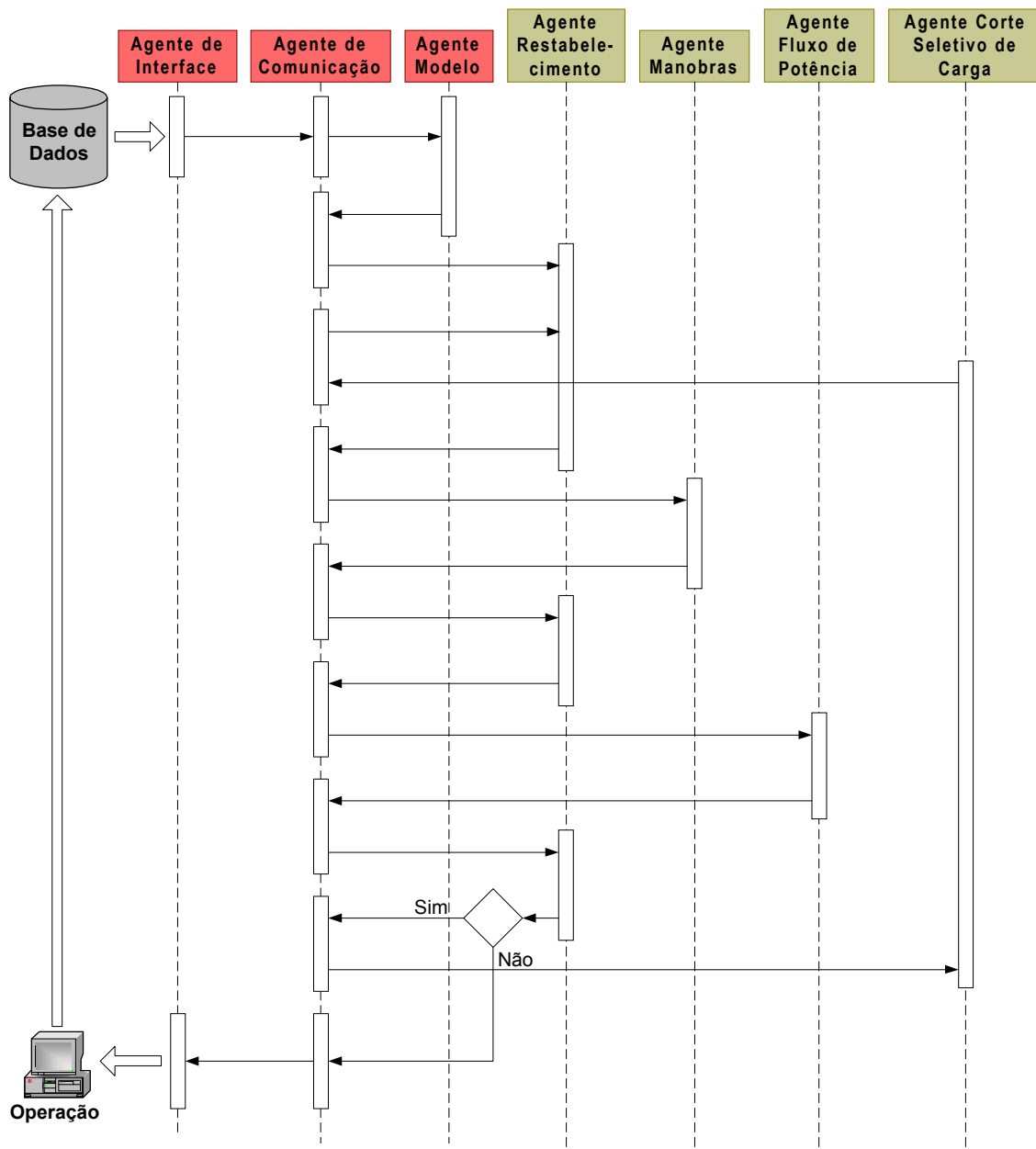


Figura 4.23 – Diagrama de interação para o SMA para sistemas de distribuição

Uma vez que o “Agente Corte Seletivo de Carga” realize sua função, o “Agente Restabelecimento” inicia o processo novamente solicitando ao “Agente Manobras” que determine quais manobras atendem à nova configuração (desconsiderando as barras não prioritárias).

Esse processo é repetido mais uma vez e caso não se obtenha nenhuma solução boa, o SMA apresentará todas as soluções encontradas.

4.4 Uma Proposta de Integração dos Aplicativos para Sistemas de Distribuição e para Subestações Elétricas

A integração de aplicativos desenvolvidos utilizando técnicas de Inteligência Artificial para sistemas elétricos é um desafio a ser vencido pelos especialistas da área, uma vez que cada técnica utiliza um modelo de dados diferente e possui processos de inferência distintos (BANN *et al.*, 1996). Há também as dificuldades inerentes aos processos comunicativos, como por exemplo: meio de comunicação, modo e linguagem utilizados, bem como quais dados serão compartilhados.

A arquitetura proposta nessa tese é baseada na tecnologia de Agentes Inteligentes, a qual propõe o desenvolvimento de sistemas complexos a partir de entidades de software autônomas com capacidade de raciocínio limitada. Como os agentes possuem uma estrutura e uma linguagem para comunicação eles podem interagir entre si, dessa forma eles passam a constituir uma sociedade de agentes denominada sistema multi-agente.

Um sistema multi-agente consegue, através da interação de seus integrantes, resolver problemas maiores do que o domínio de conhecimento de cada agente isolado. Para possibilitar a integração dos sistemas desenvolvidos foi implementado um núcleo de agentes responsáveis por dar suporte aos agentes especializados.

A estrutura integrada dos dois aplicativos está apresentada na Figura 4.24, e sugere a divisão em três grupos:

- **Agentes de Suporte:** responsáveis por prestar serviços que auxiliem o processo de tomada de decisões, como: estabelecer e coordenar a comunicação entre os agentes e os sistemas multi-agentes, manter um modelo computacional do mundo real atualizado, interface com o mundo real, entre outros.
- **Agentes Especializados em Restabelecimento de Subestações Elétricas:** responsáveis por analisar as

condições dos diversos equipamentos existentes na SE e sugerir ações para o restabelecimento da mesma.

- **Agentes Especializados em Restabelecimento de Sistemas de Distribuição:** responsáveis por analisar as condições do sistema de distribuição, buscar manobras para o seu restabelecimento e analisar os efeitos dessas manobras nas condições operativas do sistema em análise.

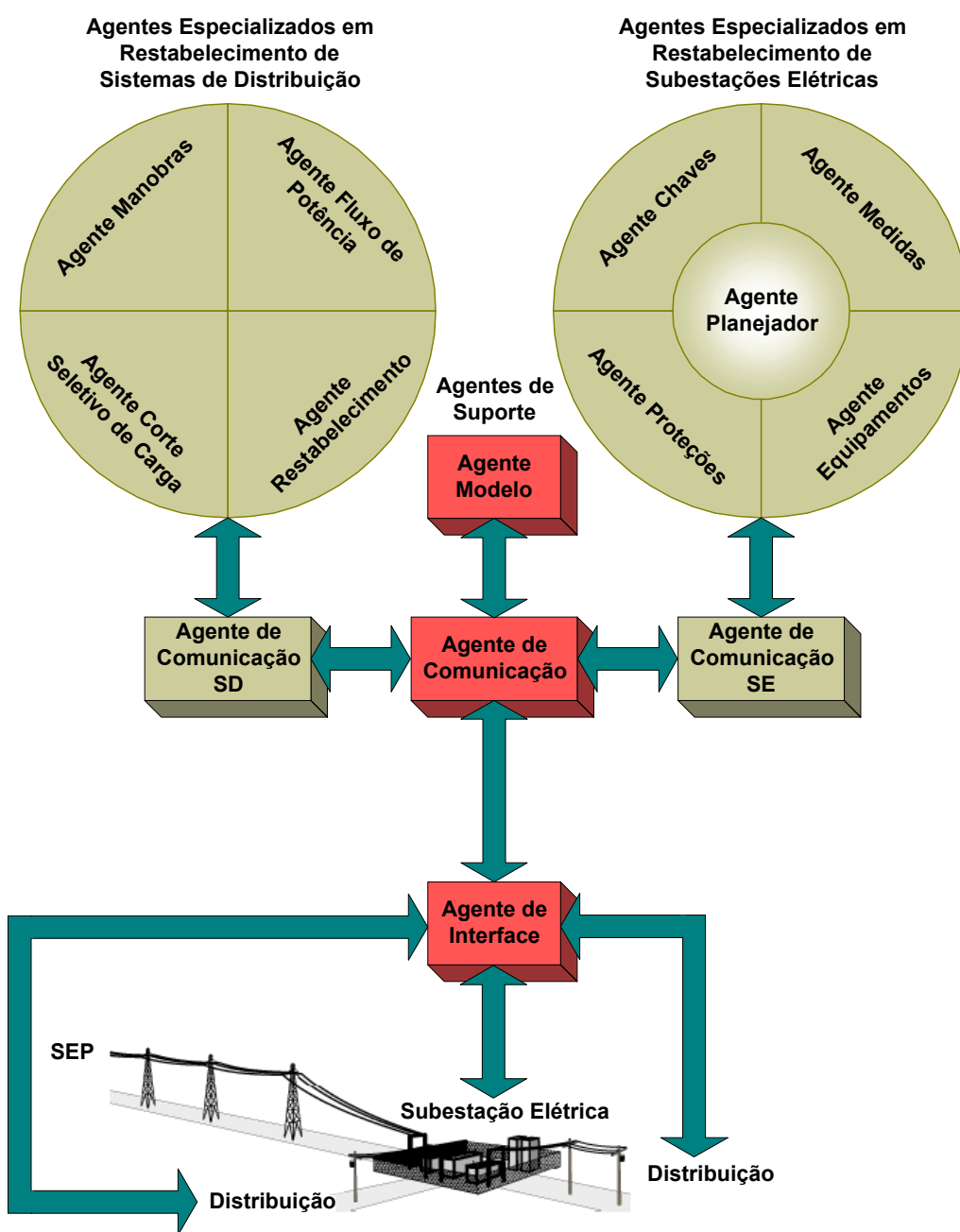


Figura 4.24 – Sistema integrado para restabelecimento de sistemas elétricos

Têm-se então três programas computacionais desenvolvidos na mesma plataforma: um programa principal composto pelos agentes de suporte, um programa para restabelecimento de subestações elétricas composto pelos seus agentes especializados mais o “Agente de Comunicação SE” e um programa para restabelecimento de sistemas de distribuição também composto pelos seus agentes especializados mais o “Agente de Comunicação SD”. Deve-se implementar um “Agente de Comunicação” para cada pacote de agentes especializados para otimizar o processo de troca de informação interna dos seus integrantes e facilitar a comunicação com outros sistemas multi-agentes.

Com a utilização dos agentes de suporte é possível ter uniformidade e coerência nos dados utilizados pelos dois pacotes de agentes especializados. Como há sobreposição dos dados vistos pelos pacotes de agentes especializados as alterações solicitadas por um aplicativo serão percebidas pelo outro aplicativo também.

Por exemplo, o aplicativo para o restabelecimento de subestações elétricas enxerga até a barra de saída da subestação, mais seus alimentadores juntamente com os disjuntores desses; já o aplicativo para o restabelecimento de sistemas de distribuição visualiza o sistema em análise desde da barra da SE até a última barra do ramal do alimentador.

Toda ação sugerida por qualquer um dos programas especializados para um estado operativo do sistema elétrico será percebida pelo outro programa no próximo estado operativo.

A implementação computacional dessa proposta de integração constitui um desenvolvimento futuro para avançar a pesquisa nessa área.

5 TESTES, RESULTADOS E IMPLEMENTAÇÃO

5.1 Falha na Barra da Subestação

O Sistema Inteligente para a Supervisão e Controle de Subestações Elétricas (SISCO - Subestações) foi desenvolvido empregando a arquitetura apresentada no item 4.2 para a planta da subestação ETD Estuário, localizada na região da Baixada Santista e na época pertencente a Bandeirante Energia S/A, Figura 5.1.

Essa é uma subestação de distribuição de 138/13,8 kV que possui duas linhas de entrada: LTS BOQ ESU-1 e LTS BOQ ESU-2. Cada linha possui um transformador de 40/60 MVA de três enrolamentos, sendo: primário em delta com tap variável, secundário e terciário em estrela aterrada. Cada trafo alimenta duas barras, em cada barra há quatro alimentadores e um banco de capacitores de 3,6 MVar. Existe um disjuntor entre as barras BA-1 e BA-2 e outro entre as barras BA-3 e BA-4.

Foi simulada uma sobrecorrente transitória no barramento BA-1, com a medição do valor de 2200 [A] via amperímetro BA-1_A, Figura 5.2.

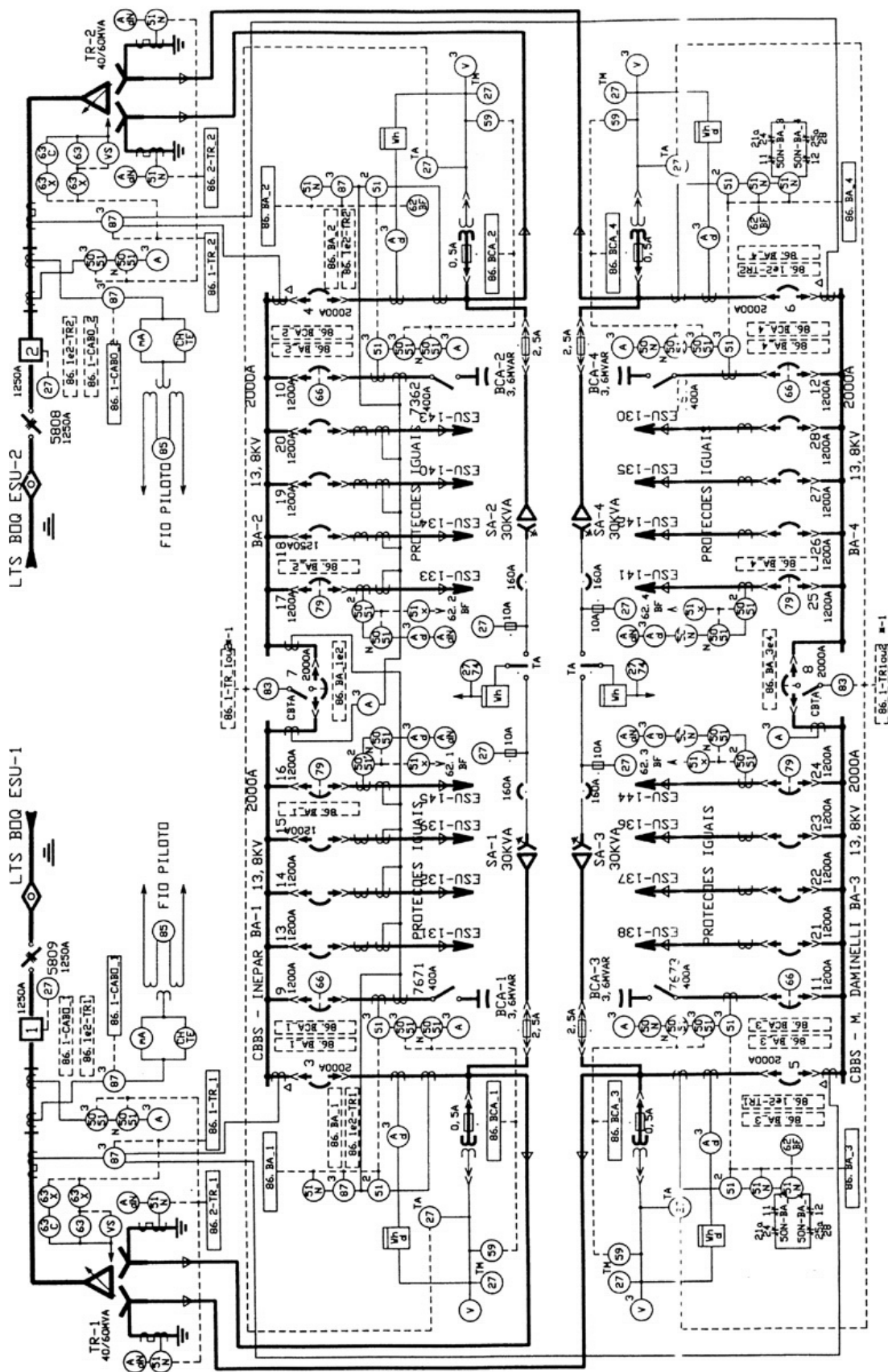


Figura 5.1 – ETD Estuário – Bandeirante Energia S/A

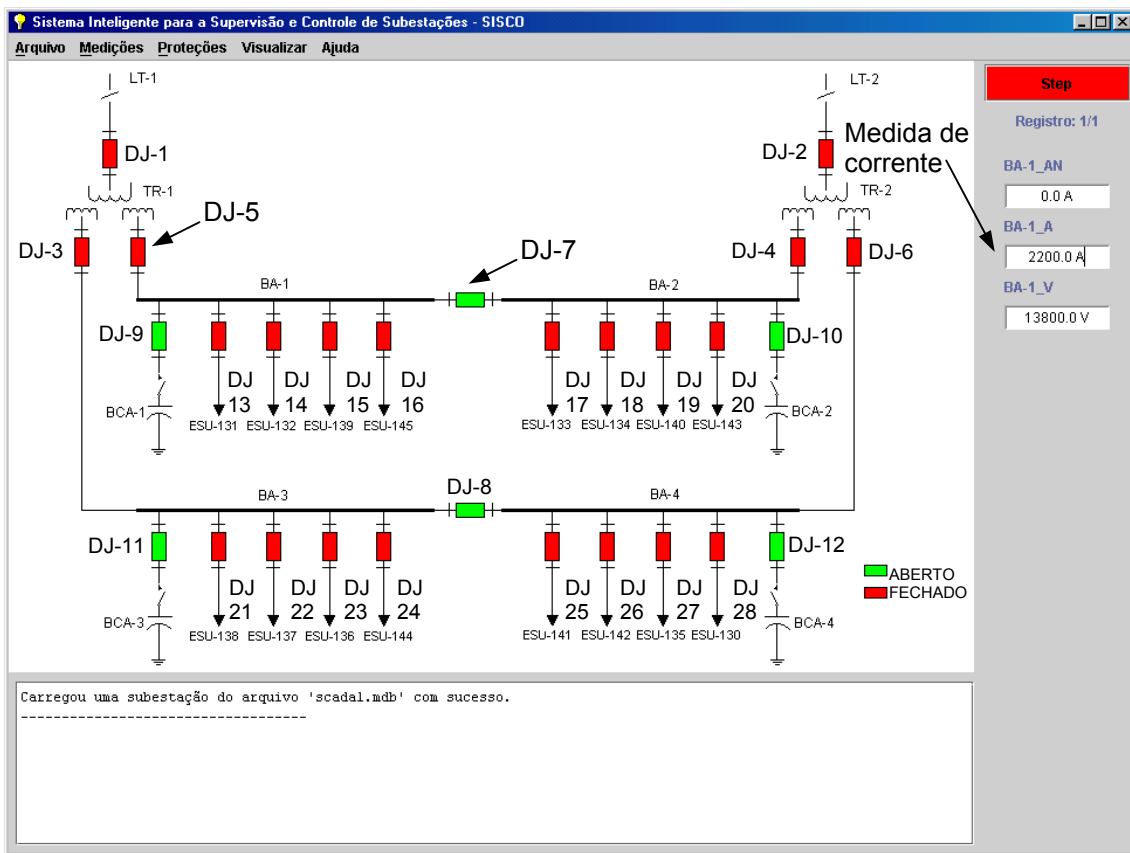


Figura 5.2 – Falha em BA-1 - Passo 1

Primeiramente, o relé BA-1_51 de sobrecorrente irá atuar. A proteção sugere a abertura do disjuntor DJ-5 de alimentação da barra e do disjuntor DJ-7 de paralelo. Como o disjuntor DJ-7 já se encontrava aberto o “Agente Planejador” retirou essa sugestão da lista final, Figura 5.3.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:
- Abrir Disjuntor DJ-5. (ProtectionAgent)
 - Atuar Proteção BA-1_51. (ProtectionAgent)
-

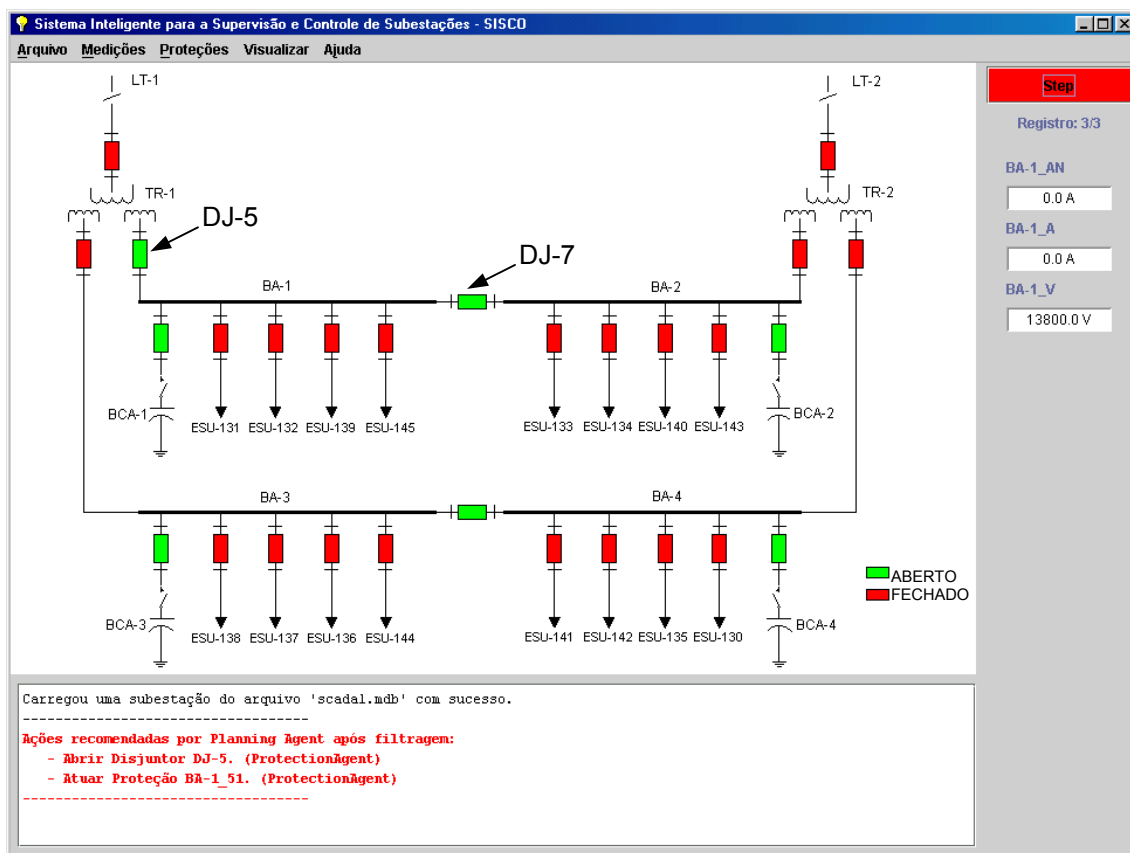


Figura 5.3 – Falha em BA-1 - Passo 2

Com a falta de tensão na barra, sugere-se a abertura de todos os disjuntores dos alimentadores (Figura 5.4), pois é padrão operativo da empresa somente alimentar cargas após ser totalmente restabelecida a energia na barra alimentadora.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:**
- Abrir Disjuntor DJ-16. (MeasureAgent)
 - Abrir Disjuntor DJ-15. (MeasureAgent)
 - Abrir Disjuntor DJ-14. (MeasureAgent)
 - Abrir Disjuntor DJ-13. (MeasureAgent)
 - Desatuar Proteção BA-1_51. (ProtectionAgent)

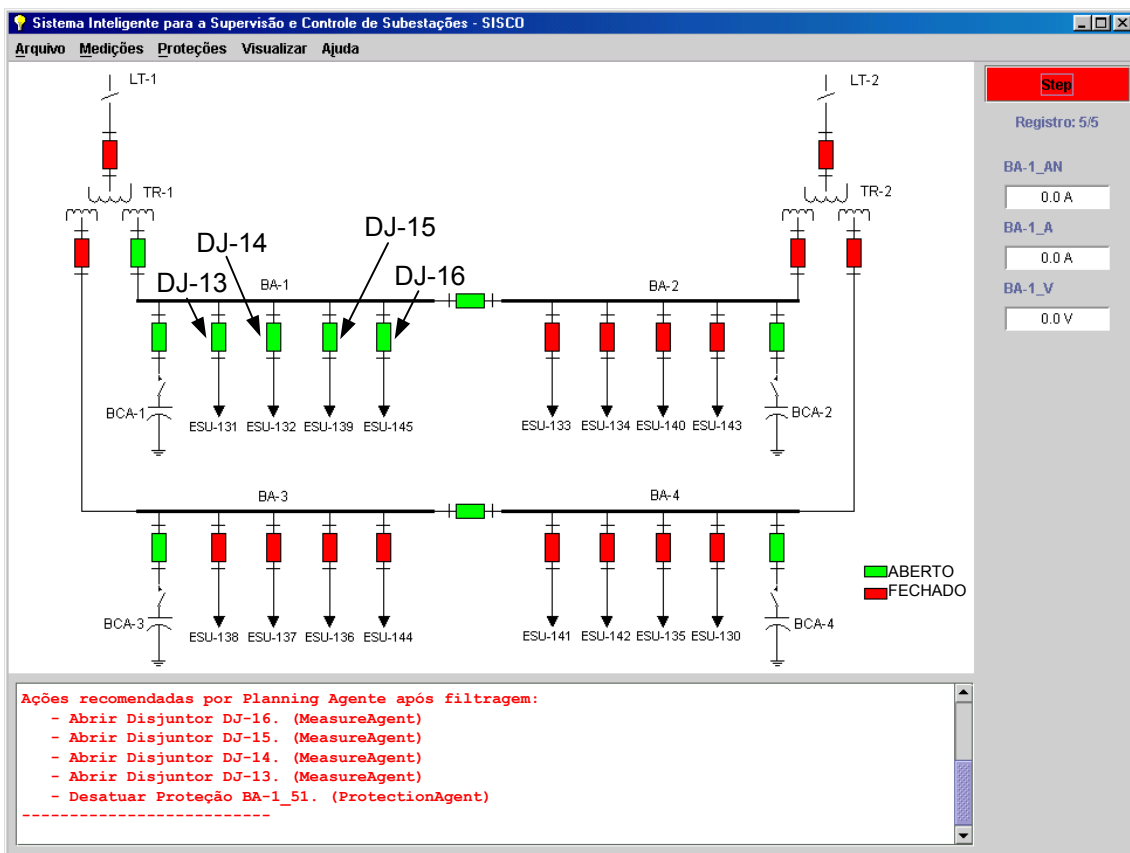


Figura 5.4 – Falha em BA-1 - Passo 3

Verificado o fim da falha inicia-se o processo de restabelecimento sugerindo-se o fechamento do disjuntor DJ-5 de alimentação da barra, Figura 5.5. As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

```

*****
* Término da falha. Iniciando restabelecimento. Ajustar medições. *
*****
- Recomenda-se restabelecer o Disjuntor DJ-5 e ajustar suas
medições.
-----
    
```

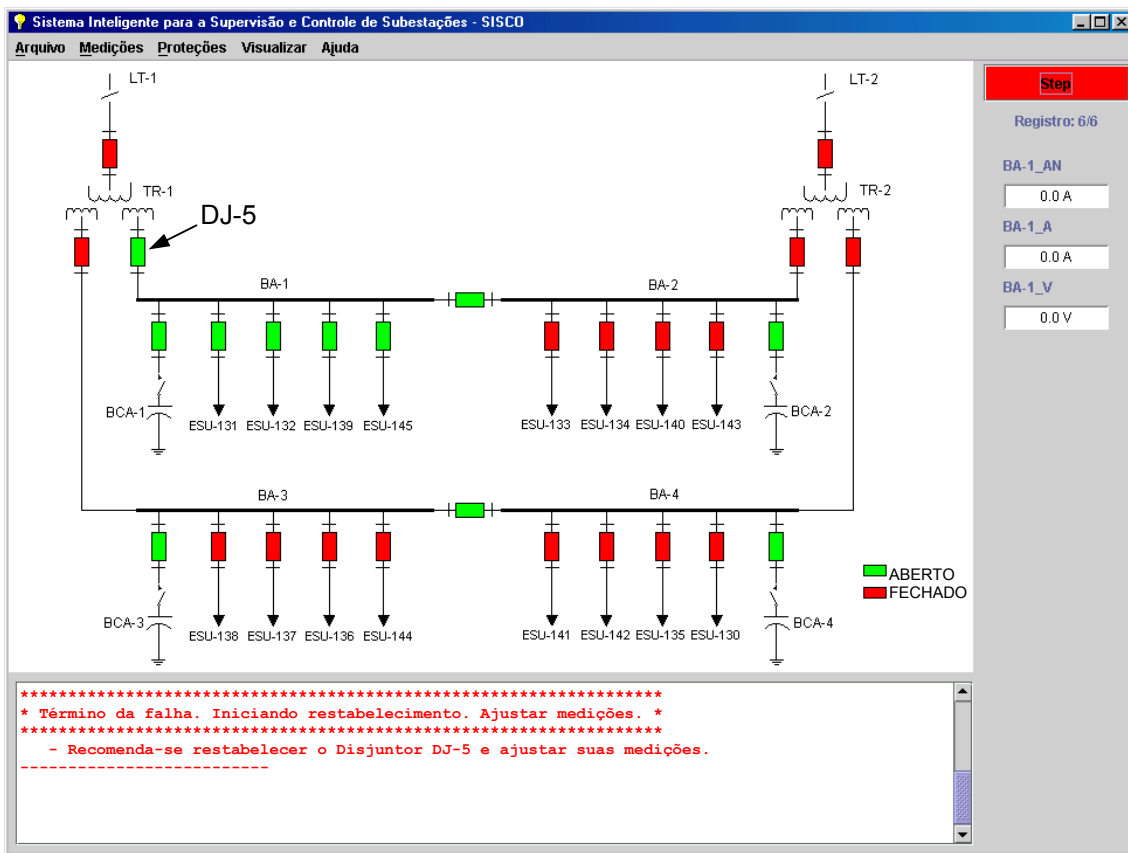


Figura 5.5 – Falha em BA-1 - Passo 4

Uma vez restabelecida a barra, o próximo passo, assim que verificado que a barra está energizada, é a sugestão do fechamento dos disjuntores dos alimentadores, Figura 5.6.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

```
Ações recomendadas por Planning Agent após filtragem:
- Recomenda-se restabelecer o Disjuntor DJ-13 e ajustar suas
medições.
- Recomenda-se restabelecer o Disjuntor DJ-14 e ajustar suas
medições.
- Recomenda-se restabelecer o Disjuntor DJ-15 e ajustar suas
medições.
- Recomenda-se restabelecer o Disjuntor DJ-16 e ajustar suas
medições.
-----
```

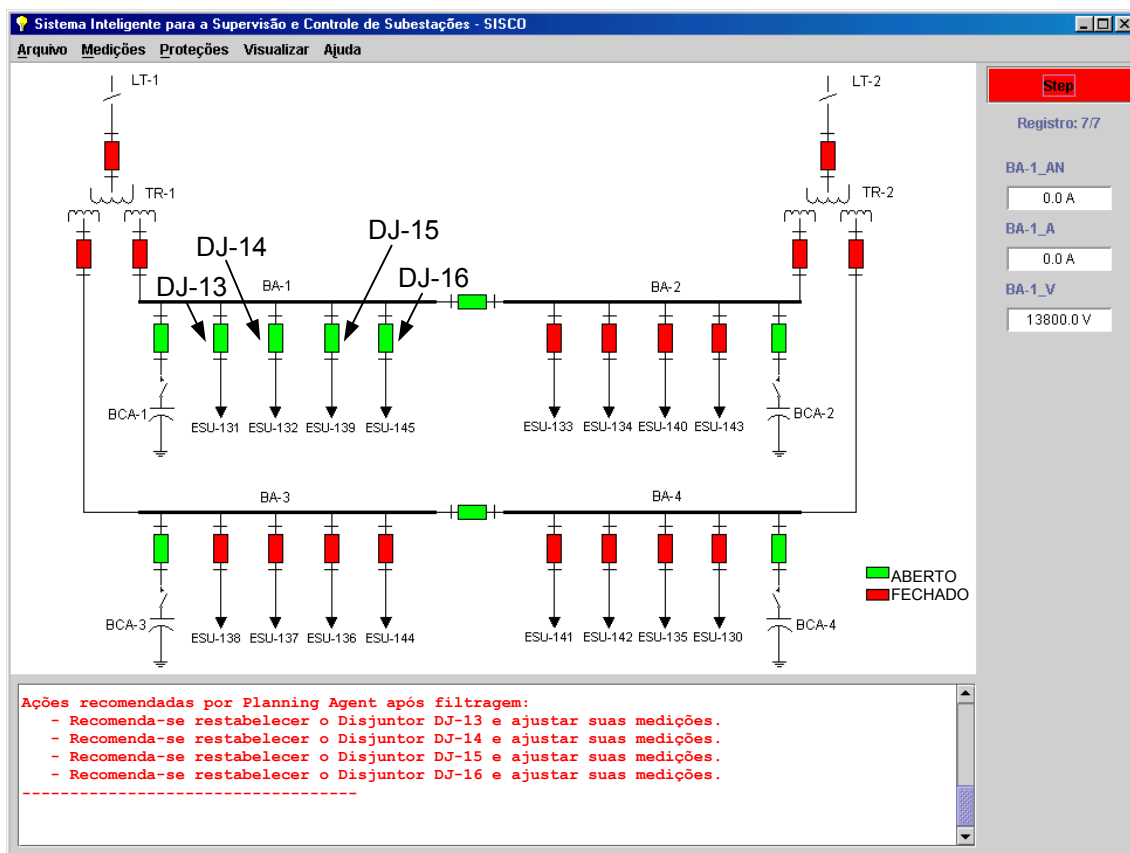


Figura 5.6 – Falha em BA-1 - Passo 5

O operador restabelece nesse passo os disjuntores DJ-13, DJ-14, DJ-15 e DJ-16, um a um, recompondo a condição normal de operação da subestação, Figura 5.7.

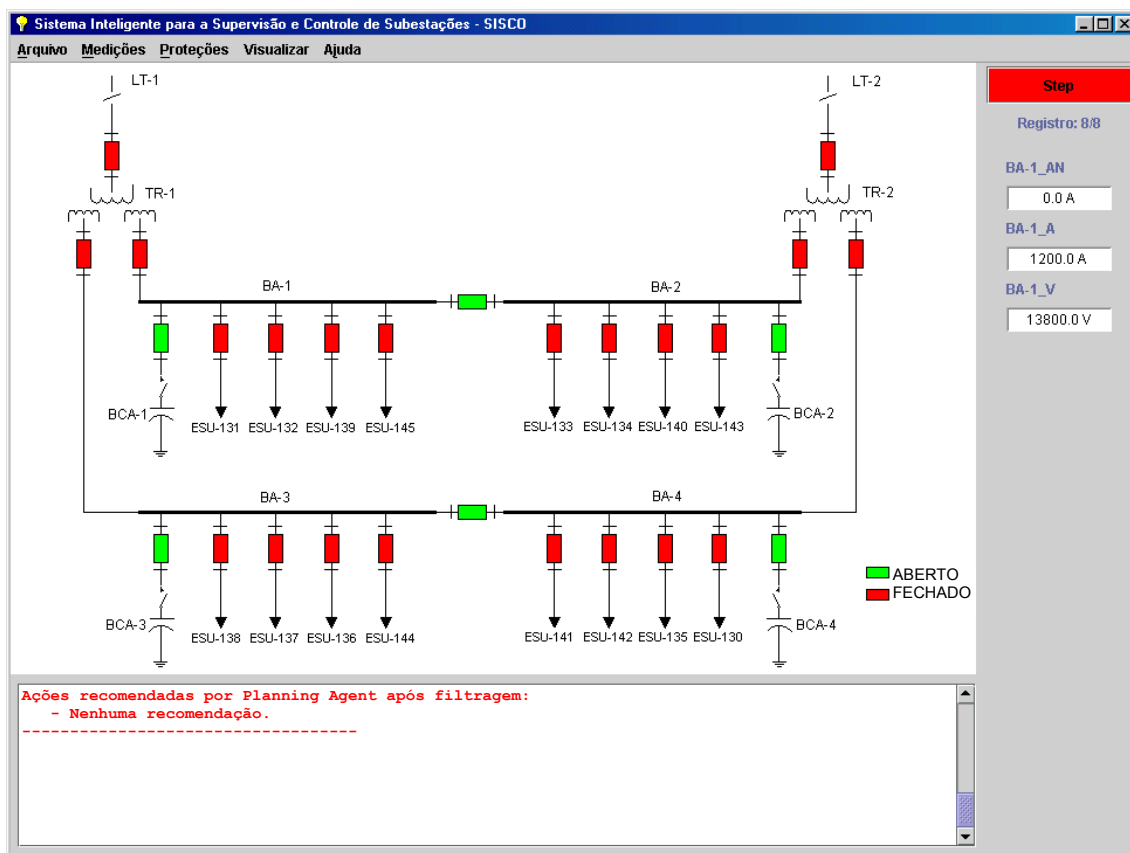


Figura 5.7 – Falha em BA-1 - Passo 6

5.2 Colapso Total na Subestação

Foram simuladas falhas de subtensão nas linhas de transmissão que alimentam a subestação, com a medição do valor de 100.000 [V] via voltímetro LT-1_V e LT-2_V, Figura 5.8.

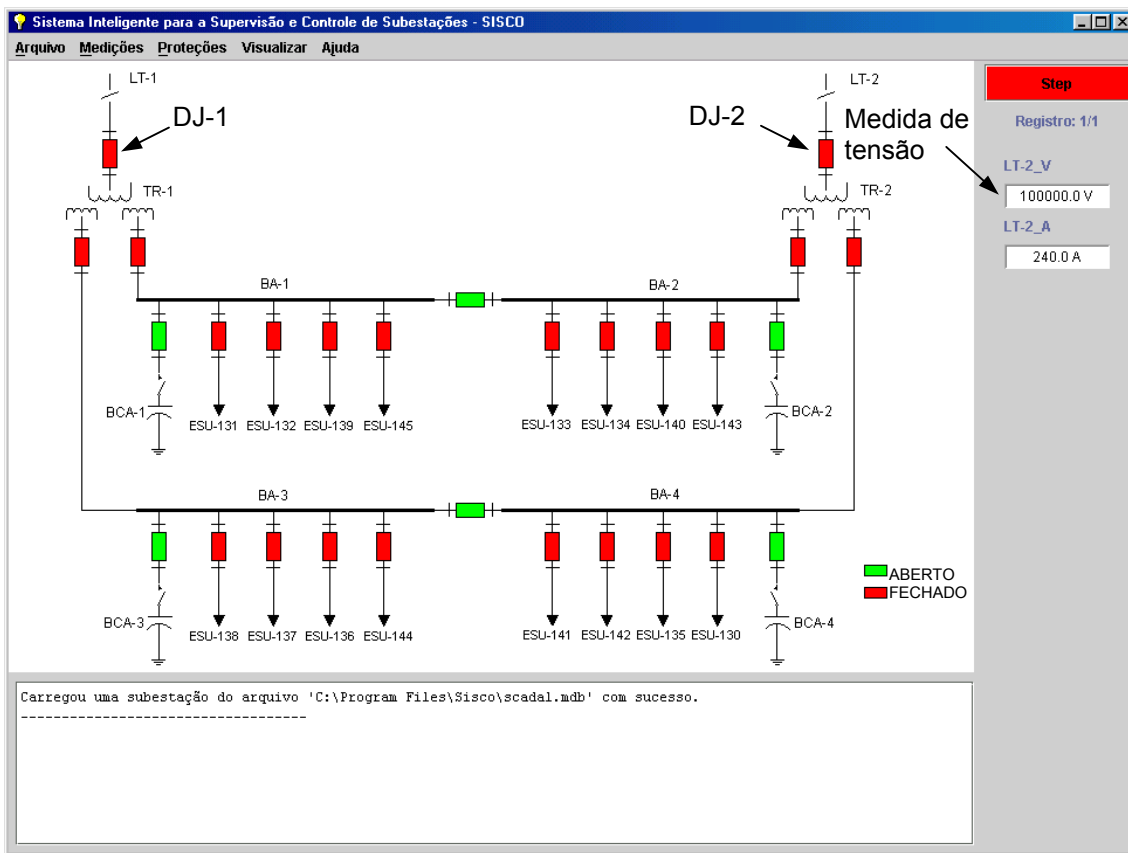


Figura 5.8 – Colapso total - Passo 1

Primeiramente, os relés de subtensão LT-1_27 e LT-2_27 irão atuar, e será sugerida a abertura dos disjuntores de alta dos transformadores, desenergizando toda a SE, Figura 5.9.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:
- Abrir Disjuntor DJ-2. (ProtectionAgent)
 - Abrir Disjuntor DJ-1. (ProtectionAgent)
 - Atuar Proteção LT-2_27. (ProtectionAgent)
 - Atuar Proteção LT-1_27. (ProtectionAgent)

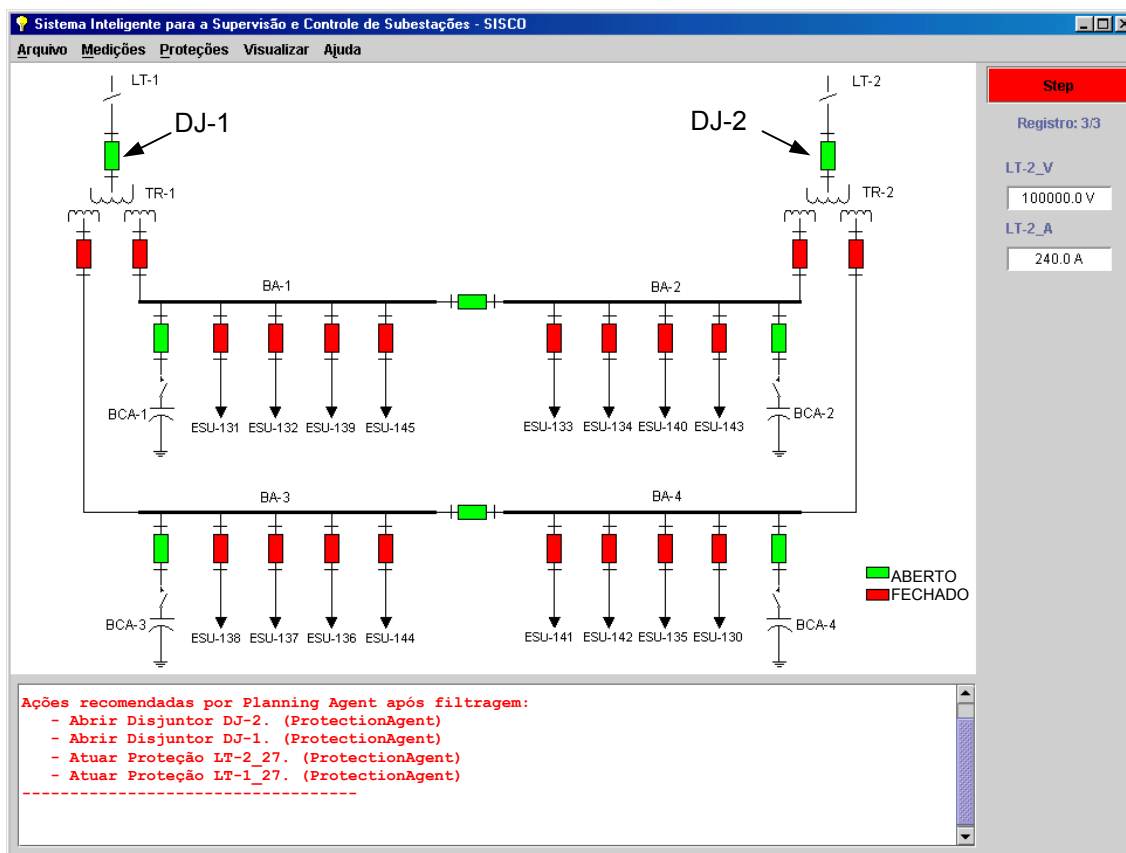


Figura 5.9 – Colapso total - Passo 2

Conforme norma operativa da empresa, com a falta de tensão em todas as barras, todos os disjuntores da subestação receberão sugestão para sua abertura, o que está de acordo com a etapa de preparação da subestação para o seu restabelecimento após um colapso total, Figura 5.10.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:
- Abrir Disjuntor DJ-28. (MeasureAgent)
 - Abrir Disjuntor DJ-27. (MeasureAgent)
 - Abrir Disjuntor DJ-26. (MeasureAgent)
 - Abrir Disjuntor DJ-25. (MeasureAgent)
 - Abrir Disjuntor DJ-24. (MeasureAgent)
 - Abrir Disjuntor DJ-23. (MeasureAgent)
 - Abrir Disjuntor DJ-22. (MeasureAgent)
 - Abrir Disjuntor DJ-21. (MeasureAgent)
 - Abrir Disjuntor DJ-20. (MeasureAgent)
 - Abrir Disjuntor DJ-19. (MeasureAgent)
 - Abrir Disjuntor DJ-18. (MeasureAgent)
 - Abrir Disjuntor DJ-17. (MeasureAgent)

- Abrir Disjuntor DJ-16. (MeasureAgent)
- Abrir Disjuntor DJ-15. (MeasureAgent)
- Abrir Disjuntor DJ-14. (MeasureAgent)
- Abrir Disjuntor DJ-13. (MeasureAgent)
- Abrir Disjuntor DJ-6. (MeasureAgent)
- Abrir Disjuntor DJ-5. (MeasureAgent)
- Abrir Disjuntor DJ-4. (MeasureAgent)
- Abrir Disjuntor DJ-3. (MeasureAgent)

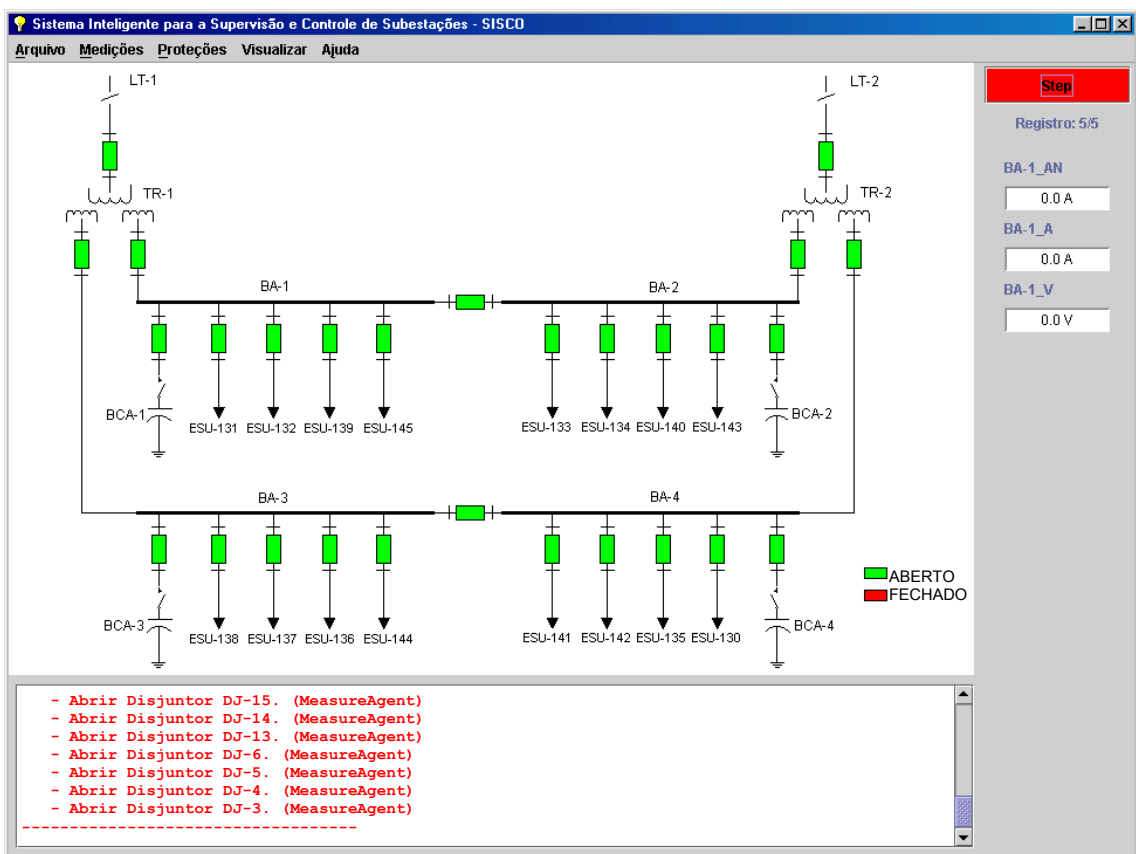


Figura 5.10 – Colapso total - Passo 3

Verificado o fim da falha inicia-se o processo de restabelecimento sugerindo-se a re-energização dos transformadores através do fechamento dos seus disjuntores, Figura 5.11. As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- ```

* Término da falha. Iniciando restabelecimento. Ajustar medições. *

- Recomenda-se restabelecer o Disjuntor DJ-1 e ajustar suas
medições.
- Recomenda-se restabelecer o Disjuntor DJ-2 e ajustar suas
medições.

```

- Recomenda-se restabelecer o Disjuntor DJ-3 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-5 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-4 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-6 e ajustar suas medições.
- 

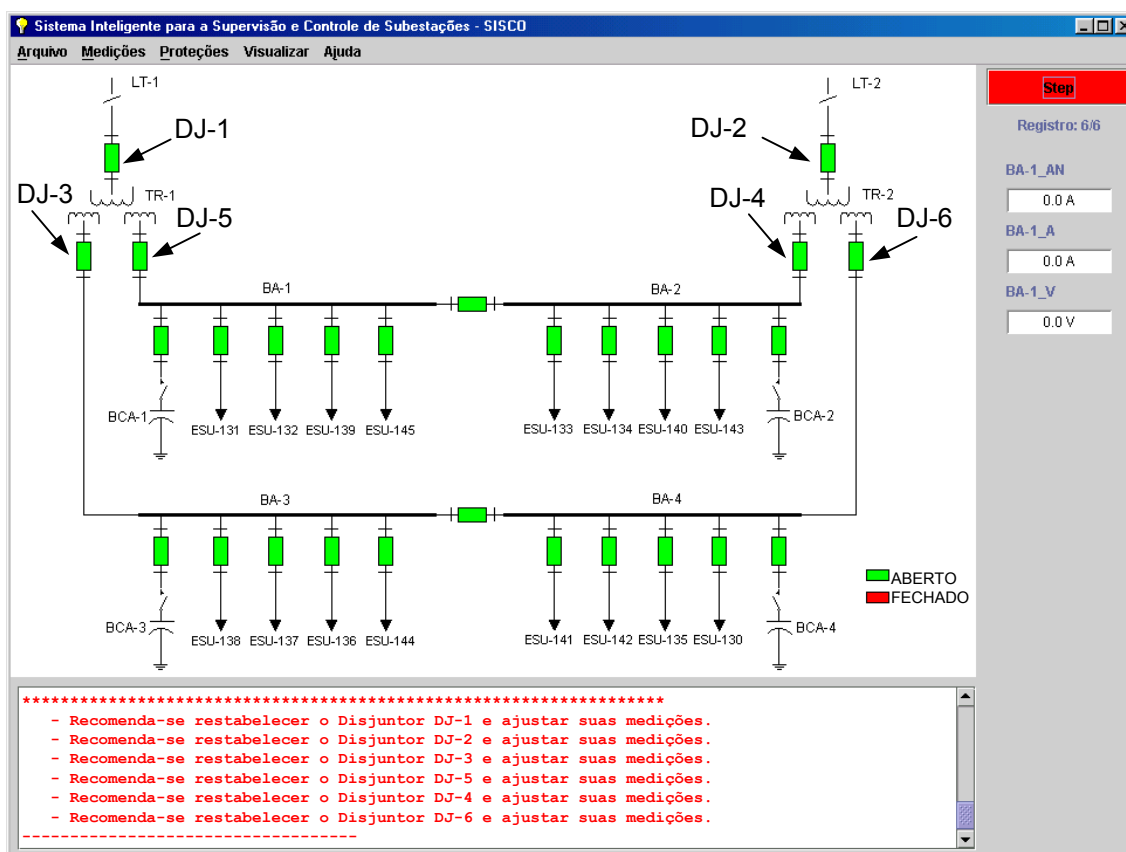


Figura 5.11 – Colapso total - Passo 4

O operador restabelece nesse passo os disjuntores DJ-1, DJ-2, DJ-3, DJ-5, DJ-4 e DJ-6, nesta ordem, re-energizando as barras da SE, Figura 5.12.

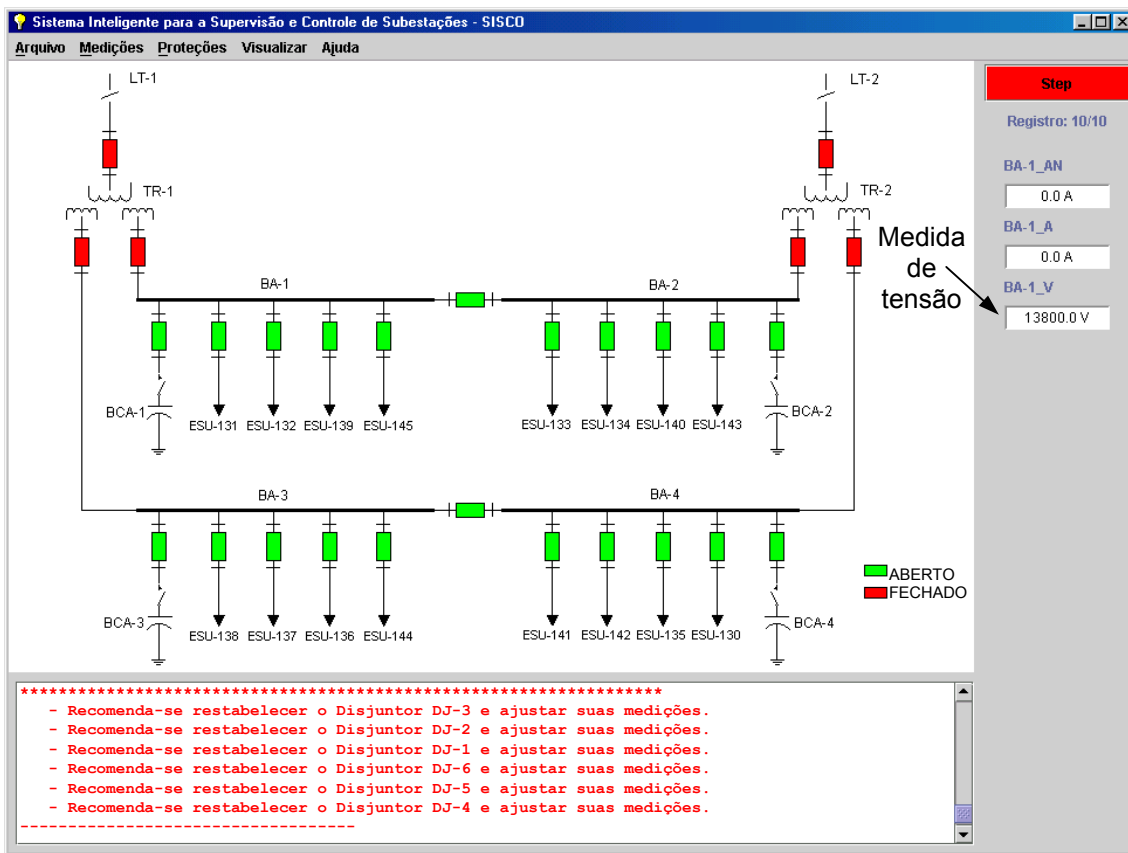


Figura 5.12 – Colapso total - Passo 5

Com a verificação de tensão nas barras, é sugerido o fechamento dos disjuntores dos alimentadores, Figura 5.13. As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:**
- Recomenda-se restabelecer o Disjuntor DJ-13 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-14 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-15 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-16 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-17 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-18 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-19 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-20 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-21 e ajustar suas medições.

- Recomenda-se restabelecer o Disjuntor DJ-22 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-23 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-24 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-25 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-26 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-27 e ajustar suas medições.
  - Recomenda-se restabelecer o Disjuntor DJ-28 e ajustar suas medições.
- 

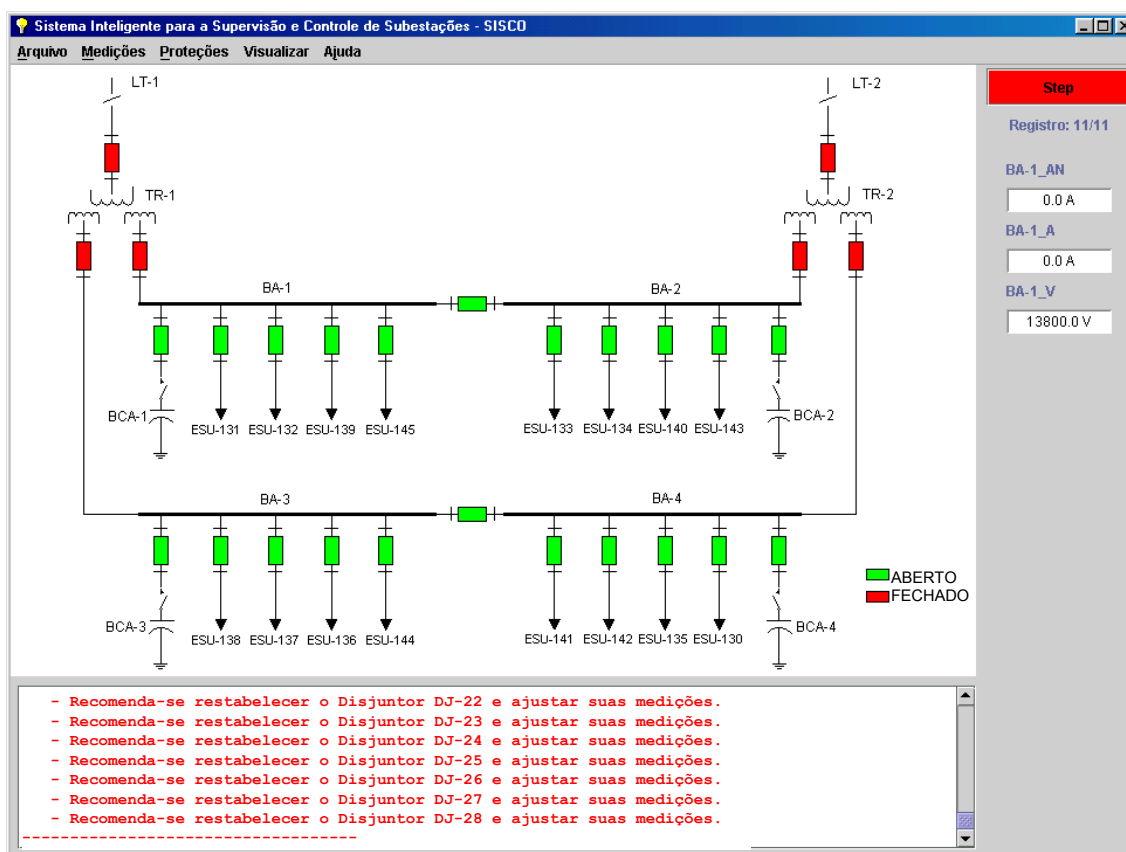


Figura 5.13 – Colapso total - Passo 6

O operador restabelece nesse passo os disjuntores DJ-13, DJ-14, DJ-15, DJ-16, DJ-17, DJ-18, DJ-19, DJ-20, DJ-21, DJ-22, DJ-23, DJ-24, DJ-25, DJ-26, DJ-27 e DJ-28, nesta ordem, re-energizando todos os alimentadores da SE, Figura 5.14.

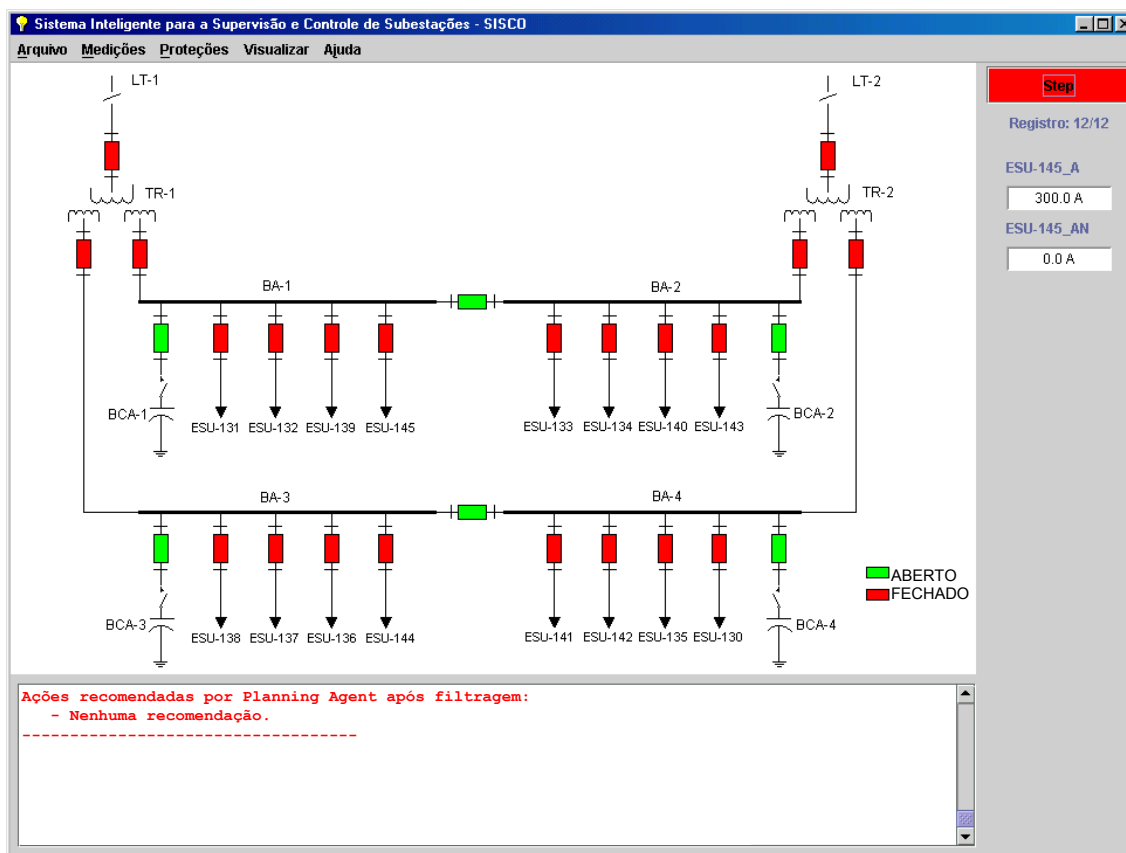


Figura 5.14 – Colapso total - Passo 7

### 5.3 Falha Permanente em Transformador

Foi simulada uma falha interna no transformador TR-1 que alimenta as barras BA-1 e BA-3, com a atuação da proteção de Buchholz TR-1\_63, Figura 5.15.

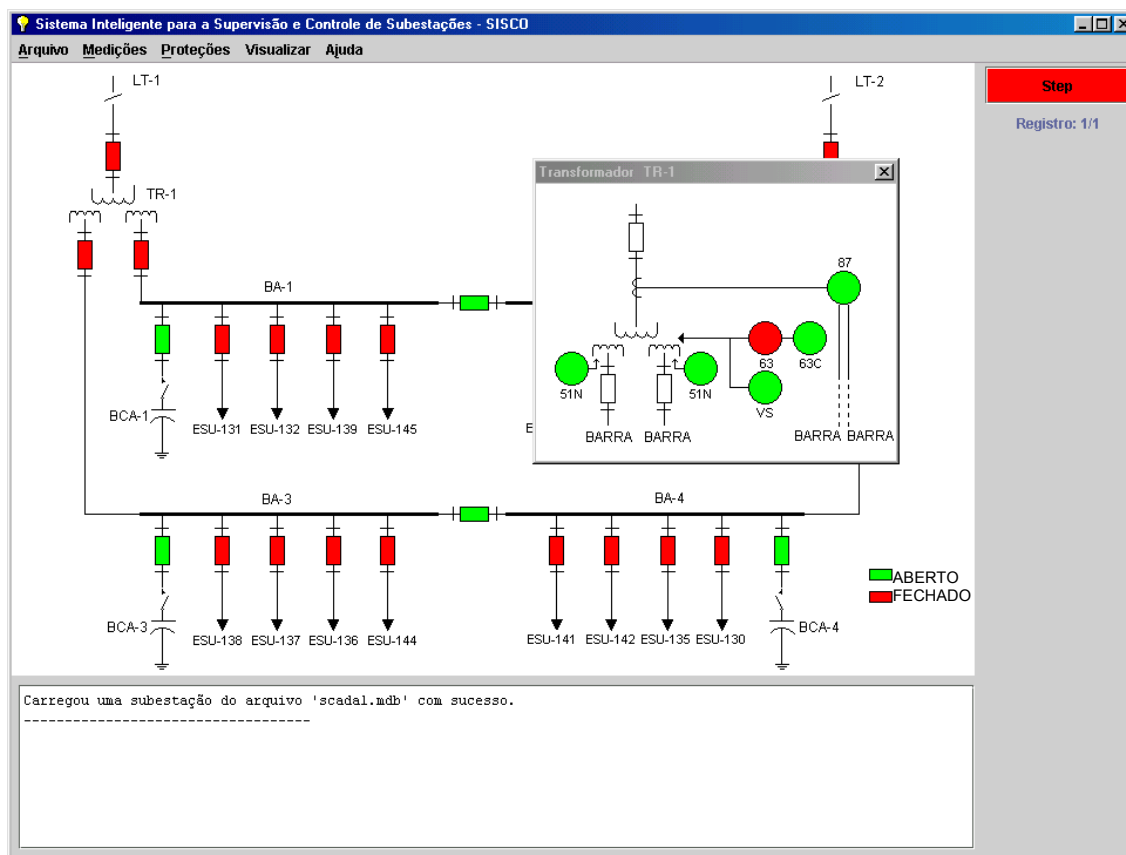


Figura 5.15 – Falha permanente em transformador - Passo 1

Com a atuação da proteção será sugerida a abertura dos disjuntores de alta e de baixa do transformador, desenergizando o equipamento e as barras BA-1 e BA-3, Figura 5.16.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:
- Abrir Disjuntor DJ-1. (ProtectionAgent)
  - Abrir Disjuntor DJ-3. (ProtectionAgent)
  - Abrir Disjuntor DJ-5. (ProtectionAgent)
-

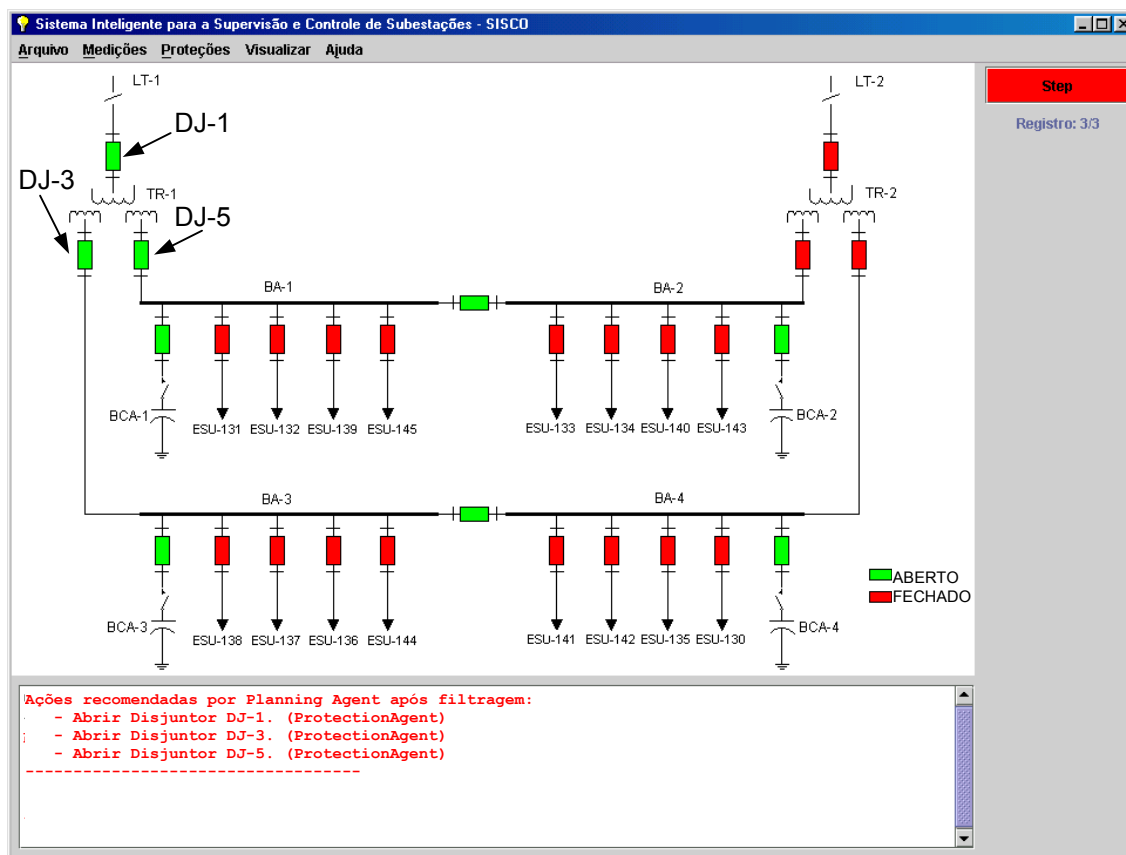


Figura 5.16 – Falha permanente em transformador - Passo 2

Conforme norma operativa da empresa, com a falta de tensão nas barras todos os disjuntores dos alimentadores ligados a elas receberão sugestão para sua abertura, Figura 5.17.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

- Ações recomendadas por Planning Agent após filtragem:
- Abrir Disjuntor DJ-13. (MeasureAgent)
  - Abrir Disjuntor DJ-14. (MeasureAgent)
  - Abrir Disjuntor DJ-15. (MeasureAgent)
  - Abrir Disjuntor DJ-16. (MeasureAgent)
  - Abrir Disjuntor DJ-21. (MeasureAgent)
  - Abrir Disjuntor DJ-22. (MeasureAgent)
  - Abrir Disjuntor DJ-23. (MeasureAgent)
  - Abrir Disjuntor DJ-24. (MeasureAgent)



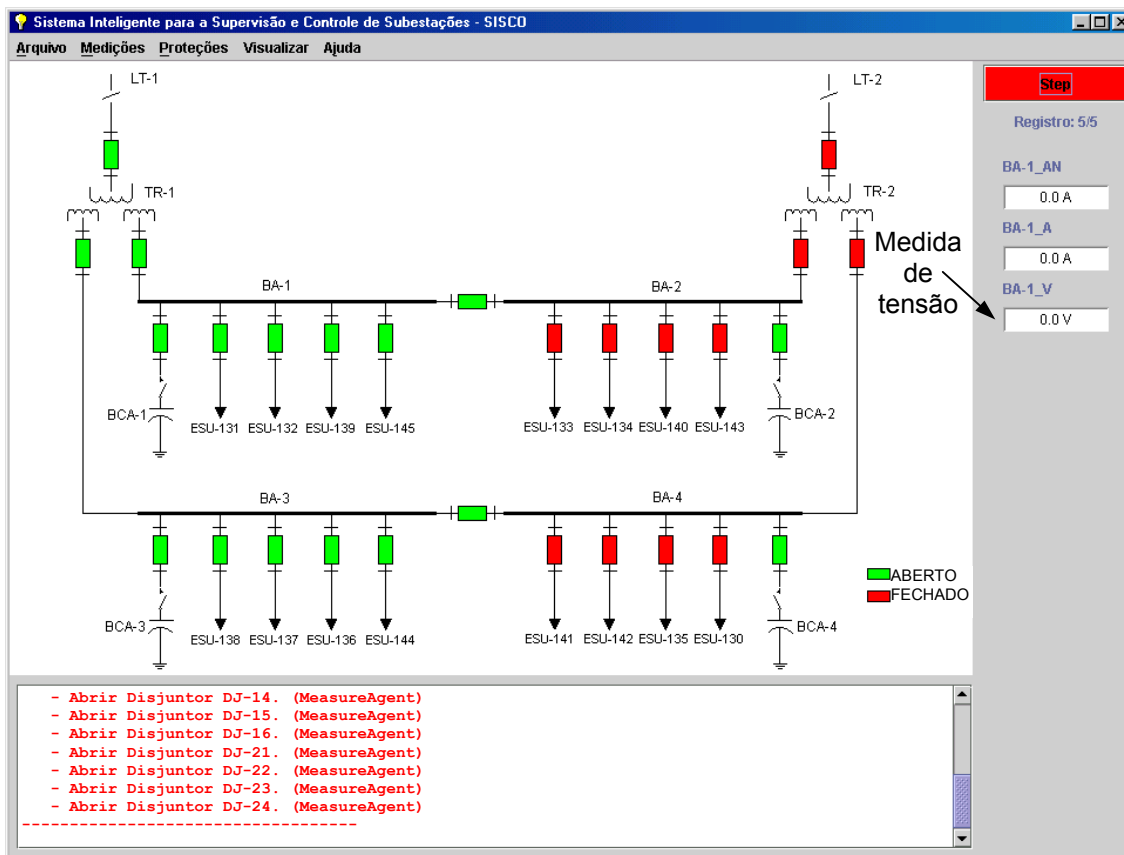


Figura 5.17 – Falha permanente em transformador - Passo 3

Como se trata de uma falha permanente, pois a proteção permanece atuada, inicia-se o processo de restabelecimento sugerindo-se a re-energização das barras através do fechamento dos seus disjuntores de paralelo, Figura 5.18. As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

```

* Término da falha. Iniciando restabelecimento. Ajustar medições. *

- Recomenda-se restabelecer o Disjuntor DJ-7 e ajustar suas
medições.
- Recomenda-se restabelecer o Disjuntor DJ-8 e ajustar suas
medições.

```

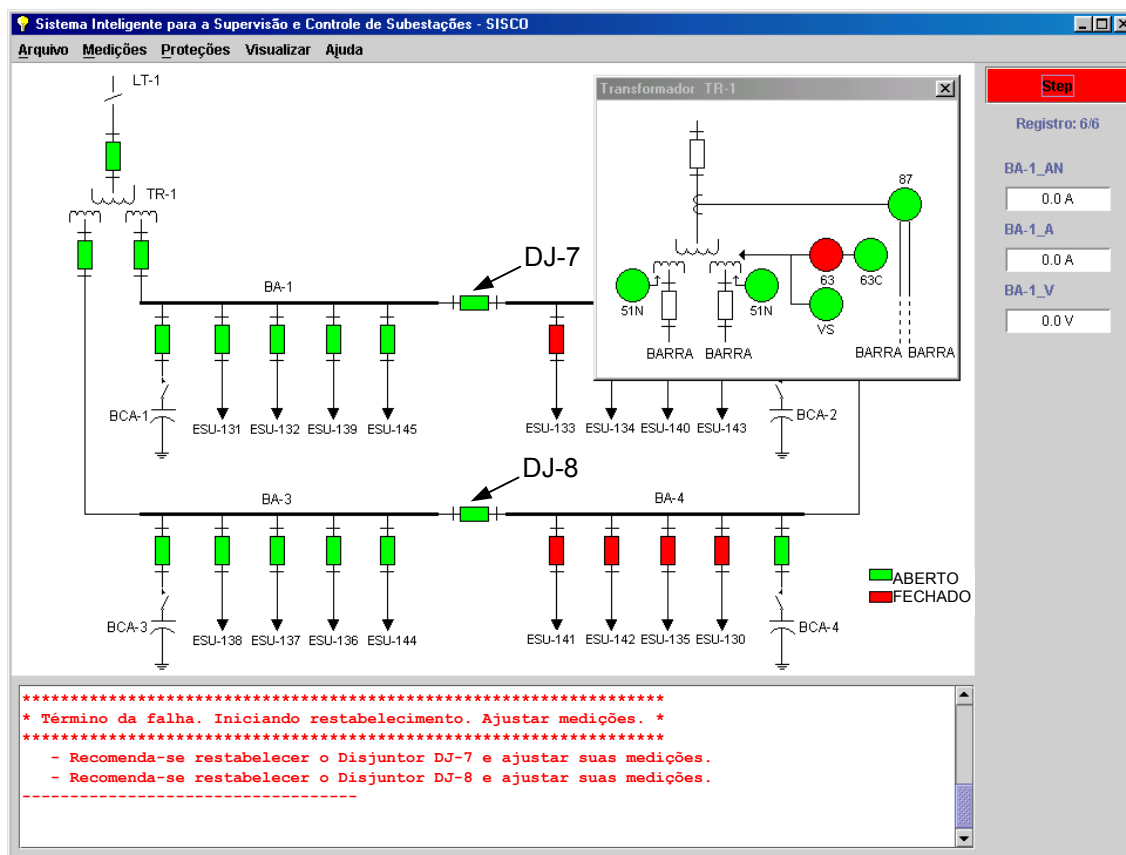


Figura 5.18 – Falha permanente em transformador - Passo 4

O operador restabelece nesse passo os disjuntores DJ-7 e DJ-8, nesta ordem, re-energizando as barras BA-1 e BA-3 da SE, Figura 5.19.

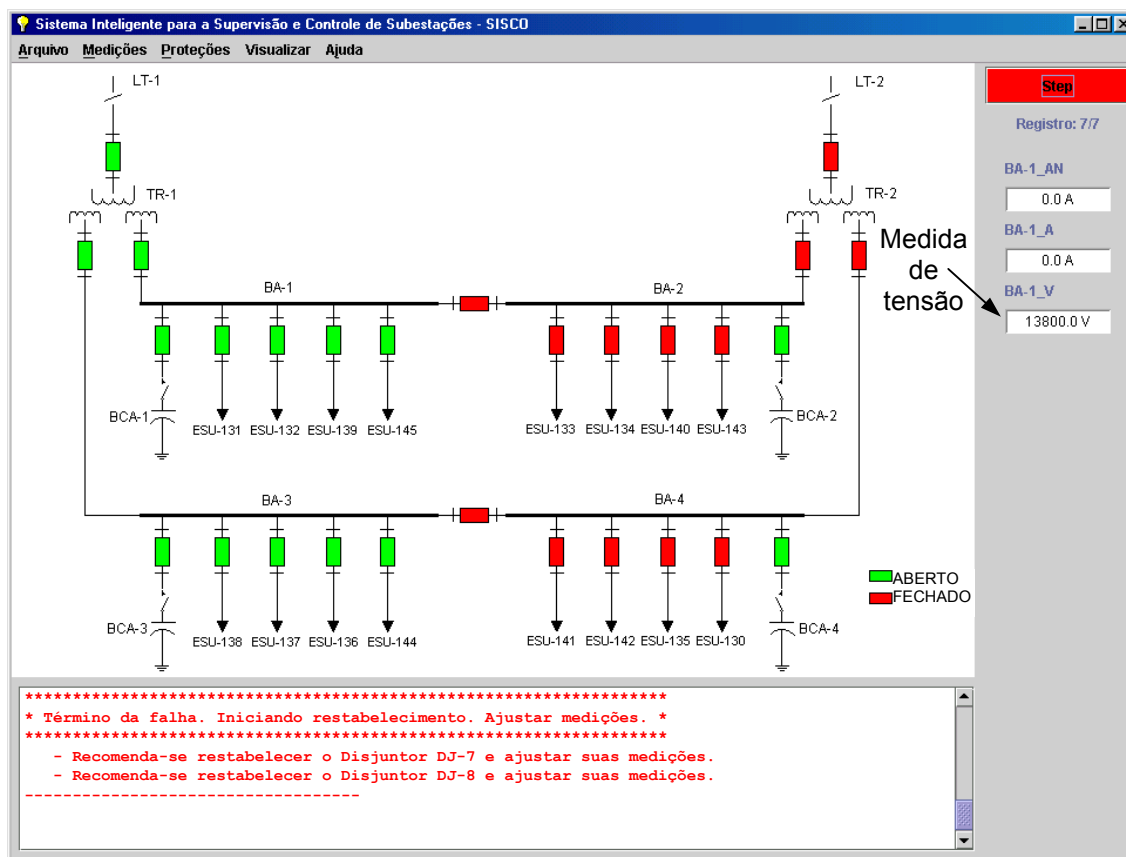


Figura 5.19 – Falha permanente em transformador - Passo 5

Com a verificação de tensão nas barras, é sugerido o fechamento dos disjuntores dos alimentadores ESU-131 e ESU-138, Figura 5.20. As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

**Ações recomendadas por Planning Agent após filtragem:**

- Recomenda-se restabelecer o Disjuntor DJ-13 e ajustar suas medições.
- Recomenda-se restabelecer o Disjuntor DJ-21 e ajustar suas medições.

-----

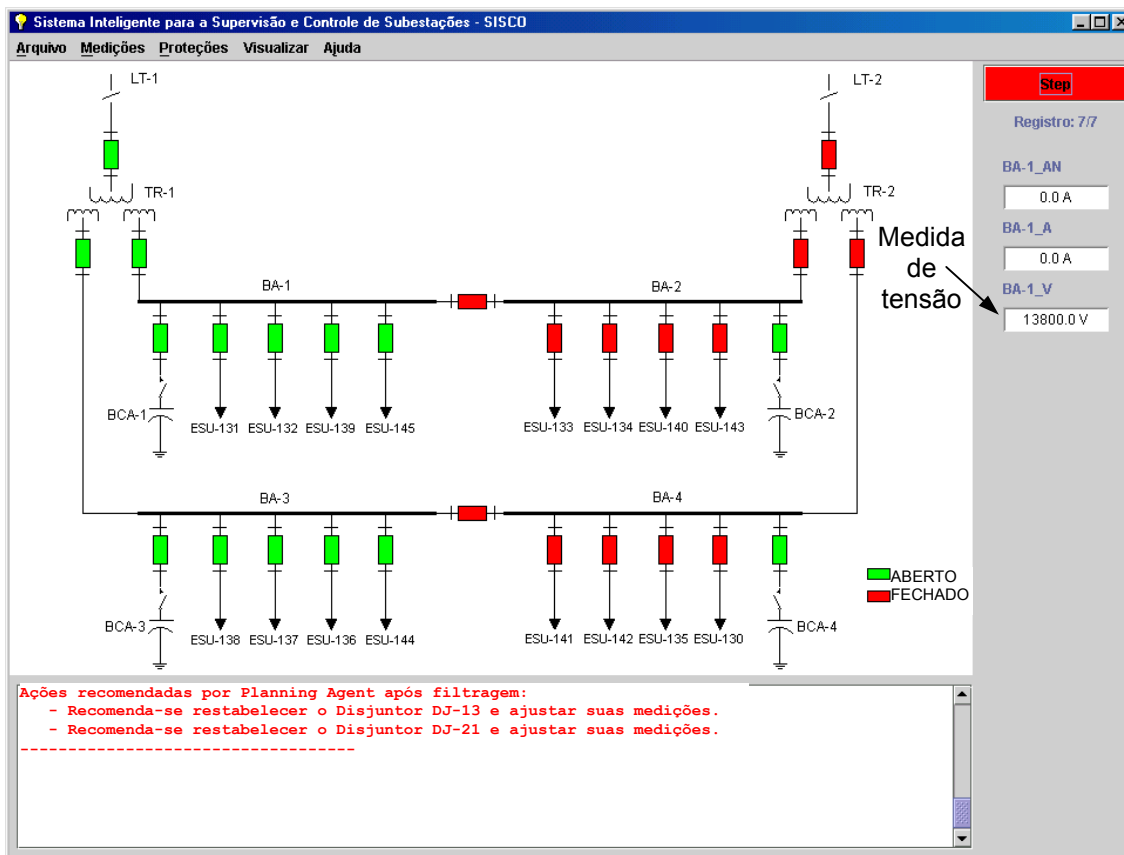


Figura 5.20 – Falha permanente em transformador - Passo 6

O operador restabelece nesse passo os disjuntores DJ-13 e DJ-21, nesta ordem, re-energizando os alimentadores da ESU-131 e ESU-138, Figura 5.21. As correntes nas barras BA-2 e BA-4 aumentam em 300 [A] cada, que é a corrente nos alimentadores re-energizados, mas ainda não atingiram a sua capacidade de transmissão que é de 2000 [A]. Sendo assim, será sugerido o fechamento de mais dois alimentadores, um em cada barra.

As ações sugeridas pelo “Agente Planejador” estão apresentadas abaixo:

Ações recomendadas por Planning Agent após filtragem:  
 - Recomenda-se restabelecer o Disjuntor DJ-14 e ajustar suas medições.  
 - Recomenda-se restabelecer o Disjuntor DJ-22 e ajustar suas medições.

-----

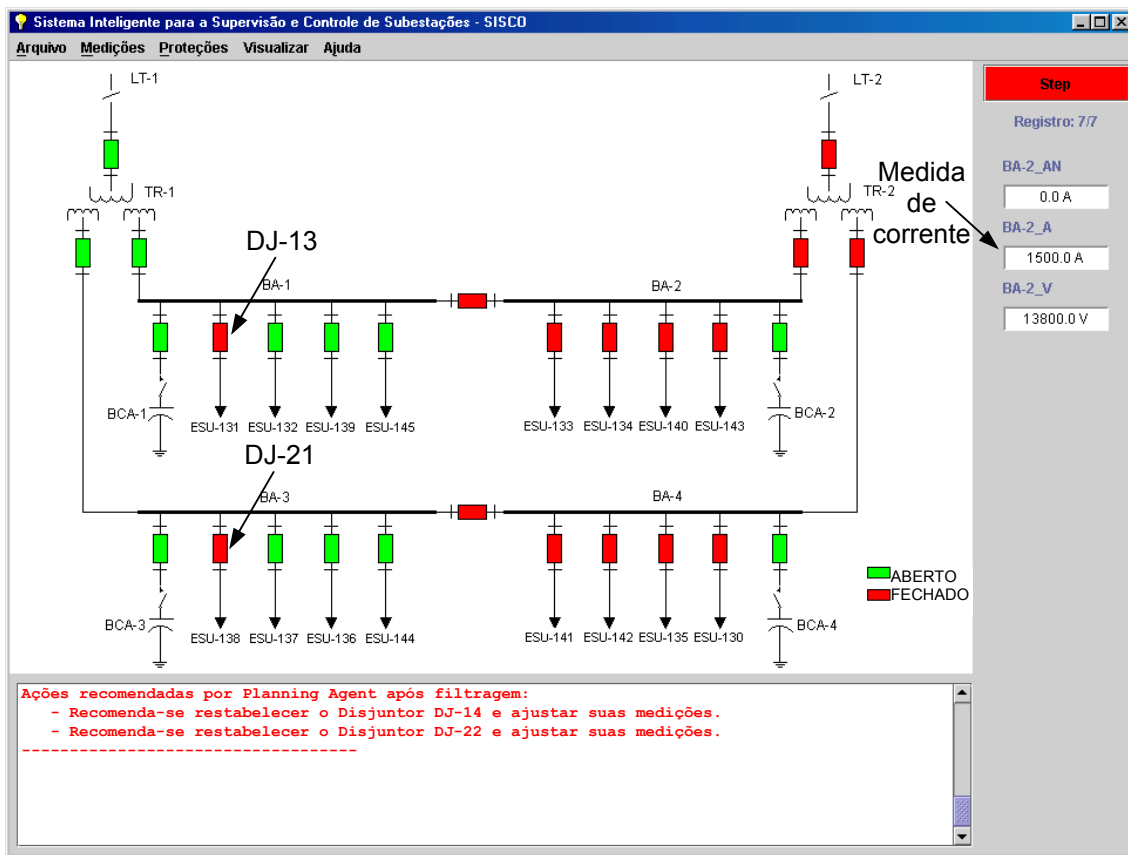


Figura 5.21 – Falha permanente em transformador - Passo 7

O operador restabelece nesse passo os disjuntores DJ-14 e DJ-22, nesta ordem, re-energizando os alimentadores da ESU-132 e ESU-137, Figura 5.22. As correntes nas barras BA-2 e BA-4 aumentam em 300 [A] cada, que é a corrente nos alimentadores re-energizados.

Ainda não foi atingida a capacidade de transmissão das barras, mas com a inclusão de mais dois alimentadores, um em cada barra, esse limite seria ultrapassado, portanto não há mais sugestões de ações de restabelecimento.

Ações recomendadas por Planning Agent após filtragem:  
 - Nenhuma recomendação.

-----

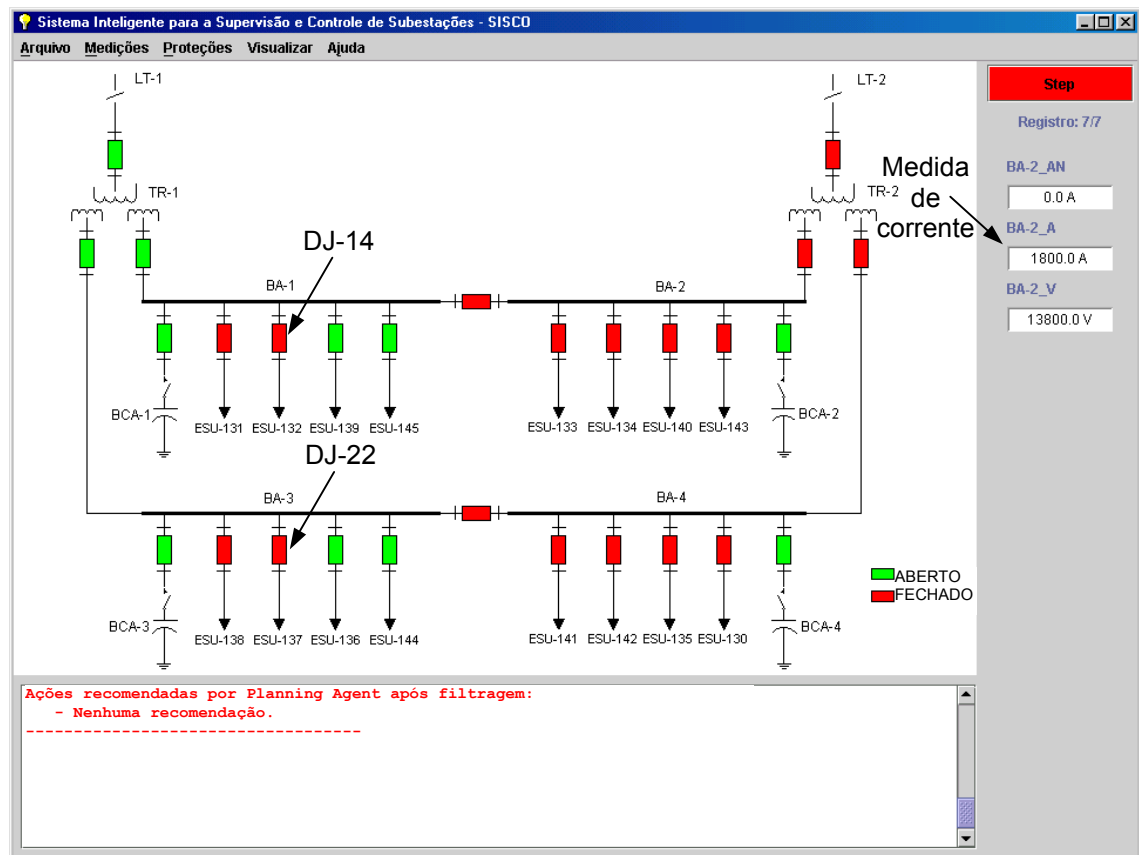


Figura 5.22 – Falha permanente em transformador - Passo 8

## 5.4 Falha Total em Alimentador de Distribuição

Para realização de testes no programa SISCO – Distribuição foram utilizados dados da CEB – Companhia Energética de Brasília, com os alimentadores da subestação que atende Taguatinga/DF: 0801, 0802, 0803, 0805, 0806, 0807, 0808, 0809, 0810, 0811 e 0813 (Figura 5.23), que possui as seguintes características:

- 3.731 Barras;
- 614 Transformadores;
- 25 Disjuntores;
- 1.118 Chaves;
- 43 Chaves de socorro;
- 1.904 linhas;
- Carga ativa de 7,387 MW e reativa de 2,386 MVar.

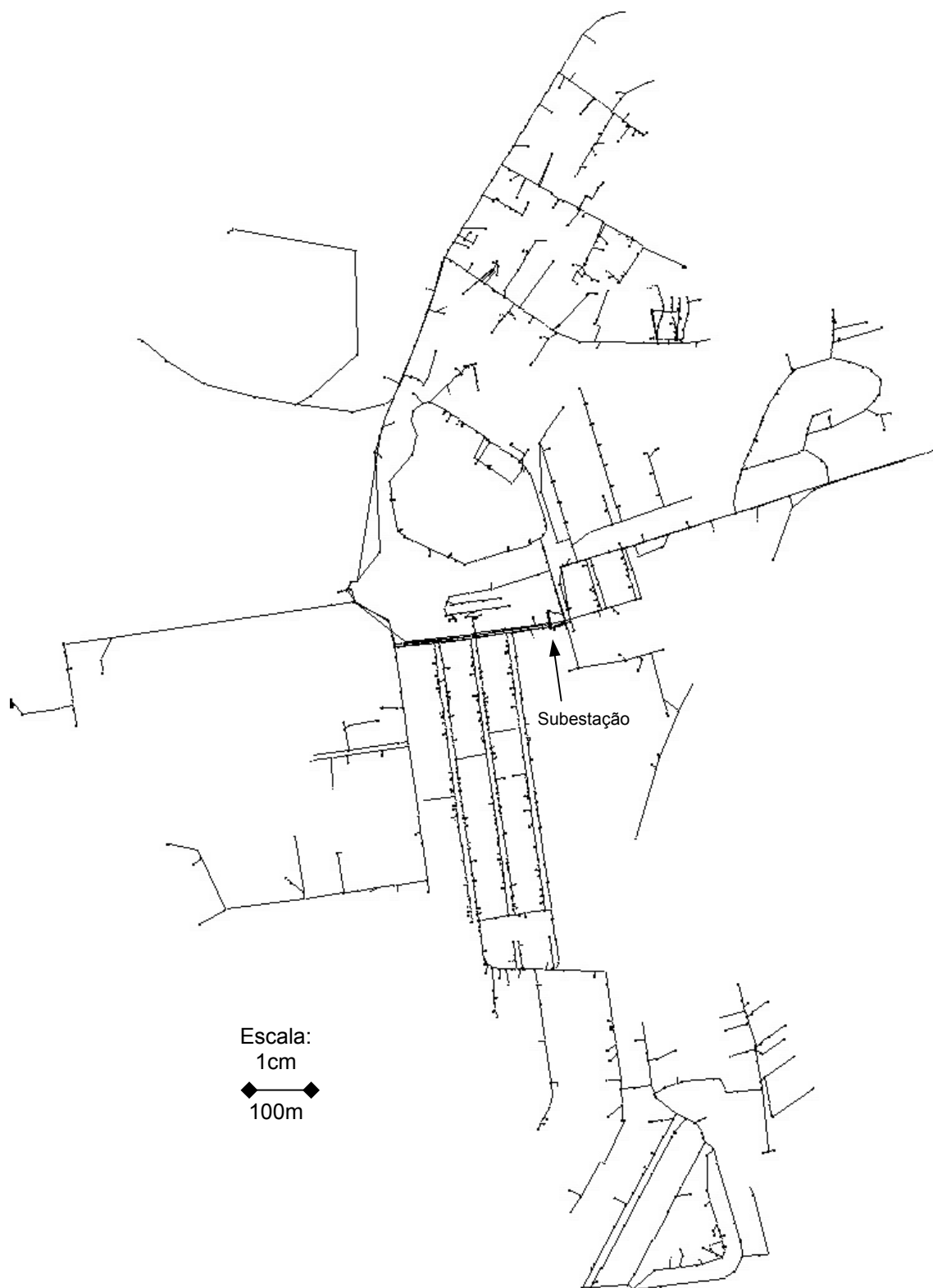


Figura 5.23 – Sistema de distribuição da CEB – Subestação de Taguatinga/DF

A simulação foi realizada para uma falha no disjuntor DI-0809 do alimentador 0809, deixando todo o ramal desligado, em vermelho na Figura 5.24.

Obteve-se um tempo médio de processamento equivalente a 15 segundos em um Pentium III 750 MHz com 128 MB de RAM. O tempo de processamento é importante, pois trata-se de um desligamento que requer uma ação imediata do operador. A solução encontrada está apresentada abaixo:

|                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Solução encontrada:</b></p> <p><b>BARRAS EM FALTA: 3</b><br/>(Sem conexão) Fechar FA-7730</p> <p><b>BARRAS EM FALTA: 306</b><br/>Fechar FA-8653<br/>    Subtensões: 240<br/>Fechar FA-7718<br/>    Subtensões: 278<br/>Fechar FA-1611<br/>    Subtensões: 276<br/>Fechar FA-1975<br/>    Sobrecargas: 1<br/>    Subtensões: 32</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Foram encontradas duas regiões sem energia, a primeira com 3 barras desenergizadas, e a segunda com 306 barras desenergizadas.

Na verdade a falha gerada é caracterizada pela segunda região desenergizada, sendo que o primeiro caso surge devido à sobreposição de dados do sistema, uma vez que essas três barras devem ser empregadas para conexão com outro alimentador, o qual não pertence à lista de dados passada para o programa.

No caso da primeira falha encontrada, o programa encontra uma solução para a reenergização (FA-7730), porém como não encontra dados suficientes para analisar a manobra é colocado um comentário "Sem Conexão" para indicar a falta de informação sobre a outra parte do sistema (do outro lado da chave).



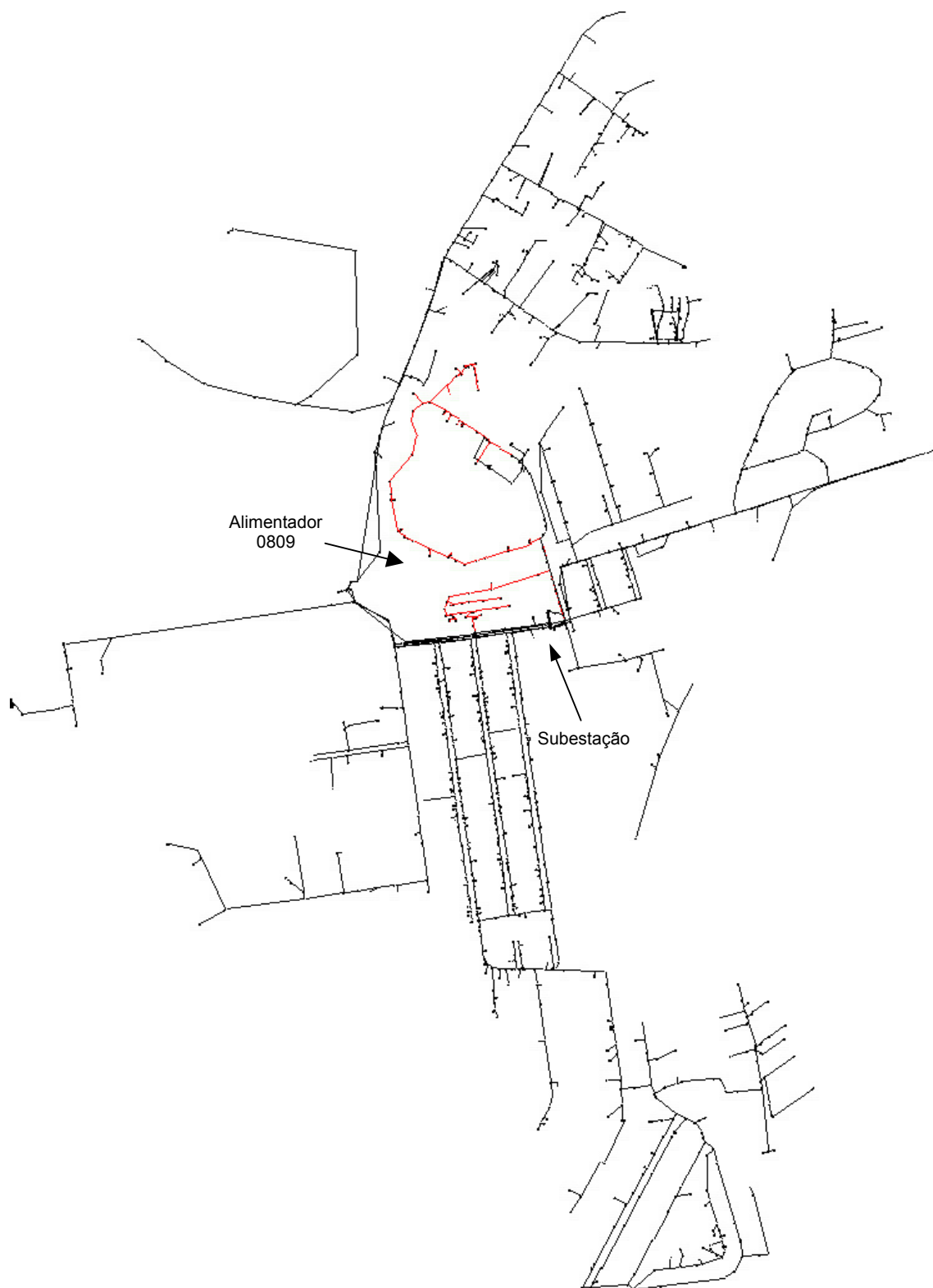


Figura 5.24 – Alimentador 0809

Agora, para a segunda região desenergizada que corresponde a falha simulada foram determinadas quatro possibilidades de manobras, ordenadas de acordo com a prioridade de escolha de cima para baixo, Figura 5.25. Para cada manobra é indicada a quantidade de barras que estarão com subtensões, ou a quantidade de linhas em sobrecarga. Pode-se ainda ajustar a saída de dados para apresentar uma listagem de todas as barras e linhas que apresentarem algum tipo de problema.

A primeira solução encontrada (FA-8653) é a chave de socorro externa que liga o circuito 0809 ao circuito 0808. A segunda solução (FA-7718) e a terceira (FA-1611) são chaves de socorro externas que ligam o circuito 0809 ao circuito 0813. Já a última solução (FA-1975) é uma chave de socorro externa que liga o circuito 0809 ao circuito 0806.

Este teste foi realizado com o acompanhamento de uma equipe técnica da CEB, a qual considerou bastante satisfatória a resposta sugerida, e ainda considerou também o número de manobras encontradas, bem como a qualidade das informações do sistema, suficientes para auxiliar na tomada de decisões dos operadores do sistema.

Essas soluções estão de acordo com as soluções sugeridas por operadores experientes da empresa.

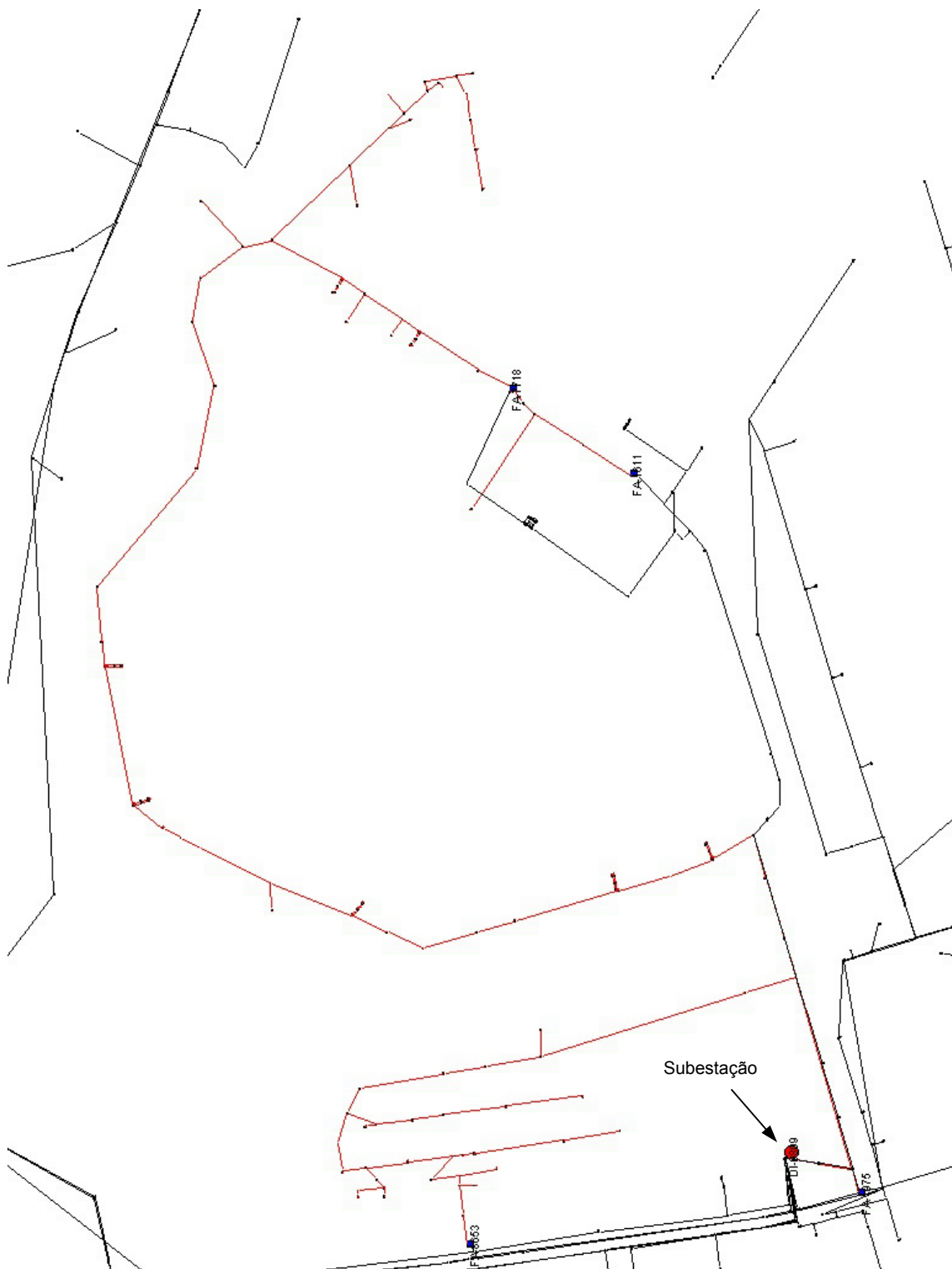


Figura 5.25 – Soluções para a falha total no alimentador

## 5.5 Falha Parcial em Alimentador de Distribuição

Nesse teste foi utilizado o circuito 0802 (em vermelho na Figura 5.26) e foi simulada uma falha no transformador DP-3309/0075, que é um transformador de distribuição de 75 kVA com o primário em delta (13,8 kV) e o secundário em estrela aterrada (380 V).

Essa ocorrência provoca a abertura da chave FA-7929 (destacada na Figura 5.27) desenergizando parcialmente o alimentador. Obteve-se um tempo médio de processamento equivalente a 12 segundos em um Pentium III 750 MHz com 128 MB de RAM. A solução encontrada está apresentada abaixo:

```
Solução encontrada:

BARRAS EM FALTA: 3
(Sem conexão) Fechar FA-7730

BARRAS EM FALTA: 94
Fechar FC-0015
 Sobrecargas: 5
 Subtensões: 809
Fechar FA-1607
 Sobrecargas: 5
 Subtensões: 809
Fechar FC-0011
 Sobrecargas: 5
 Subtensões: 809
```

Foram encontradas duas regiões sem energia, a primeira com 3 barras desenergizadas, e a segunda com 94 barras desenergizadas.

Mais uma vez o primeiro caso surge devido à sobreposição de dados do sistema, uma vez que essas três barras devem ser empregadas para conexão com outro alimentador, o qual não pertence à lista de dados passada para o programa.

Agora, para a segunda região desenergizada que corresponde a falha simulada foram determinadas três possibilidades de manobras, ordenadas de acordo com a prioridade de escolha de cima para baixo, Figura 5.27. As três soluções encontradas (FC-0015, FA-1607 e FC-0011) são chaves de socorro externo que liga o circuito 0802 ao circuito 0801. Essas soluções estão de acordo com as soluções sugeridas por operadores experientes da empresa.



Figura 5.26 – Alimentador 0802



Figura 5.27 – Soluções para falha parcial no alimentador

## 5.6 Implementação Real

O aplicativo SISCO – Distribuição foi implementado na Companhia Energética de Brasília (CEB). Esse *software* encontra-se hoje em uso no Centro de Operações da CEB, contando ainda com uma estrutura completa de suporte para integração com o Sistema de Apoio a Operação (SAO) e com o sistema SCADA.

A integração do sistema multi-agente (SMA) com o SAO está apresentada no fluxograma da Figura 5.28, neste é explicitado o caminho da informação, bem como o processamento empregado em cada parcela do conjunto de dados para a obtenção do resultado. Os processos em amarelo são desempenhados pelo SAO, enquanto que os em vermelho são executados pelo SMA.

O operador irá gerar uma solicitação de análise de reconfiguração através de uma interface apropriada no SAO da CEB. Essa solicitação é então transformada automaticamente em um arquivo texto que contém a chave em falha e os dados do sistema que devem ser utilizados na busca de soluções. Esse tipo de procedimento foi adotado por ser exatamente o mesmo procedimento já empregado pelos operadores da CEB para outros tipos de análise, como por exemplo cálculo de fluxo de potência.

O arquivo da solicitação será gravado num diretório compartilhado entre o SAO e o SMA, como os dois sistemas lêem e escrevem no mesmo diretório, não é permitida a realização de operações simultâneas de leitura/escrita pelos sistemas.

Para o controle da concorrência, visando evitar que o SAO e o SMA tentem acessar os arquivos de integração simultaneamente, foi implementado um método de exclusão mútua baseado na existência ou não de arquivos auxiliares: EAL.LOC (*External Application Link*) e SMA.LOC. Esses arquivos auxiliares são arquivos textos comuns, sem conteúdo, somente com o nome e a terminação mudados.

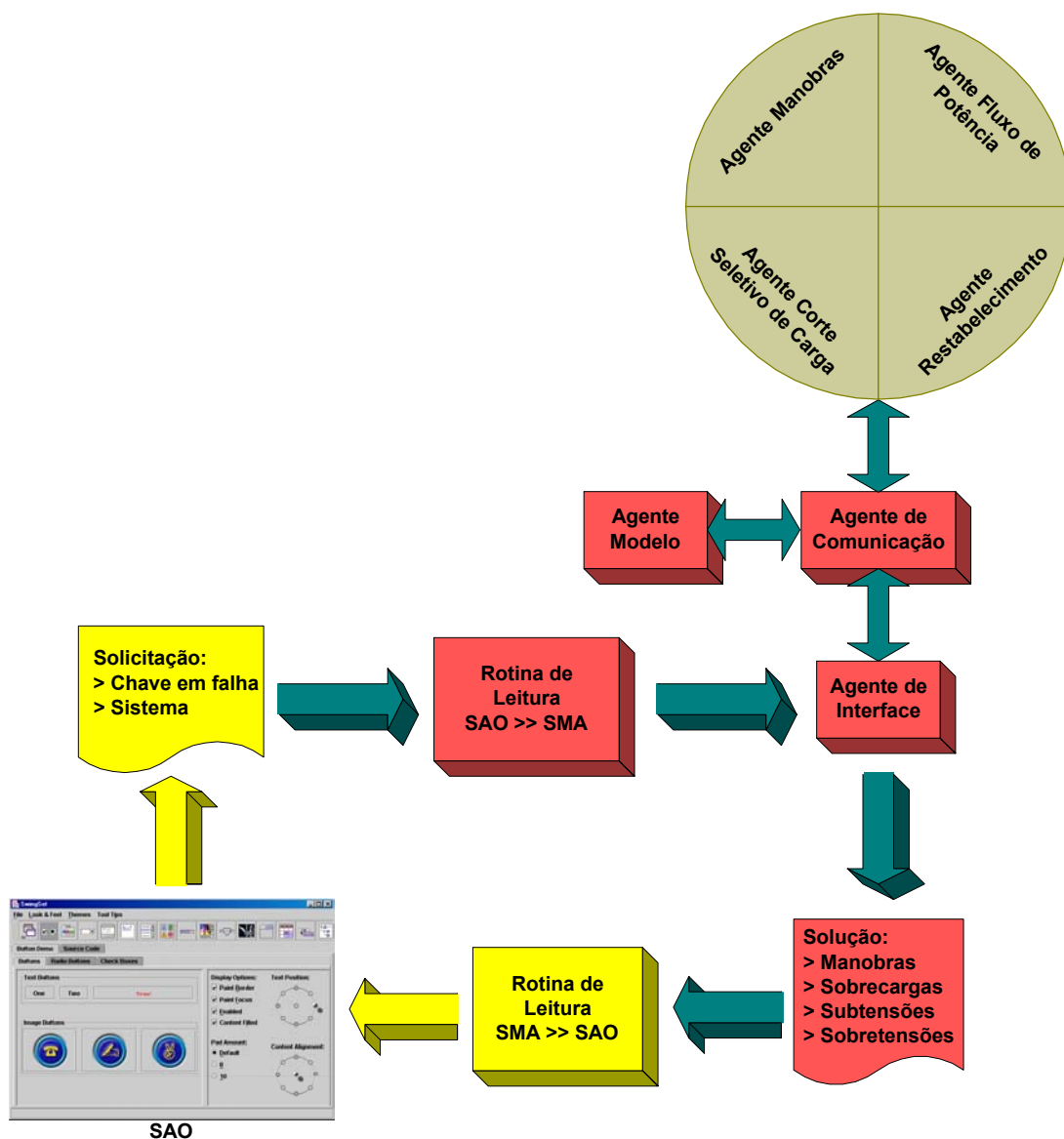
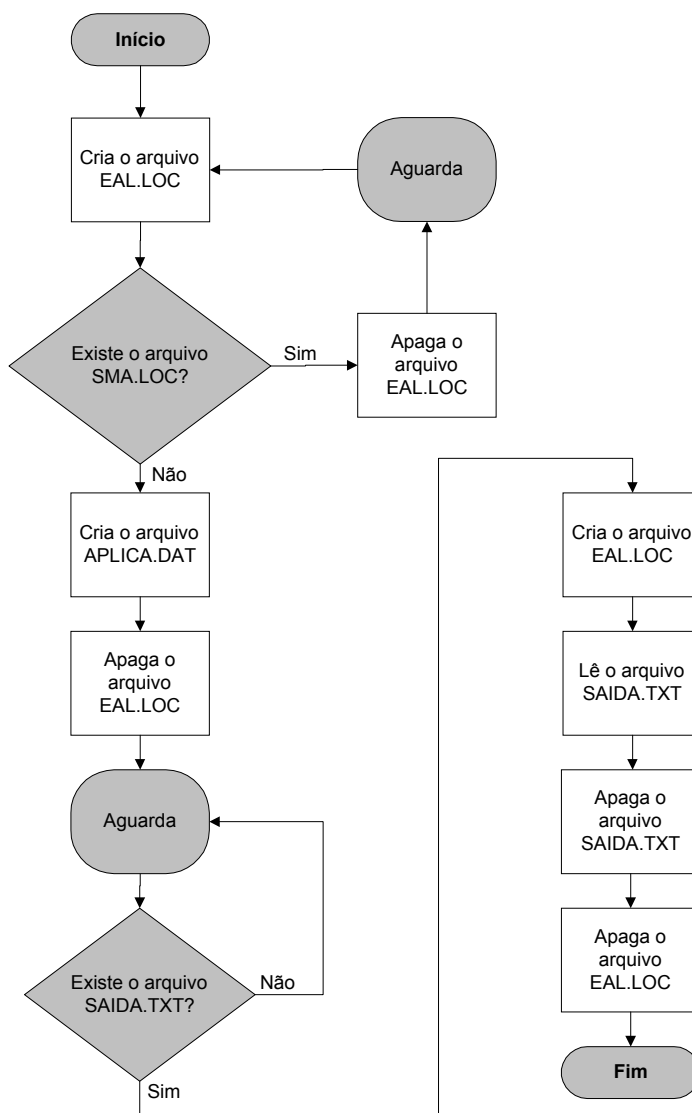


Figura 5.28 – Fluxograma de integração do SISCO – Distribuição / SAO

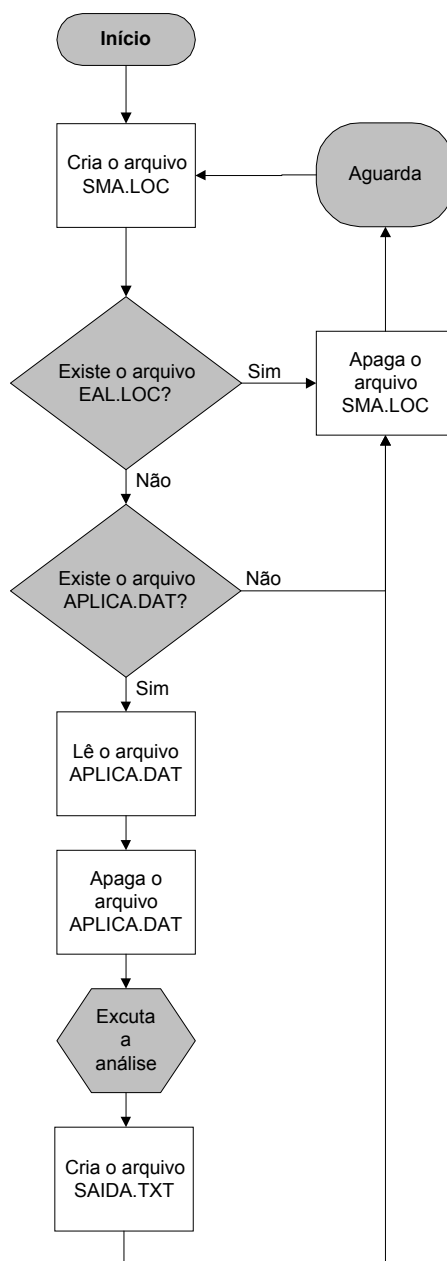
Na Figura 5.29 é apresentado o fluxograma de controle de concorrência do SAO. Primeiramente, o SAO cria o arquivo EAL.LOC e verifica se existe o SMA.LOC. Caso exista, isso significa que o SMA está escrevendo no diretório compartilhado, logo o SAO irá apagar o arquivo EAL.LOC e esperar um tempo determinado para repetir a operação. Caso contrário, o SAO irá escrever o arquivo APLICA.DAT, que é a solicitação de análise de reconfiguração, apagar o arquivo EAL.LOC e esperar pela resposta do SMA.





**Figura 5.29** – Fluxograma de Controle de Concorrência do SAO

Na Figura 5.30 é apresentado o fluxograma de controle de concorrência do SMA. Primeiramente, o SMA cria o arquivo SMA.LOC e verifica se existe o arquivo EAL.LOC. Caso exista, isso significa que o SAO está acessando o diretório compartilhado, logo o SMA irá apagar o arquivo SMA.LOC e esperar um tempo determinado para repetir a operação. Caso contrário, o SMA irá verificar se existe uma solicitação de análise de reconfiguração. Caso não exista um arquivo APLICA.DAT então o SMA irá apagar o arquivo SMA.LOC e esperar um tempo determinado para repetir a operação. Caso exista uma solicitação, o SMA irá ler o arquivo APLICA.DAT e apagá-lo, então será iniciado o processamento do SMA conforme apresentado no item 4.3.2.



**Figura 5.30** – Fluxograma de Controle de Concorrência SMA

Terminada a análise o SMA cria o arquivo SAIDA.TXT, apaga o SMA.LOC e aguarda um tempo determinado para repetir a operação.

Uma vez que o SMA escreveu o arquivo SAIDA.TXT no diretório compartilhado o SAO entra em operação novamente para realizar a leitura desse arquivo e apresentar a solução para o operador numa interface apropriada desenvolvida no próprio SAO, Figura 5.29.

Paralelamente a essa implementação foi desenvolvido outro projeto intitulado “Desenvolvimento de Sistema Inteligente de Locomoção de Viaturas para Atendimento da Rede de Distribuição da CEB”, o qual tinha por objetivo a confecção de um *software* inteligente que colaborasse com o operador na tarefa de determinar qual a melhor viatura para atender a uma ocorrência na rede de distribuição.

Esse aplicativo foi desenvolvido dentro do âmbito desse trabalho de doutorado utilizando a mesma arquitetura multi-agente proposta nessa tese e sua configuração é a mesma apresentada no item 3.3, considerando-se ao invés de táxis, viaturas da companhia, e ainda, ao invés de uma pessoa solicitando um táxi tem-se a chave a ser manobrada. Mais detalhes da implementação computacional desse aplicativo estão contidos no Anexo E.

O procedimento operacional global de integração dos dois aplicativos com o SAO está apresentado na Figura 5.31.

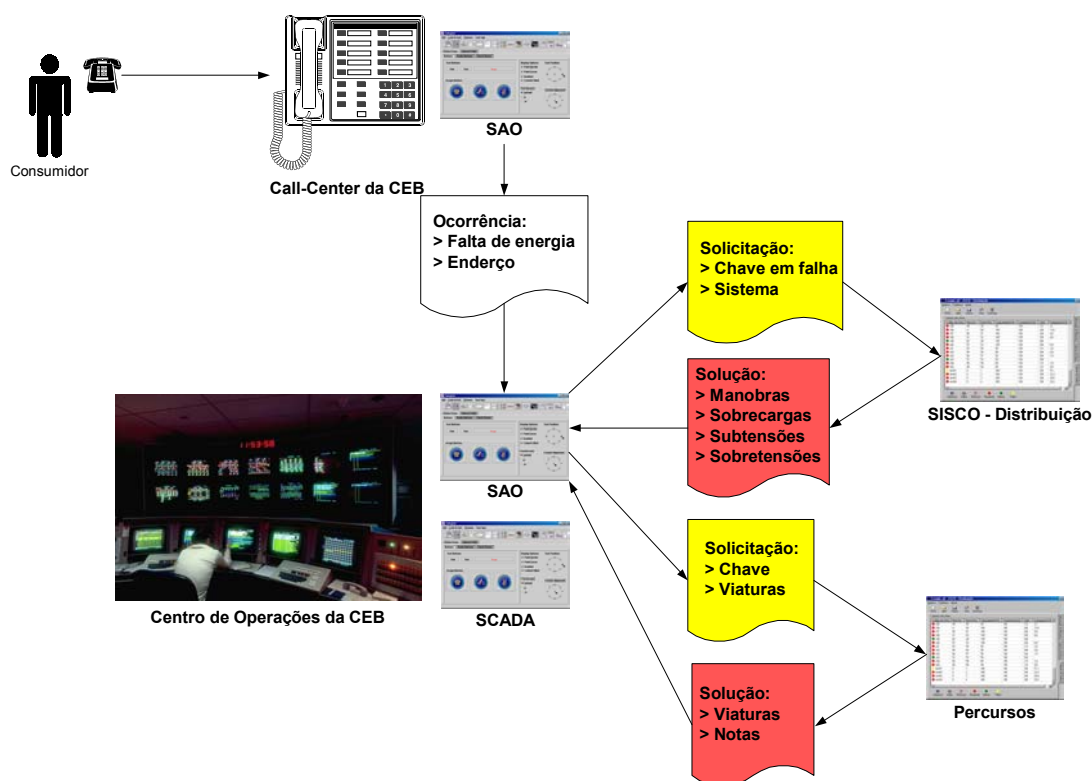


Figura 5.31 – Fluxograma de integração global dos sistemas

O consumidor irá acionar o *Call-Center* da CEB quando houver uma falha no fornecimento de energia elétrica, então o atendente do *Call-Center* irá

registrar essa ocorrência em um dos aplicativos do SAO. Essa ocorrência é enviada para o Centro de Operações e um operador irá solicitar ao SISCO – Distribuição que analise a falha e determine quais são as possíveis manobras e suas conseqüências. O operador irá selecionar qual manobra deseja realizar e acionar o programa “Percurso” para analisar qual a melhor via para efetuar essa manobra.

# 6 CONCLUSÕES

---

## 6.1 Resultados

Esta tese iniciou com a apresentação de duas áreas da ciência, Inteligência Artificial e Vida Artificial, as quais estão bastante relacionadas entre si, porém possuem diferenças fundamentais no desenvolvimento de aplicações práticas. Em seguida, foram definidos: o problema a ser resolvido e o objetivo deste trabalho.

Percebe-se que a aplicação de IA na solução de problemas relativos aos sistemas elétricos é uma área recente e que desperta muito interesse na comunidade científica e nas empresas do setor, haja visto a evolução dos trabalhos desenvolvidos e dos eventos realizados pelos seus pesquisadores.

O problema proposto para esta pesquisa foi: **“Desenvolver uma estrutura computacional capaz de integrar diversas aplicações de IA para sistemas elétricos possibilitando a realização de análises globais e a troca de informações entre as diversas partes do sistema.”**

Após analisar as alternativas de tecnologias e soluções existentes, escolheu-se a seguinte: **“Desenvolver uma arquitetura de sistema multi-agente baseada na estrutura dos sistemas elétricos e nas necessidades específicas de cada parte do sistema.”**

No processo de construção dessa arquitetura encontrou-se um primeiro problema de representação das entidades físicas num modelo computacional. Para superar tal dificuldade, foi desenvolvido um modelo orientado a objeto do sistema elétrico visando um grau de abstração de forma a permitir o reaproveitamento em outras plantas independente de sua topologia.

Esse modelo se constitui em uma das contribuições dessa pesquisa, porém não é o foco desse trabalho. O SMA mostrou-se adequado para trabalhar com o modelo orientado a objeto, dadas as características de distribuição de ambas as tecnologias.

A modelagem orientada a objeto oferece ao sistema benefícios como:

- **Representação distribuída:** permitindo ao modelo virtual a reprodução das propriedades distribuídas do sistema elétrico.
- **Flexibilidade:** dando ao modelo virtual adaptabilidade às alterações que podem ocorrer como mudanças topológicas no sistema elétrico.
- **Reaproveitamento:** utilizando o modelo desenvolvido para a representação de outras subestações, através da redefinição dos dispositivos e acréscimo ou mudança de propriedades.
- **Arquitetura aberta:** permitindo acréscimo de componentes e a expansão do modelo como parte de um sistema maior.

A principal contribuição desse trabalho é a arquitetura multi-agente proposta, a qual foi baseada na implementação de dois pacotes de agentes. O primeiro composto por “Agentes de Suporte”, responsáveis por manter o modelo orientado a objetos atualizado, prover o sistema multi-agente de uma estrutura de comunicação e de meios de acessar e enviar informações para o mundo real. O segundo pacote, formado por “Agentes Especializados”, os quais sintetizam o conhecimento específico necessário para desenvolver as análises no domínio da aplicação.

Essa arquitetura emprega o modo de comunicação de sistemas confederados, o qual utiliza um agente como o centralizador do processo comunicativo, facilitando a distribuição dos agentes em unidades computacionais diferentes, bem como a inserção de novos agentes no SMA.

O pacote de “Agentes de Suporte” pode ser empregado para diversas aplicações diferentes e permite a integração dessas implementações, uma vez que todas estão baseadas no mesmo procedimento computacional. Essa é uma das vantagens da arquitetura proposta nessa tese, outra vantagem é o uso de agentes

inteligentes, os quais permitem a implementação de qualquer técnica internamente, seja de IA ou não, possibilitando o reaproveitamento de programas antigos.

Os principais benefícios alcançados com a arquitetura multi-agente proposta foram:

- **Comportamento cooperativo:** é caracterizado por trocas dinâmicas de informação entre os agentes e possibilitando a divisão de tarefas, reduzindo as características hierárquicas do processo de decisão.
- **Processamento concorrente:** melhora o uso da capacidade de hardware e diminui o tempo de resposta.
- **Topologia distribuída:** o que torna possível no processo de abstração a divisão do sistema em um número menor de partes, limitado pelas características funcionais.
- **Arquitetura aberta:** permitindo a adição de componentes e a expansão do sistema, como parte de um sistema maior com a criação de novos níveis de análise.

Baseadas nessa arquitetura proposta foram desenvolvidas duas aplicações: uma para subestações elétricas e outra para sistemas de distribuição.

O programa SISCO – Subestações possui uma lógica de raciocínio que permite a mudança da topologia da subestação em estudo sem grandes desenvolvimentos para colocá-lo novamente em operação. Basicamente, seria necessário construir uma nova base de dados (baseada no modelo OO) para a nova subestação e (se for o caso) adicionar algumas particularidades de manobras de operação referentes àquela topologia.

Esse programa pode funcionar de forma automática e conectado ao sistema SCADA. Essa aplicação foi concebida, estruturada e implementada de forma modular, a fim de garantir flexibilidade quanto à expansão e implementação de modificações operacionais ou novas topologias. A abordagem modular, característica de sistemas multi-agente, permite a inclusão de novos agentes e a expansão do sistema como parte de um sistema maior, mantendo as características de distribuição do mundo real.

O processo de raciocínio empregado é baseado na concorrência entre quatro agentes: “Agente Chaves”, “Agente Medidas”, “Agente Equipamentos” e “Agente Proteções”. Cada agente sugere ações de operação baseado na sua visão e conhecimento do mundo real. O “Agente Planejador” funciona como o árbitro desse processo decidindo e ordenando as ações que serão encaminhadas para o operador.

Os principais benefícios alcançados com a implantação de um programa como o SISCO - Subestações são:

- Serve como ferramenta de treinamento para os novos operadores;
- Ajuda a manter o conhecimento e a experiência da equipe técnica de forma estruturada;
- Auxílio na tomada de decisões do operador, aumentando a sua eficiência na execução de tarefas;
- Rapidez na definição e implementação das ações de restabelecimento;
- Contribui para a diminuição do tempo de interrupção do fornecimento de energia.

Já o programa SISCO – Distribuição foi desenvolvido sobre a mesma plataforma do anterior, inclusive empregando o mesmo modelo orientado a objetos. No entanto, o processo de raciocínio desenvolvido foi diferente, sendo que nesse programa não houve a necessidade de se utilizar o “Agente Planejador”. O papel de “Agente Planejador” ficou a cargo do “Agente de Restabelecimento”, o qual gerencia todo o processo de busca de soluções.

Como principais benefícios da implantação de um programa como o SISCO – Distribuição podem-se citar:

- A padronização dos procedimentos de reconfiguração da distribuição, já que o programa sempre apresenta as mesmas alternativas de resposta para a mesma situação do sistema, cabendo ao operador escolher a melhor opção no momento.
- Esta padronização de procedimentos e a análise automática da reconfiguração do sistema, tende a diminuir o tempo de reparo



da rede primária e a minimizar problemas de erros operativos, aumentando assim a segurança operativa do sistema.

O programa poderá também ser utilizado pelos despachantes para simular a reconfiguração do sistema visando futuras condições operativas, como concentração de carga em certos ramais ou desligamentos para manutenções programadas. Isto leva a redução da interrupção de energia dos clientes.

Ambos os programas não fornecem subsídios somente para as tarefas de restabelecimento, pois outras manobras podem ser verificadas, como por exemplo:

- Manobras para manutenção de um certo equipamento de uma subestação ou sistema de distribuição podem ser testadas bastando para isso simular o desligamento desse equipamento;
- Testar a flexibilidade de manobrar o sistema de distribuição em determinados horários de pico, ou considerando um aumento de carga, bastando para isso criar uma cópia do sistema e alterar a carga em determinadas barras e simular falhas comuns do sistema;
- Testar a flexibilidade de manobrar sistemas de distribuição ou subestações frente a uma expansão de suas topologias.

Por fim, no âmbito do desenvolvimento desse projeto de Doutorado foram realizadas diversas publicações, as quais estão listadas no Anexo A.

Esse tipo de aplicativo proporcionará um benefício direto às Concessionárias através da correta monitoração e controle de subestações e do sistema de distribuição, pois isto faz com que os equipamentos funcionem dentro de sua capacidade nominal, contribuindo para a diminuição da possibilidade de ocorrência de defeitos e de faltas de energia elétrica.

Outro aspecto importante está ligado a duração das faltas de energia devido as manobras nas subestações e nos sistemas de distribuição, pois com o plano de restabelecimento gerado automaticamente pelos aplicativos desenvolvidos, o tempo de restabelecimento do sistema tenderá a diminuir.

E ainda, esse tipo de aplicativo poderá contribuir com outros benefícios para as Concessionárias de Energia, como: a redução do seu custo operacional nas manobras do sistema, a melhoria da qualidade do fornecimento de energia elétrica, o aumento na qualidade dos serviços prestados ao consumidor, a melhor administração da carga e a possibilidade de guardar o conhecimento dos operadores mais antigos.

## 6.2 Desenvolvimentos Futuros

Para que esse trabalho de pesquisa possa ter continuidade no futuro, o autor sugere que seja implementada a proposta de integração das duas ferramentas apresentada no item 4.4. Para tal, será necessário estudar quais as informações relevantes que devem ser passadas de um SMA para o outro, e mais importante ainda como e quando essas informações devem ser trocadas.

A integração dos dois aplicativos visa consolidar a potencialidade da arquitetura multi-agente proposta. Dessa forma, novos desenvolvimentos poderão empregar a mesma estrutura e se beneficiar das vantagens inerentes dos SMA.

Pode-se ainda, implementar a arquitetura proposta para outras topologias de subestações, e ainda, desenvolver um programa com uma visão macro de sistema elétrico de potência que comandasse a operação desses dois aplicativos já desenvolvidos.

Por fim, essa tese apresenta o primeiro esforço de modelagem de um sistema multi-agente aplicado em sistemas elétricos, vislumbra-se como o próximo grande passo, o desenvolvimento de um modelo que consiga refletir o comportamento de todos os centros de operação trabalhando simultaneamente. Mais do que sistemas inteligentes funcionando em cada centro de operação, será necessário representar o comportamento inteligente dos operadores como um todo. Essa tarefa envolverá mais conhecimentos da área da vida artificial, cujas técnicas poderão ser aplicadas na tentativa de vencer esse desafio.

## REFERÊNCIAS

ALVES DA SILVA, A. P.; INSFRAN, A. H. F.; DA SILVEIRA, P. M. & LAMBERT-TORRES, G. Neural Networks for Fault Location in Substations. **IEEE Transactions on Power Delivery**, v. 11, n. 1, p. 234-239, January 1996.

AMIN, M. Toward Self-Healing Energy Infrastructure Systems. **IEEE Computer Applications in Power**, v. 14, n. 1, p. 20-28, January 2001.

AOKI, A. R.; LAMBERT TORRES, G.; DE SOUZA L. E. & DE MORAES, C. H. V. Solving Power Substation Restoration using Intelligent Planning. In: IASTED International Conference on Power and Energy Systems, 1999. **Proceedings...** Las Vegas: p. 224-228.

ADIBI, M. M. **Power System Restoration: Methodologies & Implementations Strategies**. IEEE Press Power Engineering Series, 2000. 704 p.

AUSTIN, J. L. **How To Do Things With Words, 2<sup>nd</sup> Edition**. Harvard University Press, 1988. 176 p.

AZEVEDO, G. P.; FEIJÓ, B.; COSTA, M. Control Centers Evolve with Agent Technology. **IEEE Computer Applications in Power**, v. 13, n. 3, p. 48-53, July 2000.

BANN, J.; IRISARRI, G.; KIRSCHEN, D.; MILLER, B. & MOKHTARI, S. Integration of Artificial Intelligence Applications in the EMS: Issues and Solutions. **IEEE Transactions on Power Systems**, v. 11, n. 1, p. 475-482, February 1996.

BIGUS, J. P.; BIGUS, J. **Constructing Intelligent Agents With Java: A Programmer's Guide to Smarter Applications**. New York: Wiley Computer Publishing, 1997. 379 p.

BOOCH, G. **Object-Oriented Analysis and Design with Applications, 3<sup>rd</sup> Edition**. MA: Addison Wesley. 1999. 704 p.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML – Guia do Usuário**. Rio de Janeiro: Campus, 2000. 472 p.

CHANG, C.-K.; LEE, S.-J.; PARK, J.-K.; LEE, H.-J. Application of Multi-Agent System for Over-Current Protection System of Industrial Power System. In: International Conference on Intelligent System Application to Power System (ISAP 2001), 2001. **Proceedings...** Budapest, Hungary: DYNAdata Ltd., p. 73-77.

COHEN, P. R.; LEVESQUE, H. J. Communicative Actions for Artificial Agents. In: International Conference on Multi-Agent Systems (ICMAS-95), 1., 1995. **Proceedings...** San Francisco: AAI Press, p. 65-72.

CORKILL, D. D.; GAALLAGHAR, K. O.; MURRAY, D. E. GBB: A Generic Blackboard Development System. In: National Conference in Artificial Intelligence (AAAI-86), 5., 1986. **Proceedings...** Philadelphia: AAI Press, p. 1008-1014.

DAVIS, R.; SMITH, R. G. Negotiating as a Metaphor for Distributed Problem Solving. **Artificial Intelligence**, 20(1), p. 63-109, January 1983.

DURFEE, E. H.; MONTGOMERY, T. A. MICE: A Flexible Testbed for Intelligent Coordination Experiments. In: Distributed Artificial Intelligence Workshop, 1989. **Proceedings...** p. 25-40.

ESMIN, A. A. A.; AOKI, A. R.; LAMBERT-TORRES, G. Particle Swarm Optimization For Fuzzy Membership Functions Optimization. In: 2002 IEEE International Conference on System, Man and Cybernetics, 2002,. **Proceedings...** Hammamet, Tunísia: ITSS (Information Technology Support Systems).

ETZIONI, O.; WELD, D. Intelligent Agents on the Internet: Fact, Fiction, and Forecast. **IEEE Expert**, p. 44-49, August 1995.

EXTERNAL INTERFACES WORKING GROUP ARPA KNOWLEDGE SHARING EFFORT. **Specification of the KQML Agent-Communication Language**. Working Paper, June 1993.

FERBER, J. **Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence**. Addison-Wesley, 1999. 509 p.

FININ, T.; FRITZSON, R.; MCKAY, D.; MCENTIRE, R. KQML as an Agent Communication Language. In: International Conference on Information and Knowledge Management (CIKM'94), 3., 1994a. **Proceedings...** Gaithersburg, Maryland: ACM Press, p. 456-463.

FININ, T.; FRITZSON, R.; MCKAY, D.; MCENTIRE, R. KQML – A Language and Protocol for Knowledge and Information Exchange. In: FUCHI, K.; YOKOI, T. (Eds.). **Knowledge Building and Knowledge Sharing**. IOS Press, 1994b. p. 249-262.

FIPA – FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. **PC00089D: FIPA Domains and Policies Specification**. August 10, 2001a.

FIPA – FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. **XC00061E: FIPA ACL Message Structure Specification**. August 10, 2001b.

FIPA – FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. **XC00007BE: FIPA Content Language Library Specification**. August 10, 2001c.

GASSER, L.; BOND, A. H. **Readings in Distributed Artificial Intelligence**. San Mateo: Morgan Kaufmann, 1988.

GASSER, L. Social Conceptions of Knowledge and Action: DAI Foundations and Open Systems. **Artificial Intelligence**, 47(1-3), p. 107-138, January 1991.

HATVANY, J. Intelligence and Cooperation in Heterarchical Manufacturing Systems. In: CIRP International Seminar on Manufacturing Systems, 16., 1984. **Proceedings...** Tokyo. p. 1-4.

HEWITT, C. Viewing Control Structures as Patterns of Passing Messages. **Artificial Intelligence**, 8(3), p. 323-364, June 1977.

HOGG, L. M. J.; JENNINGS, N. R. Socially Intelligent Reasoning for Autonomous Agents. **IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans**, v. 31, n. 5, p. 381-393, September 2001.

JENNINGS, N. R. Specification and Implementation of a Belief Desire Joint-Intention Architecture for Collaborative Problem Solving. **Journal of Intelligent and Cooperative Information System**, 2(3), p. 289-318, 1993.

JENNINGS, N. R. Coordination Techniques for Distributed AI. In: O'HARE, G. M. P.; JENNINGS, N. R. (Eds.). **Foundations of Distributed Artificial Intelligence**. Wiley, John & Sons, 1996. p. 187-210.

KENNEDY, J.; EBERHART, R. C. **Swarm Intelligence**. San Francisco: Morgan Kaufmann, 2001. 512 p.

LABROU, Y.; FININ, T. A Semantics Approach for KQML – A General Purpose Communication Language for Software Agents. In: International Conference on Information and Knowledge Management (CIKM'94), 3., 1994. **Proceedings...** Gaithersburg, Maryland: ACM Press, p.447-455

LAMBERT-TORRES, G.; RIBEIRO, G. M.; COSTA, C. I. A.; ALVES DA SILVA, A. P.; QUINTANA, V. H. Knowledge Engineering Tool for Training Power-Substation Operators. **IEEE Transactions on Power Systems**, v. 12, n. 2, p. 694-699, April 1997.

MANGINA, E. E.; MCARTHUR, S. D. J.; MCDONALD, J. R. Multi Agent System Knowledge Representation for Power Plant Condition Monitoring Based on Modal Logic. In: International Conference on Intelligent System Application to Power System (ISAP 2001), 2001. **Proceedings...** Budapest, Hungary: DYNAdat Ltd., p. 67-72.

MEYER, B. **Object Oriented Software Construction, 2<sup>nd</sup> Edition**. New Jersey: Prentice Hall, 1997. 1254 p.

MULLER, J. P. A Conceptual Model for Agent Interaction. In: International Working Conference on Cooperative Knowledge Based Systems (CKBS'94), 2., 1994. **Proceedings...** University of Keele: Dake Centre, p. 213-233.

NII, H. P. Part One: Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. **AI Magazine**, 7(2), p. 38-53, Summer 1986.

PANDIT, S.; SOMAN, S. A.; KHAPARDE, S. A. Object-Oriented Design for Power System Applications. **IEEE Computer Applications in Power**, v. 13, n. 4, p. 43-47, October 2000.

PANDIT, S.; SOMAN, S. A.; KHAPARDE, S. A. Object-Oriented Network Topology Processor. **IEEE Computer Applications in Power**, v. 14, n. 2, p. 42-46, April 2001.

PARUNAK, H. V. D. Applications of Distributed Artificial Intelligence in Industry. In: O'HARE, G. M. P.; JENNINGS, N. R. (Eds.). **Foundations of Distributed Artificial Intelligence**. Wiley, John & Sons, 1996. p. 139-168.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. New Jersey: Prentice-Hall, 1995. 932 p.

SAKAGUCHI, T.; MATSUMOTO, K. Development of a Knowledge Based System for Power System Restoration. **IEEE Transactions on Power Apparatus and Systems**, v. PAS-102, n. 2, p. 320-329, February 1983.

SANZ-BOBI, M. A.; VILLAR, J.; SÁNCHEZ-ÚBEDA, E. F.; PLAZA, G.; ROBLEDANO, L. F.; REVUELTA, S.; KAZI, A. Multi-Agent Environment for Intelligent Diagnosis in Power Systems. In: International Conference on Intelligent System Application to Power System (ISAP 2001), 2001. **Proceedings...** Budapest, Hungary: DYNAdata Ltd., p. 61-66.

SEARLE, J. R. **Speech Acts – An Essay in Philosophy of Language**. Cambridge University Press, 1990. 212 p.

SMITH, R. G. The Contract Net Protocol: High Level Communication and Control in Distributed Problem Solver. **IEEE Transactions on Computers**, C-29(12), p. 1104-1113, 1980.

STEELS, L. Cooperation Between Distributed Agents Through Self-Organisation. In: European Workshop on Modeling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89), 1., 1990. **Proceedings...** Amsterdam: Elsevier Science Publishers B.V. p. 175-196.

SYCARA, K. P. Multiagent systems. **AI Magazine**, 19(2), p. 79-92, Summer 1998.

VALIQUETTE, B.; LAMBERT-TORRES, G.; MUKHEDKAR, D. An Expert System Based Diagnosis and Advisor Tool for Teaching Power System Operation Emergency Control Strategies. **IEEE Transactions on Power Systems**, 6(3), p. 1315-1322, 1991.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent Agents: Theory and Practice. **The Knowledge Engineering Review**, 10(2), p. 115-152, 1995.

ZHU, J.; LUBKEMAN, D. Object-Oriented Development fo Software for Power System Simulations. **IEEE Transactions on Power Systems**, p. 1002-1008, 1997.



# ANEXO A

## PUBLICAÇÕES REALIZADAS

### A.1 Artigos em Congressos

1. AOKI, Alexandre Rasi; ESMIN, Ahmed Ali Abdalla; LAMBERT-TORRES, Germano. Multi-Agent System for Distribution System Operation. In: 3º Congresso de Lógica Aplicada à Tecnologia, 2002, São Paulo/Brasil. **Proceedings of the 3º LAPTEC 2002**. 2002.
2. ESMIN, Ahmed Ali Abdalla; AOKI, Alexandre Rasi; LAMBERT-TORRES, Germano. Particle Swarm Optimization For Fuzzy Membership Functions Optimization. In: 2002 IEEE International Conference on System, Man and Cybernetics, 2002, Hammamet/Tunisia. **Proceedings of the IEEE SMC 2002**. 2002.
3. ESMIN, Ahmed Ali Abdalla; AOKI, Alexandre Rasi; LAMBERT-TORRES, Germano. Particle Swarm Optimization versus Genetic Algorithms for Fitting Fuzzy Membership Functions. In: 2002 WSEAS International Conference on SYSTEMS SCIENCE (ICOSYS'02), 2002, Rio de Janeiro/Brasil. **Proceedings of the WSEAS ICOSYS'02**. 2002.
4. ESMIN, Ahmed Ali Abdalla; AOKI, Alexandre Rasi; LOPES JÚNIOR, Claudio Ribeiro; LAMBERT-TORRES, Germano. Multi-Agent Simulation and Educational Tool for Power System Operation. In: VII INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY EDUCATION, 2002, Santos/SP. **Proceedings of the INTERTECH 2002**. 2002.
5. LOPES JÚNIOR, Claudio Ribeiro; AOKI, Alexandre Rasi; ESMIN, Ahmed Ali Abdalla; LAMBERT-TORRES, Germano. Multi-Agent Model for Power Substation Restoration. In: IASTED INTERNATIONAL CONFERENCE ON POWER AND ENERGY SYSTEMS, 2001, Tampa/Florida. **Proceedings of the IASTED PES 2001**. 2001.

6. ESMIN, Ahmed Ali Abdalla; AOKI, Alexandre Rasi; LOPES JÚNIOR, Claudio Ribeiro; LAMBERT-TORRES, Germano. Multi-Agent Model for Power System Simulation. In: WSES INTERNATIONAL CONFERENCE ON SIMULATION, 2001, **Proceedings of the WSES SIM 2001**. World Scientific and Engineering Society Press, 2001. p. 2531-2535.

## A.2 Capítulos de Livro

1. AOKI, Alexandre Rasi; ESMIN, Ahmed Ali Abdalla; LAMBERT-TORRES, Germano. Multi-Agent System for Distribution System Operation. In: ABE, J. M.; SILVA FILHO, J. I. (Eds.). **Advances in Logic, Artificial Intelligence and Robotics**. IOS Press: Amsterdam, 2002. p. 38-45.
2. ESMIN, Ahmed Ali Abdalla; AOKI, Alexandre Rasi; LOPES JÚNIOR, Claudio Ribeiro; LAMBERT-TORRES, Germano. Multi-Agent Model for Power System Simulation. In: D'ATTELLIS, C. E.; KLUEV, V. V.; MASTORAKIS, N. E. (Org.). **Mathematics and Simulation with Biological, Economical and Musicoacoustical Applications**. WSES Press, 2001. p. 29-33.

# ANEXO B

## ESPECIFICAÇÕES DA FIPA

A Foundation for Intelligent Physical Agents – FIPA foi formada em 1996 para produzir padrões de software para agentes heterogêneos e interativos e para sistemas baseados em agentes.

Na produção desses padrões, FIPA solicita dados e colaboração dos seus membros e da comunidade científica de agentes em geral para construir especificações que possam ser utilizadas para alcançar interoperabilidade entre sistemas baseados em agentes desenvolvidos por diferentes companhias e organizações.

A FIPA é organizada e estruturada de acordo com dois grupos; aqueles envolvidos na produção e desenvolvimento de padrões e aqueles envolvidos em mecanismos de suporte da FIPA.

As especificações da FIPA estão divididas conforme Figura B.1.

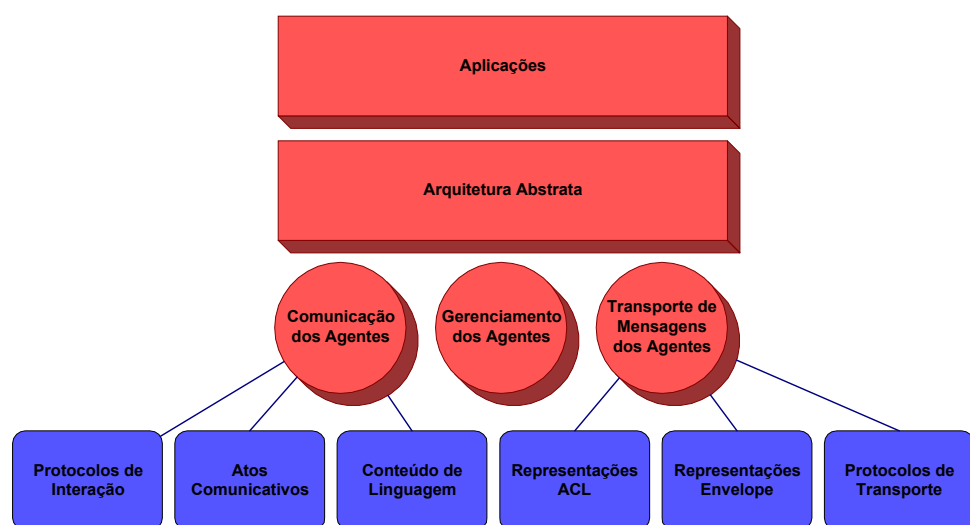


Figura B.1 – Divisão das especificações da FIPA

## B.1 Aplicações

| Número   | Título                                                         | Descrição                                                                                                                                                                                                                                                                                                                                                                       |
|----------|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00014D | FIPA Nomadic Application Support Specification                 | Esta especificação trata dos meios do agente para suporte de aplicações em ambientes nômades.                                                                                                                                                                                                                                                                                   |
| XC00079B | FIPA Agent Software Integration Specification                  | Esta especificação trata das tecnologias que habilitam a integração de serviços providos por softwares comuns em uma comunidade multi-agente.                                                                                                                                                                                                                                   |
| XC00080B | FIPA Personal Travel Assistance Specification                  | Esta especificação estende o padrão FIPA através de uma especificação de uma aplicação para indústria de turismo: um overview da indústria atual em respeito aos agentes, uma arquitetura de referência para um SMA nesta indústria, exemplos dos detalhes de gerenciamento dos agentes, dos detalhes da comunicação dos agentes, e da integração de agentes/softwarees comuns. |
| XC00081B | FIPA Audio/Visual Entertainment and Broadcasting Specification | Esta especificação descreve a avaliação das especificações da FIPA de uma aplicação em Difusão de Áudio/Vídeo e Entretenimento. As necessidades identificadas para essa aplicação são: filtragem, recuperação, padronização e compatibilidade de informação, amigável ao usuário, e automação, educação e entretenimento no lar.                                                |
| XC00082B | FIPA Network Management and Provisioning Specification         | Esta especificação apresenta as características dos agentes necessárias para gerenciamento e provisão de conexões em redes.                                                                                                                                                                                                                                                     |
| XC00083B | FIPA Personal Assistant Specification                          | Esta especificação trata das características gerais de uma aplicação em assistentes pessoais.                                                                                                                                                                                                                                                                                   |
| XC00092B | FIPA Message Buffering Service Specification                   | Esta especificação trata do armazenamento de mensagens entre agentes.                                                                                                                                                                                                                                                                                                           |

## B.2 Arquitetura Abstrata

| Número   | Título                                   | Descrição                                                              |
|----------|------------------------------------------|------------------------------------------------------------------------|
| XC00001J | FIPA Abstract Architecture Specification | Esta especificação define os elementos da arquitetura e suas relações. |

|          |                                         |                                                                                                                                                                                                                             |
|----------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PC00089D | FIPA Policies and Domains Specification | Esta especificação traz um conjunto de casos de uso e elementos da arquitetura abstrata que podem ser usados para guiar a especificação de mecanismos de plano de ação em arquiteturas de plataformas de agentes concretas. |
|----------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## B.3 Comunicação de Agentes

| Número   | Título                                   | Descrição                                                                                                                                                                                                                                           |
|----------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00061E | FIPA ACL Message Structure Specification | Este documento contém especificações para os elementos para mensagem da FIPA ACL incluindo: a lista de elementos para mensagens atuais da FIPA ACL, procedimentos para manutenção dessa lista, e critérios para adotar novos elementos nessa lista. |

### B.3.1 Protocolos de Interação

| Número   | Título                                               | Descrição                                                                                                                                                                                                                                                                                                                                                           |
|----------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00025E | FIPA Interaction Protocol Library Specification      | Este documento contém: especificações para estruturar a Biblioteca de Protocolos de Interação FIPA (IPL) incluindo um estado do Protocolo de Interação FIPA (IPs), manutenção da biblioteca e critérios de inclusão para novos IPs, uma descrição de como entender e expressar IPs usando AUML (Agent Unified Modeling Language), e a lista de registro de IP FIPA. |
| XC00026F | FIPA Request Interaction Protocol Specification      | O Protocolo de Interação de Requisição FIPA (IP) simplesmente permite um agente solicitar a outro para executar alguma ação que ele não pode.                                                                                                                                                                                                                       |
| XC00027F | FIPA Query Interaction Protocol Specification        | No Protocolo de Interação de Consulta FIPA (IP), o agente destinatário é solicitado a executar algum tipo de ação de solicitação de informação.                                                                                                                                                                                                                     |
| XC00028F | FIPA Request When Interaction Protocol Specification | O Protocolo de Interação de Solicitação Quando FIPA (IP) provê uma estrutura para o ato comunicativo de solicitar-quando. O iniciador usa a ação solicitar-quando para solicitar ao participante que execute alguma ação uma vez que uma dada pré-condição seja verdadeira.                                                                                         |

|          |                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00029F | FIPA Contract Net Interaction Protocol Specification          | Esta especificação apresenta uma versão do Protocolo de Contrato de Rede amplamente utilizado, e originalmente desenvolvido por Smith e Davis. O Protocolo de Contrato de Rede FIPA (IP) é uma pequena modificação do original, onde foram adicionados atos comunicativos de rejeição e confirmação.                                                                                                                                                                                                                                                                                                                                |
| XC00030F | FIPA Iterated Contract Net Interaction Protocol Specification | O Protocolo de Interação de Contrato de Rede Interativo FIPA (IP) é uma extensão do Protocolo de Interação de Contrato de Rede FIPA básico, e difere por permitir ordenação interativa multi-etapa.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| XC00031F | FIPA English Auction Interaction Protocol Specification       | No Protocolo de Interação de Leilão Inglês FIPA (IP), o leiloeiro procura o preço de mercado de um bem através de inicialmente propor um preço abaixo daquele suporte valor de mercado e então aumenta o preço gradativamente. Cada vez que o preço é anunciado, o leiloeiro espera para ver se algum comprador irá sinalizar que pagará o preço proposto. Assim que um comprador indica que aceitará o preço o leiloeiro inicia uma nova chamada com um preço maior, este leilão continua até que nenhum comprador aceite o novo preço. Se o último preço for maior que o preço reservado do leiloeiro o bem é vendido, senão não. |
| XC00032F | FIPA Dutch Auction Interaction Protocol Specification         | No Protocolo de Interação de Leilão Holandês (IP), o leiloeiro procura achar o preço de mercado para um bem começando o leilão com um preço muito maior que o valor de mercado esperado, então progressivamente vai reduzindo o preço até que um dos compradores aceite o preço. A taxa de redução do preço fica a cabo do leiloeiro e geralmente ele tem um preço de reserva o qual ele não ultrapassa.                                                                                                                                                                                                                            |
| XC00033F | FIPA Brokering Interaction Protocol Specification             | O Protocolo de Interação com Corretagem FIPA (IP) foi desenvolvido para suportar interações de corretagem em SMAs, onde um agente solicita um serviço para um corretor, que busca os melhores agentes para fazê-lo e depois apresenta a resposta com os agentes selecionados para o requerente original.                                                                                                                                                                                                                                                                                                                            |

|          |                                                    |                                                                                                                                                                                                                                               |
|----------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00034F | FIPA Recruiting Interaction Protocol Specification | O Protocolo de Interação com Recrutamento FIPA (IP) foi desenvolvido para suportar interações de corretagem em SMAs, e difere do anterior pelo fato da resposta dos agentes selecionados ir diretamente para o requerente original.           |
| XC00036F | FIPA Propose Interaction Protocol Specification    | Na Especificação do Protocolo de Interação por Propostas FIPA (IP), um agente iniciador propõe aos agentes receptores que ele irá fazer as ações descritas no ato comunicativo proposto quando os agentes receptores aceitarem esta proposta. |

### B.3.2 Atos Comunicativos

| Número   | Título                                       | Descrição                                                                                                                                                                                                            |
|----------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00037H | FIPA Communicative Act Library Specification | Este documento contém especificações para estruturar a Biblioteca de Atos comunicativos (CAL) incluindo: estado de um ato comunicativo concordante ao modelo FIPA, manutenção da biblioteca e critérios de inclusão. |

### B.3.3 Conteúdo de Linguagem

| Número   | Título                                      | Descrição                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00007B | FIPA Content Language Library Specification | Este documento contém especificações para estruturar a Biblioteca de Linguagem de Conteúdo FIPA (CLL) incluindo: estado da linguagem de conteúdo de um modelo concordante ao FIPA, manutenção da biblioteca e critérios de inclusão. E ainda, mantém uma lista atualizada de linguagens de conteúdo provendo a informação necessária para identificar essas linguagens e recuperar suas especificações relativas. |
| XC00008G | FIPA SL Content Language Specification      | Esta especificação define uma sintaxe concreta para a Linguagem Semântica (SL) da linguagem de conteúdo da FIPA.                                                                                                                                                                                                                                                                                                  |

|          |                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00009B | FIPA CCL Content Language Specification | Este documento traz a especificação para a Linguagem de Escolha de Restrições (CCL) a qual é definida como uma linguagem para ser usada para comunicação de agentes, e mais especificamente como uma linguagem de conteúdo para ser usada com a FIPA ACL.                                                                                                                                                                                                                                                                                                                   |
| XC00010B | FIPA KIF Content Language Specification | Este documento traz as especificações para o rascunho do Padrão Nacional Americano (ANSkif) para o Formato de Intercâmbio de Conhecimento (KIF) como uma linguagem de conteúdo para FIPA ACL e inclui: expressão de objetos como termos e de proposições como sentenças.                                                                                                                                                                                                                                                                                                    |
| XC00011B | FIPA RDF Content Language Specification | Esta especificação descreve como a Estrutura de Descrição de Recursos (RDF) pode ser usada como uma linguagem de conteúdo em uma mensagem FIPA. E ainda, mostra como esquemas RDF pode ser definido estendendo seu modelo para expressar: objetos os quais são construções que representam uma entidade identificada (seja abstrato ou concreto) no domínio do discurso, proposições as quais são declarações expressando que alguma sentença em uma linguagem é verdadeira ou falsa, e ações as quais tentam expressar uma atividade que pode ser executada por um objeto. |

## B.4 Gerenciamento de Agentes

| Número   | Título                              | Descrição                                                                                                                                                                                                       |
|----------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00023H | FIPA Agent Management Specification | Este documento contém especificações para gerenciamento de agentes incluindo serviços de gerenciamento de agentes, ontologia de gerenciamento de agentes e da plataforma de transporte de mensagens de agentes. |



## B.5 Transporte de Mensagens de Agentes

| Número   | Título                                                | Descrição                                                                                                                                                                                                                                                                                |
|----------|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00067D | FIPA Agent Message Transport Service Specification    | Esta especificação trata do transporte de mensagens entre agentes e inclui: um modelo de referência para um serviço de transporte de mensagens de agentes, e definições para a expressão da informação de transporte de mensagens para um serviço de transporte de mensagens de agentes. |
| XC00093B | FIPA Messaging Interoperability Service Specification | Esta especificação trata da conversão de mensagens entre agentes, e contém especificações para conversão de mensagens entre protocolos de transporte de mensagens diferentes e/ou codificação.                                                                                           |

### B.5.1 Representações ACL

| Número   | Título                                                                  | Descrição                                                                                                                                                 |
|----------|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00069E | FIPA ACL Message Representation in Bit-Efficient Encoding Specification | Esta especificação trata do transporte de mensagens entre agentes e contém especificações para a representação sintática da ACL no formato Bit-Eficiente. |
| XC00070G | FIPA ACL Message Representation in String Specification                 | Esta especificação trata do transporte de mensagens entre agentes e contém especificações para a representação sintática da ACL no formato String.        |
| XC00071C | FIPA ACL Message Representation in XML Specification                    | Esta especificação trata do transporte de mensagens entre agentes e contém especificações para a representação sintática da ACL no formato XML.           |

### B.5.2 Representações Envelope

| Número   | Título                                                                    | Descrição                                                                                                                                                        |
|----------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00085H | FIPA Agent Message Transport Envelope Representation in XML Specification | Esta especificação trata do transporte de mensagens entre agentes, e contém especificações para representação sintática de uma mensagem envelope em formato XML. |

|          |                                                                                              |                                                                                                                                                                           |
|----------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XC00088B | FIPA Agent Message Transport Envelope Representation in Bit-Efficient Encoding Specification | Esta especificação trata do transporte de mensagens entre agentes e contém especificações para representação sintática de uma mensagem envelope em formato Bit-Eficiente. |
|----------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### B.5.3 Protocolos de Transporte

| Número   | Título                                                       | Descrição                                                                                                                   |
|----------|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| XC00075E | FIPA Agent Message Transport Protocol for IIOP Specification | Esta especificação trata do transporte de mensagens entre agentes usando o Protocolo Inter-Orb de Internet (IIOP).          |
| XC00076C | FIPA Agent Message Transport Protocol for WAP Specification  | Esta especificação trata do transporte de mensagens entre agentes usando o Protocolo de Aplicações sem Fio (WAP).           |
| XC00084D | FIPA Agent Message Transport Protocol for HTTP Specification | Esta especificação trata do transporte de mensagens entre agentes usando o Protocolo de Transferência de Hipertexto (HTTP). |

## **ANEXO C**

# **IMPLEMENTAÇÃO COMPUTACIONAL DO SISTEMA MULTI-AGENTE PARA O RESTABELECIMENTO DE SUBESTAÇÕES ELÉTRICAS**

O Sistema Inteligente para a Supervisão e Controle de Subestações Elétricas “SISCO – Subestações” possui uma interface de dados bastante amigável, a qual facilita a entrada de dados feita pelo usuário. Na janela inicial do SISCO – Subestações, tem-se os menus habilitados “Arquivo” e “Ajuda”, com os itens típicos dos programas atuais: “Novo”, “Abrir”, “Salvar”, “Imprimir” e “Sair”.

### **C.1 Apresentando um Caso**

Após abrir um caso, novo ou já existente, os dados do sistema serão apresentados na janela abaixo, Figura C.1. O SISCO possui uma barra de menus, uma área de trabalho, onde se apresenta o diagrama unifilar da subestação, uma área de visualização dos dados de medição e uma área de visualização da saída do planejador.

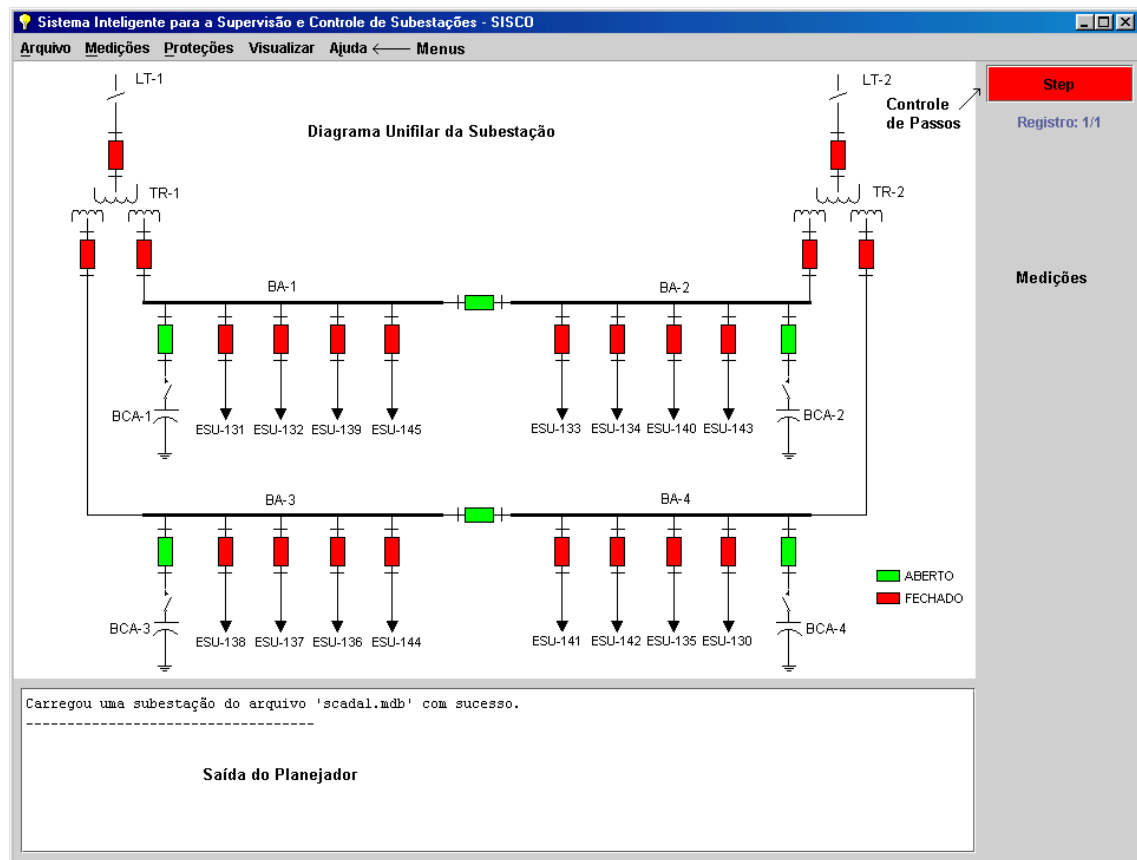


Figura C.1 – Janela principal do SISCO

O programa apresentará os seguintes menus: “Arquivo”, “Medições”, “Proteções”, “Visualizar” e “Ajuda”. Os menus “Arquivo” e “Ajuda” mantêm as mesmas opções apresentadas na janela de abertura do programa.

Ao selecionar um equipamento das listas apresentadas os seus dados de medições serão visualizados na área de medições. No menu “Medições” tem-se as opções:

- “Linha” – Apresenta uma lista das linhas disponíveis;
- “Trafo” – Apresenta uma lista dos transformadores disponíveis;
- “Barramento” – Apresenta uma lista dos barramentos disponíveis;
- “Banco de Capacitores” – Apresenta uma lista dos banco de capacitores disponíveis;
- “Alimentadores” – Apresenta uma lista dos alimentadores disponíveis.

Ao selecionar um equipamento das listas apresentadas os seus dados de proteções serão visualizados em uma janela flutuante. No menu “Proteções” tem-se as opções:

- “Linha” – Apresenta uma lista das linhas disponíveis;
- “Trafo” – Apresenta uma lista dos transformadores disponíveis;
- “Barramento” – Apresenta uma lista dos barramentos disponíveis;
- “Banco de Capacitores” – Apresenta uma lista dos bancos de capacitores disponíveis;
- “Alimentadores” – Apresenta uma lista dos alimentadores disponíveis.

No menu “Visualizar” tem-se as opções:

- “Comunicação dos Agentes” – Abre uma janela flutuante onde pode se visualizar todos os dados de comunicação entre os agentes inteligentes;
- “Saídas dos Agentes” – Abre uma janela flutuante onde se pode visualizar todas as sugestões dadas pelos agentes para solucionar a falha.

## **C.2 Visualizando e Alterando Estados de Equipamentos**

Para se visualizar os dados de medições da subestação o usuário deve acessar o menu “Medições”, depois selecionar qual tipo de equipamento deseja verificar, e por fim escolher o equipamento específico pelo código, Figura C.2.

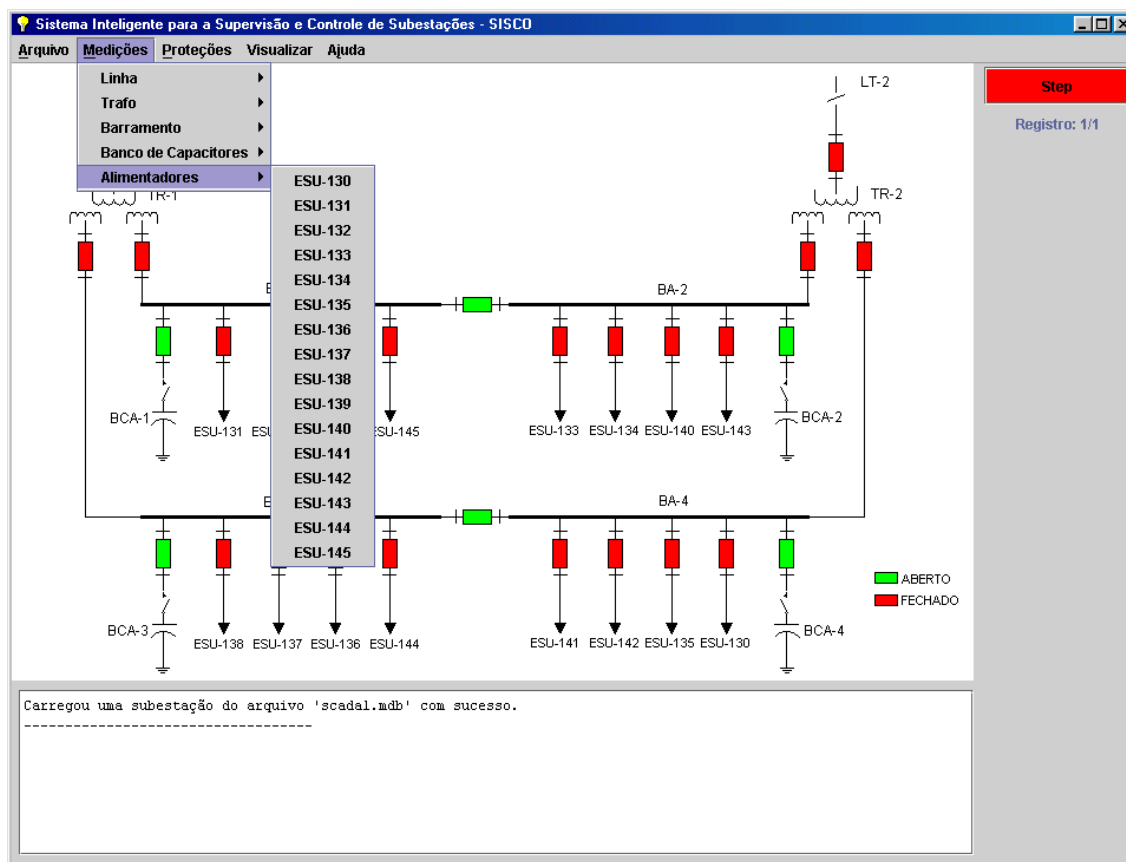


Figura C.2 – Escolhendo o equipamento para visualizar as medições

Os dados serão apresentados na área de medições, Figura C.3. O SISCO – Subestações apresenta três tipos de medição: Corrente (A), Corrente de neutro (AN) e tensão (V). Para alterar o valor de uma medição o usuário deve selecionar o campo desejado e digitar o valor no mesmo, por fim basta apertar a tecla “Enter” para validar a entrada de dados.

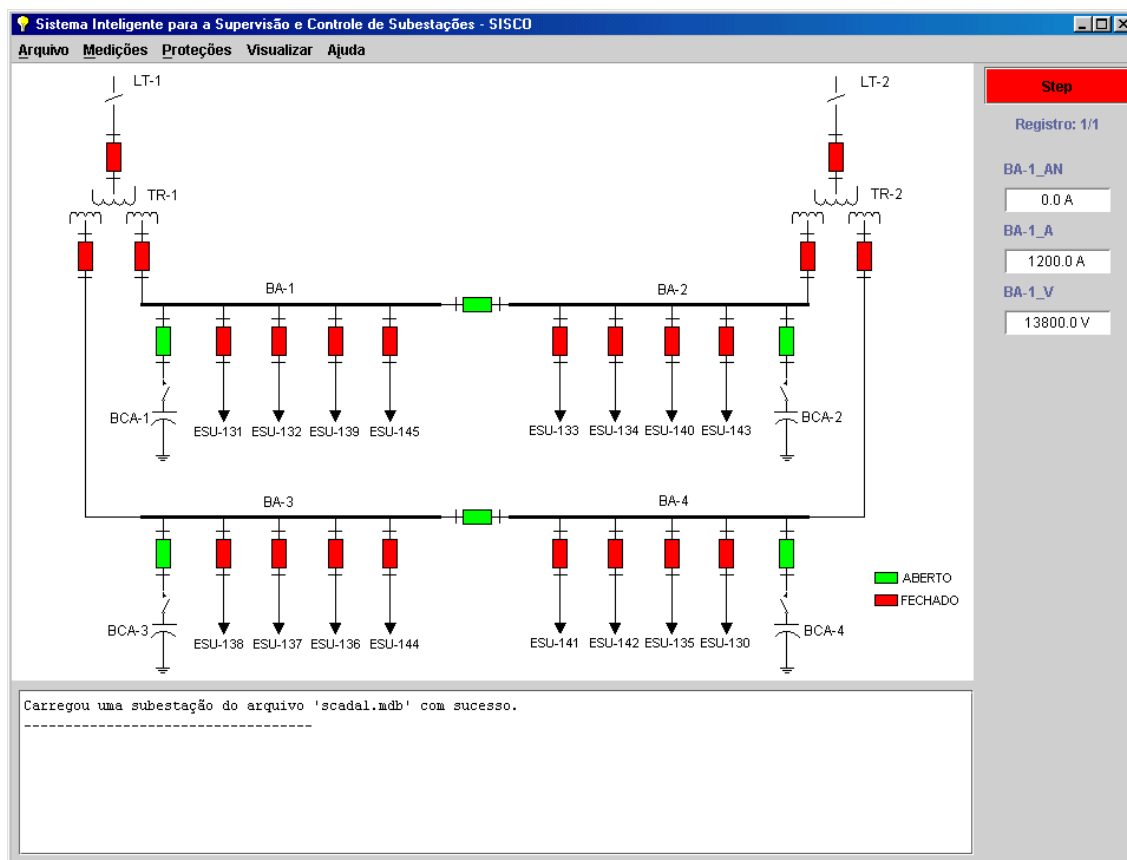


Figura C.3 – Visualizando dados de medições

Agora, para se visualizar os dados de proteções da subestação o usuário deve acessar o menu “Proteções”, depois selecionar de qual tipo de equipamento deseja verificar as proteções, e por fim escolher o equipamento específico pelo código. Os dados serão apresentados em uma janela flutuante, Figura C.4.

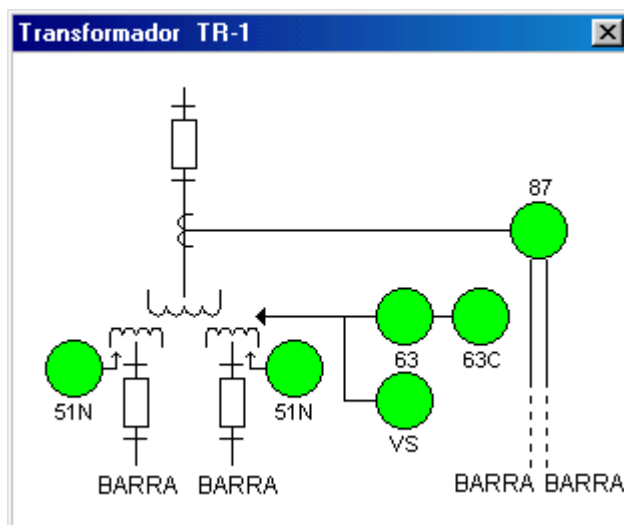


Figura C.4 – Visualizando dados de proteções

O SISCO – Subestações apresenta as proteções utilizando cores para demonstrar o seu estado: Verde para estado normal e vermelha para proteção atuada. Para alterar o estado de uma proteção o usuário deve clicar duas vezes na proteção desejada que aparecerá a janela apresentada na Figura C.5.

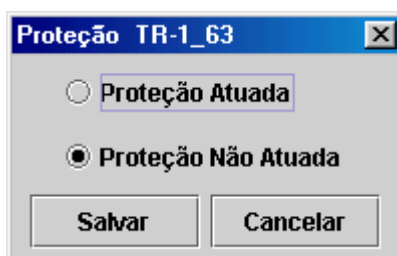


Figura C.5 – Janela de diálogo para alterar o estado das proteções

Então, o usuário seleciona o estado da proteção e clica no botão “Salvar”, ou se desejar pode cancelar a ação através do botão “Cancelar”.

Para alterar o estado de um disjuntor o usuário deve clicar duas vezes em cima da figura do disjuntor que se deseja alterar o estado que aparecerá a seguinte janela, Figura C.6.





**Figura C.6** – Janela de diálogo para alterar o estado dos disjuntores

O SISCO – Subestações apresenta dois estados possíveis para o disjuntor: aberto e fechado. Caso o usuário queira cancelar a ação basta clicar no botão “Cancelar”.

## **C.3 Simulando um Caso**

### **C.3.1 Gerando o Cenário Pré-falta**

Para simular um caso deve-se primeiramente provocar uma falha na subestação, para tal o usuário deverá ajustar os estados dos disjuntores, medições e proteções da subestação criando assim um cenário pré-falta.

Por exemplo, seja uma falha no transformador TR-1 por atuação da proteção TR-1\_63, Figura C.7.

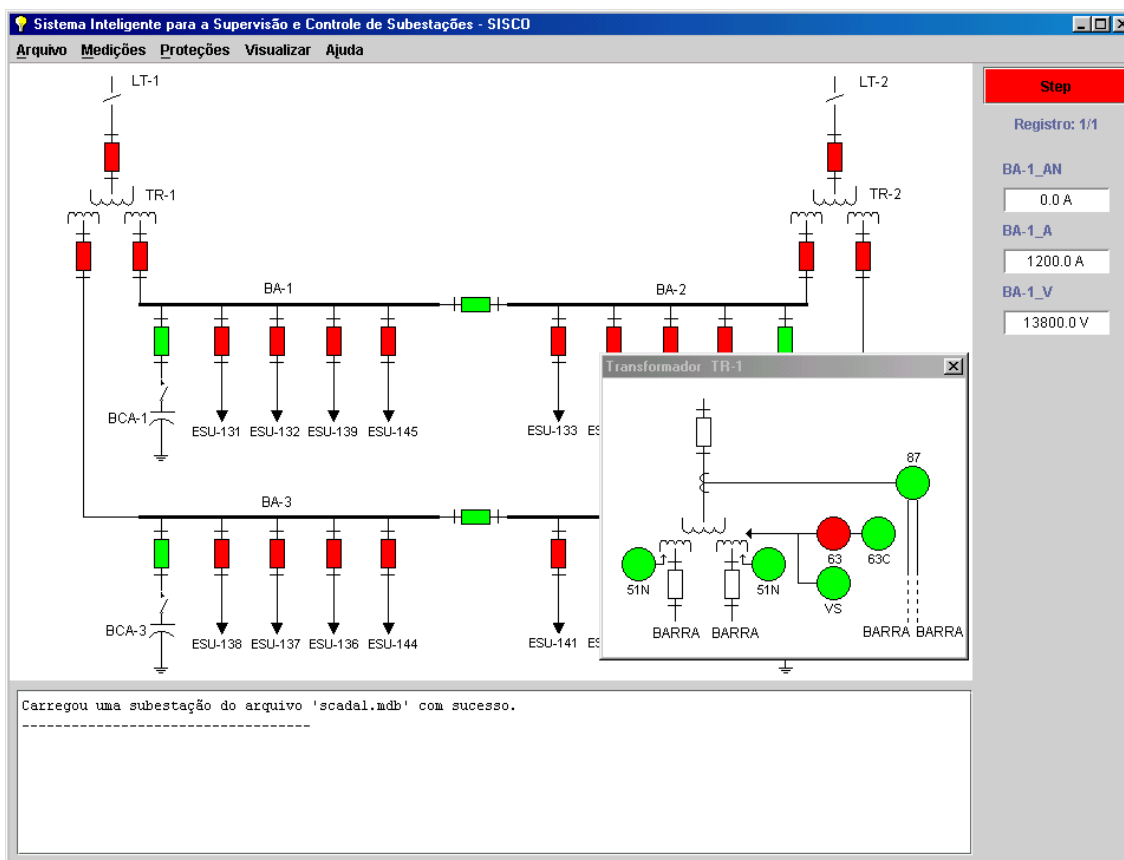


Figura C.7 – Gerando o cenário pré-falta

### C.3.2 Simulando

O usuário deve clicar no botão “Step” para iniciar a simulação. A cada passo o sistema inteligente avalia as condições de todos os objetos do modelo e toma as ações necessárias, Figura C.8.

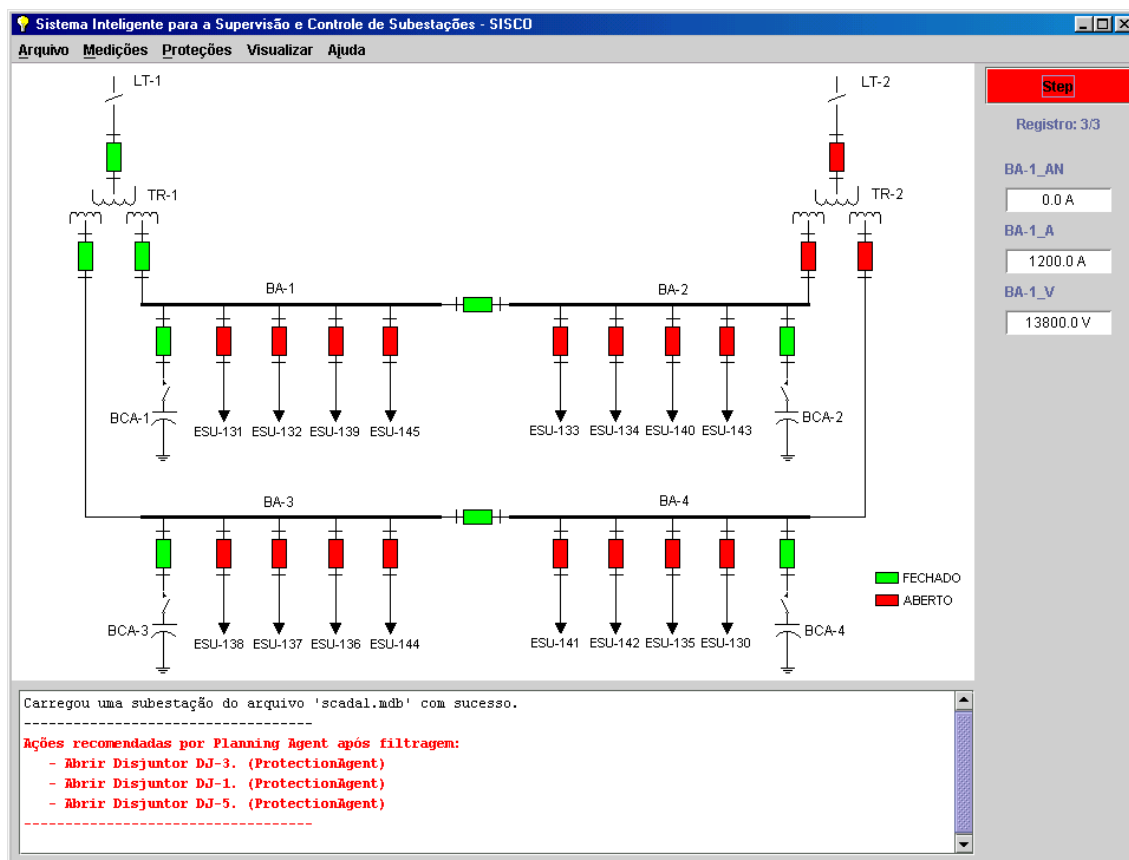


Figura C.8 – Simulando uma falha

O usuário deverá clicar no botão “Step” para que todas as conseqüências da falha sejam apresentadas, Figura C.9.

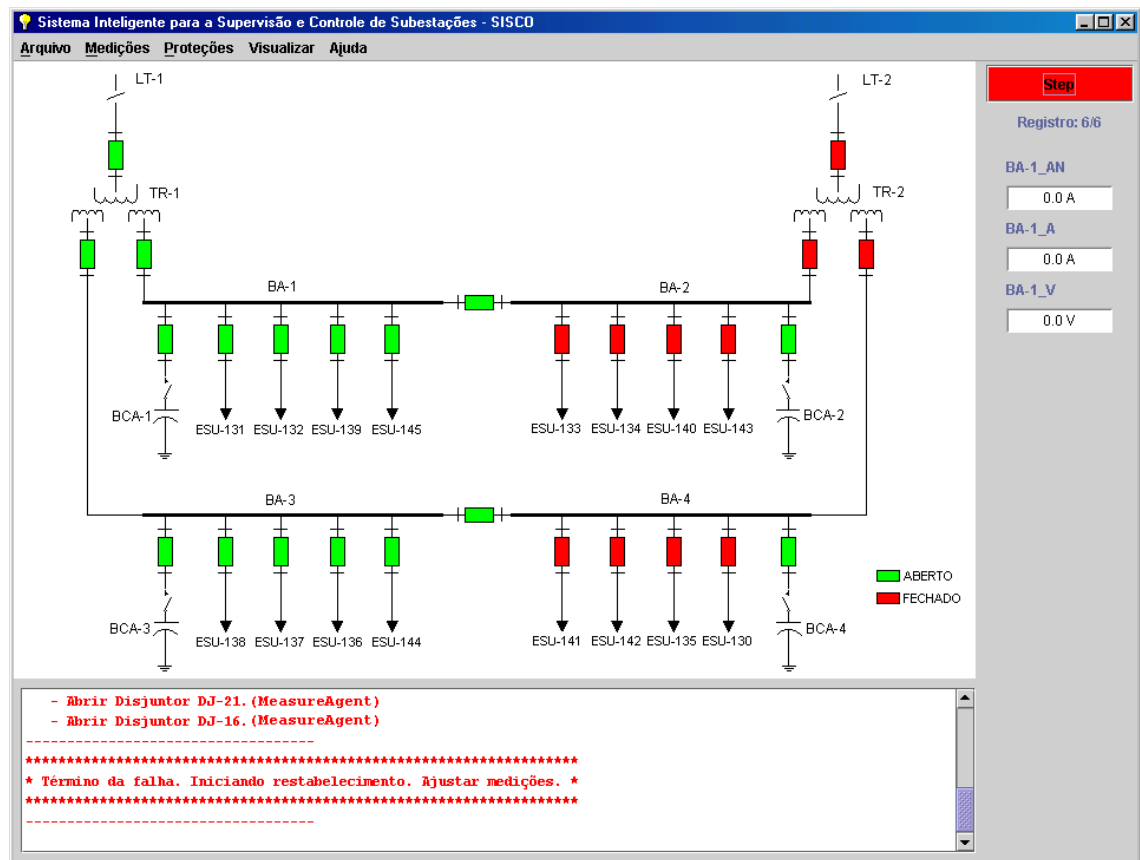


Figura C.9 – Fim da falha

Para iniciar o restabelecimento o usuário deve ajustar as medições necessárias, digitando-as nos devidos campos, e desarmar a proteção atuada, isto indicará que o problema do transformador foi resolvido. No exemplo, é necessário indicar ao sistema inteligente que há tensão na linha LT-1.

Então o planejador desenvolverá um plano inicial de restabelecimento que o usuário começará a desenvolver, Figura C.10.

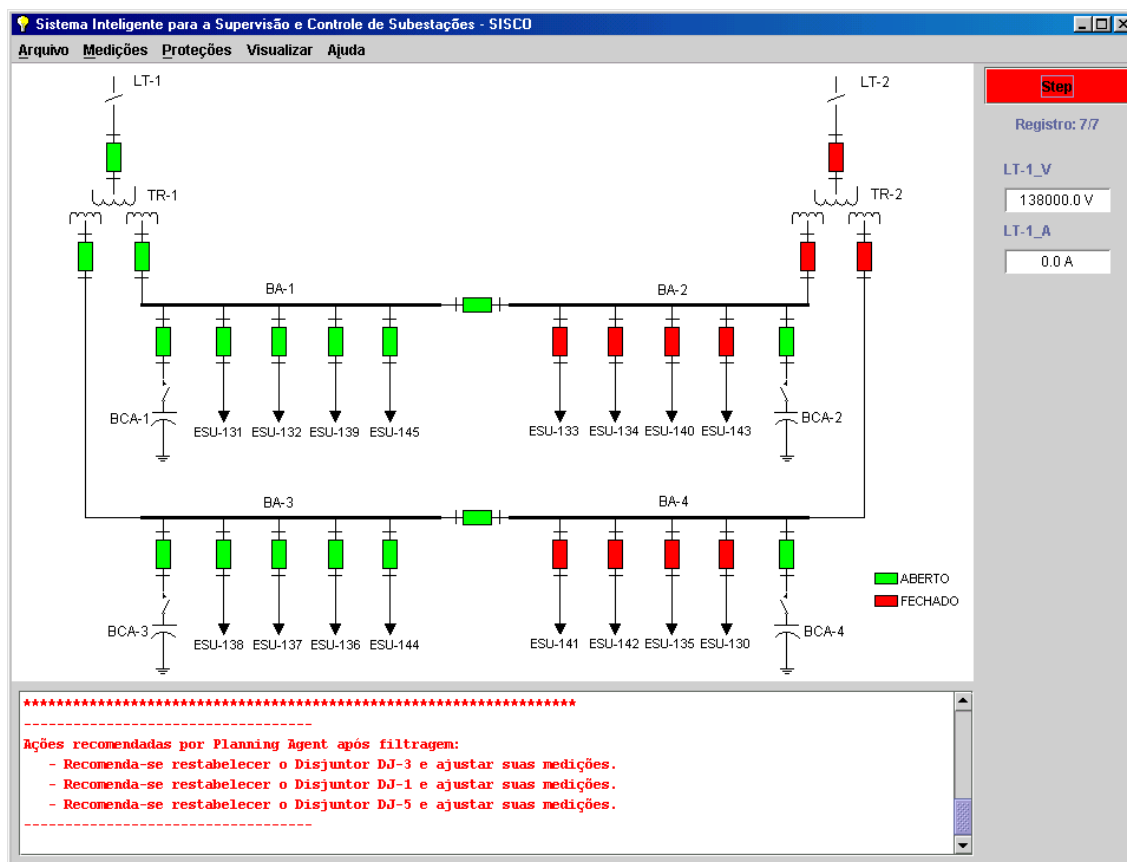


Figura C.10 – Plano inicial de restabelecimento

De posse do plano inicial de restabelecimento o usuário deverá iniciar as ações de restabelecimento na ordem apresentada pelo planejador, caso haja algum problema em executar alguma ação, o planejador está sempre avaliando se a ação tomada está correta, se houver uma ação que não está correta ele indicará a correção ou uma ação alternativa.

### C.3.3 Restabelecendo a Subestação

Para restabelecer a subestação o usuário deve estar atento às medições da subestação, pois elas comprovam que a ação foi executada corretamente, por exemplo, ao fechar o disjuntor DJ-1 e DJ-5 a barra BA-1 estará automaticamente energizada e sua medição de tensão deverá ser ajustada, Figura C.11.

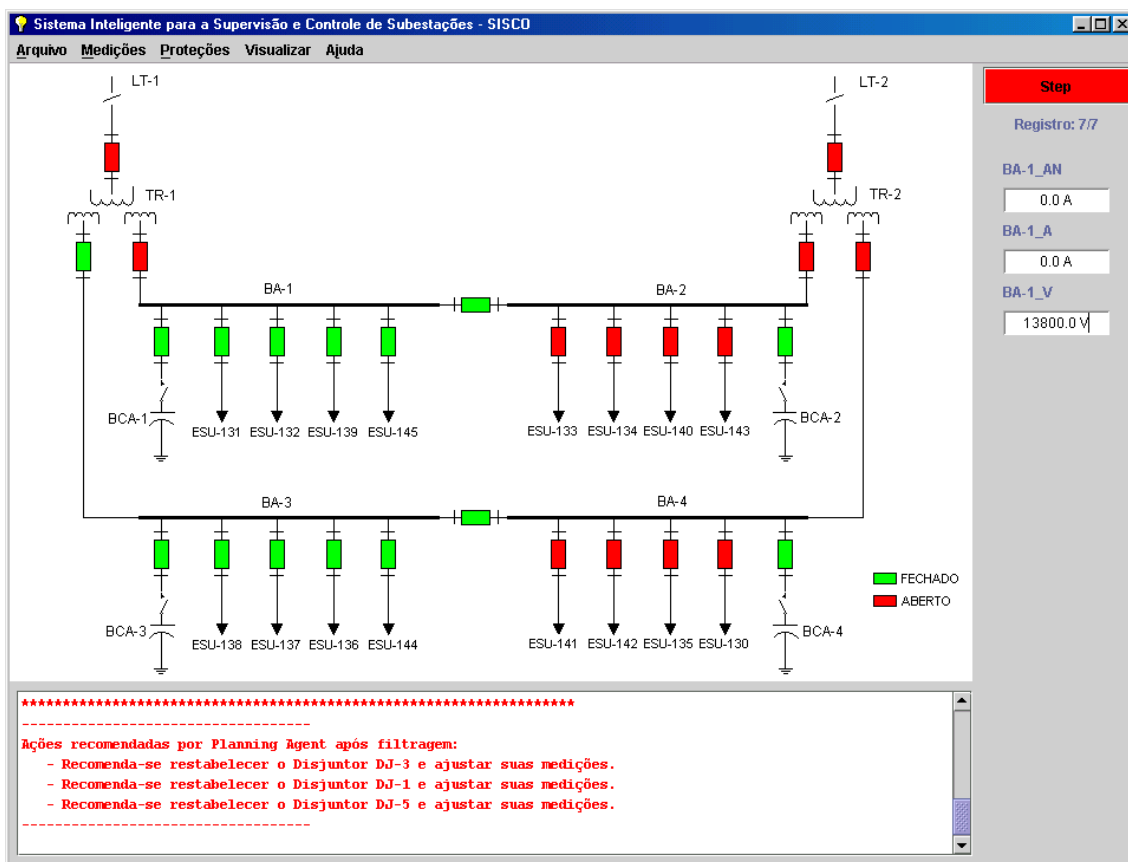


Figura C.11 – Ajustando as medições durante o restabelecimento

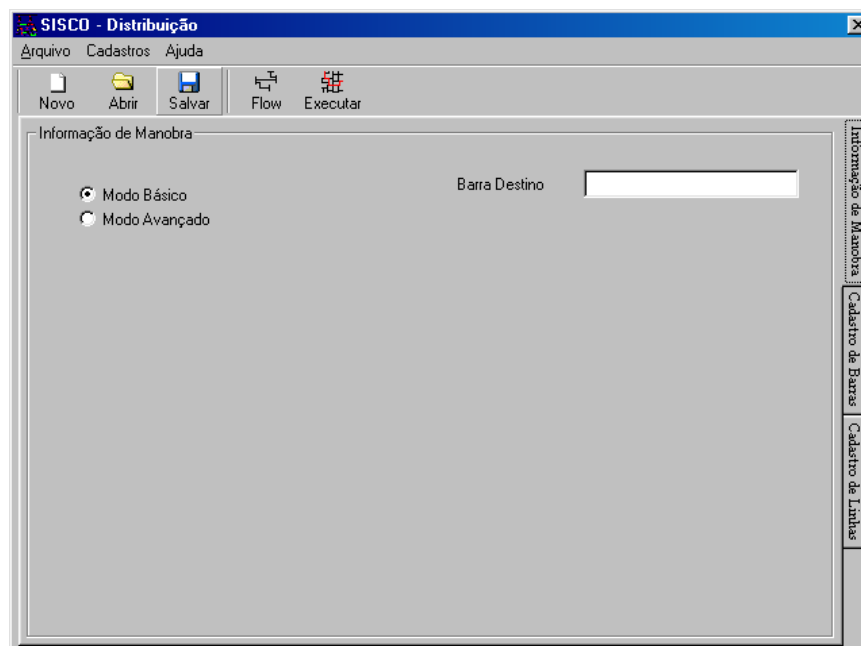
Caso o usuário não ajuste as medições o programa não avança para as próximas ações, ou sugere correções, ou até mesmo desfaz as ações do usuário por conterem inconsistências.

## ANEXO D

# IMPLEMENTAÇÃO COMPUTACIONAL DO SISTEMA MULTI-AGENTE PARA O RESTABELECIMENTO DE SISTEMAS DE DISTRIBUIÇÃO

O Sistema Inteligente para a Supervisão e Controle de Sistemas de Distribuição “SISCO – Distribuição” apresenta a seguinte janela principal para o usuário (Figura D.1) e possui um menu com as seguintes opções:

- Arquivo:
  - Novo, Abrir, Salvar e Sair – Funções típicas encontradas nos programas atuais;
  - Importar – Barras – Importa dados de barras de um sistema de distribuição;
  - Importar – Linhas – Importa dados de linhas de transmissão de um sistema de distribuição;
- Cadastros:
  - Manobras – Apresenta a janela com os dados necessários para a realização da busca da manobra;
  - Linhas – Apresenta a janela com os dados cadastrais das linhas de transmissão do sistema de distribuição;
  - Barras – Apresenta a janela com os dados cadastrais das barras do sistema de distribuição.
- Ajuda:
  - Tópicos e Sobre – Funções típicas encontradas nos programas atuais.



**Figura D.1** – Janela principal do programa “SISCO – Distribuição”

Na janela principal o usuário ainda possui uma barra de ferramentas com botões para acessar algumas funções do programa:

- Novo, Abrir e Salvar – Realizam as mesmas funções encontradas no menu;
- Flow – Executa um algoritmo de fluxo de carga para determinar o carregamento do sistema de distribuição antes de qualquer operação, ou falha ter ocorrido (caso base);
- Executar – Executa a análise da manobra necessária para a reconfiguração do sistema de distribuição.

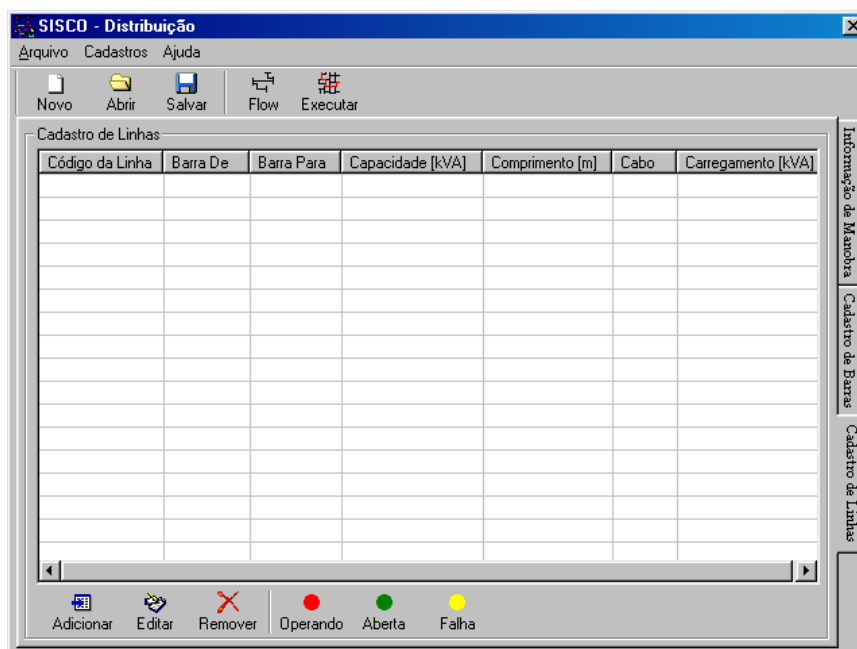
## D.1 Cadastrando um Sistema

Para cadastrar um sistema o usuário deverá cadastrar as linhas de transmissão (LTs) e as barras separadamente acessando as suas tabelas “Cadastro de Barras” e “Cadastro de Linhas”.

Ao acessar o cadastro de LTs o usuário encontrará uma nova barra de ferramentas abaixo da área de trabalho (Figura D.2) com as seguintes opções:



- Adicionar – Adiciona uma LT no sistema de distribuição existente;
- Editar – Edita a LT selecionada do sistema de distribuição existente;
- Remover – Remove a LT selecionada do sistema de distribuição existente;
- Operando – Altera o estado da LT selecionada para “Operando”;
- Aberta – Altera o estado da LT selecionada para “Aberta”;
- Falha – Altera o estado da LT selecionada para “Falha”.



**Figura D.2 – Área de cadastro de linhas**

O usuário deverá inserir todos os dados das linhas pertencentes ao sistema de distribuição, clicando no botão adicionar aparecerá a janela da Figura D.3. Os dados necessários para cadastrar uma LT são:

- Código da Linha – Código utilizado pela Concessionária para denominar a LT;
- Barra De – Barra de origem da LT;
- Barra Para – Barra de destino da LT;
- Capacidade [kVA] – Capacidade da LT em kVA;
- Comprimento [m] – Comprimento da LT em m;

- Cabo – Tipo do cabo empregado na LT.

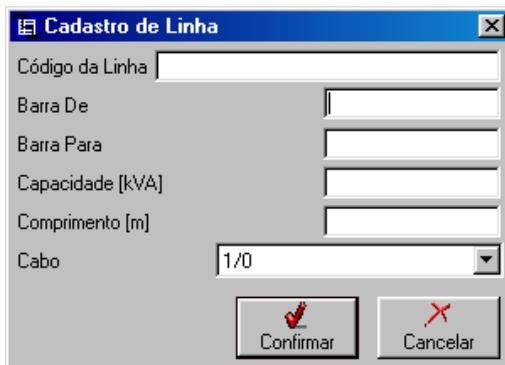


Figura D.3 – Janela para cadastro de linhas

Ao acessar o cadastro de barras o usuário também encontrará uma barra de ferramentas abaixo da área de trabalho (Figura D.4) com as seguintes opções: adicionar, editar e remover.

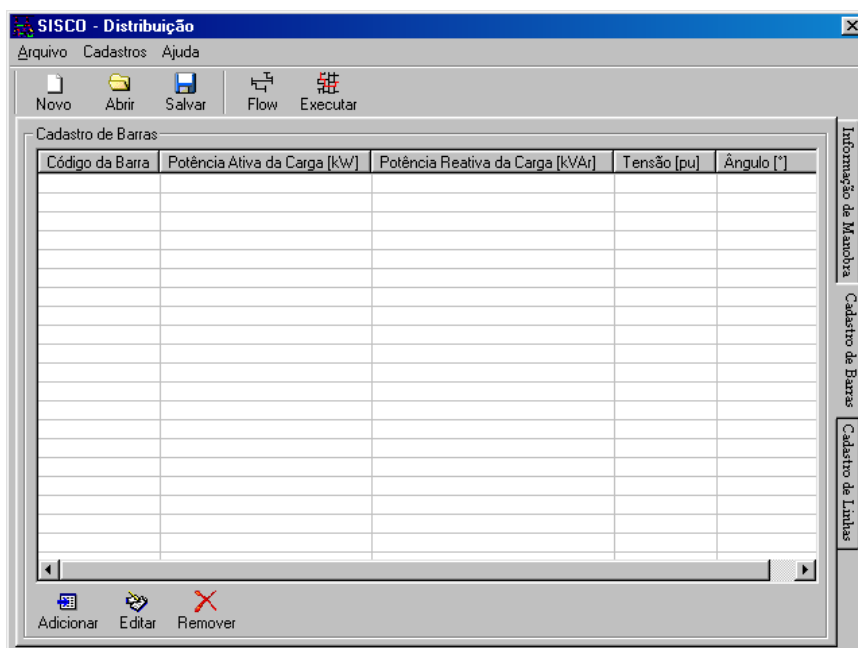


Figura D.4 – Área de cadastro de barras

O usuário deverá inserir todos os dados das barras pertencentes ao sistema de distribuição, clicando no botão adicionar aparecerá a janela da Figura D.5. Os dados necessários para cadastrar uma barra são:

- Código da Barra – Código utilizado pela Concessionária para denominar a barra;

- Potência Ativa da Carga [kW] – Potência ativa da carga da barra em kW;
- Potência Reativa da Carga [kVAr] – Potência reativa da carga da barra em kVAr.

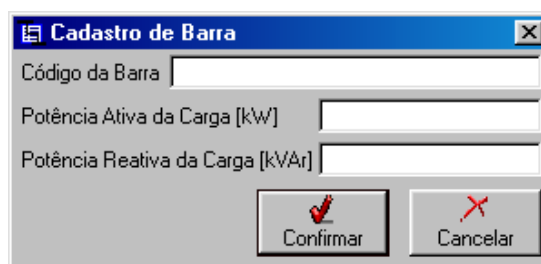


Figura D.5 – Janela para cadastro de barras

## D.2 Simulando um Sistema

O primeiro passo para simular um sistema é aplicar o algoritmo de fluxo de carga no sistema cadastrado. Para isso basta o usuário clicar no botão “Flow” na barra de ferramentas principais.

O programa então fornecerá como resposta os fluxos em todas as linhas (Figura D.6), bem como as tensões em pu e seus respectivos ângulos para todas as barras (Figura D.7). Esses valores devem ser calculados antes de qualquer operação, ou falha no sistema (caso base).

Os fluxos serão utilizados na simulação para determinar sobrecargas no sistema, antes e depois da reconfiguração. Já as tensões também servirão de parâmetro de análise de subtensões e sobretensões nas barras, antes e depois da reconfiguração.

| Código da Linha | Barra De | Barra Para | Capacidade [kVA] | Comprimento [m] | Cabo | Carregamento [k] |
|-----------------|----------|------------|------------------|-----------------|------|------------------|
| L01             | 1        | 5          | 100              | 100             | 1/0  | 16,1             |
| L02             | 5        | 7          | 100              | 100             | 1/0  | 10,1             |
| L03             | 5        | 8          | 50               | 100             | 1/0  | 4,0              |
| L04             | 8        | 13         | 2                | 100             | 1/0  | -                |
| L05             | 7        | 10         | 100              | 100             | 1/0  | 8,1              |
| L06             | 10       | 13         | 50               | 100             | 1/0  | 4,0              |
| L07             | 10       | 11         | 50               | 100             | 336  | 2,0              |
| L08             | 11       | 12         | 50               | 100             | 336  | 1,0              |
| L09             | 13       | 15         | 50               | 100             | 1/0  | 3,0              |
| L10             | 15       | 17         | 50               | 100             | 336  | 1,6              |
| L11             | 17       | 19         | 50               | 100             | 336  | 0,8              |
| L12             | 15       | 18         | 50               | 100             | 336  | 0,8              |
| L13             | 1        | 6          | 100              | 100             | 1/0  | 14,6             |
| L14             | 6        | 9          | 100              | 100             | 1/0  | 13,5             |
| L15             | 9        | 23         | 50               | 100             | 1/0  | 2,0              |
| L16             | 9        | 14         | 100              | 100             | 336  | 10,4             |
| L17             | 14       | 15         | 100              | 100             | 336  | -                |

Figura D.6 – Fluxos calculados pelo “Flow”

| Código da Barra | Potência Ativa ... | Potência Reativa ... | Tensão [pu] | Ângulo [°] |
|-----------------|--------------------|----------------------|-------------|------------|
| 0               | 0                  | 0                    | 1.0000000   | 0.0000000  |
| 1               | 100                | 20                   | 0.9999744   | -0.0028726 |
| 2               | 0                  | 0                    | 0.9999888   | -0.0012538 |
| 3               | 0                  | 0                    | 0.9999858   | -0.0016003 |
| 4               | 0                  | 0                    | 0.9999854   | -0.0016452 |
| 5               | 10                 | 1                    | 0.9999489   | -0.0038892 |
| 6               | 5                  | 2                    | 0.9999513   | -0.0037947 |
| 7               | 10                 | 1                    | 0.9999329   | -0.0045270 |
| 8               | 20                 | 1                    | 0.9999426   | -0.0041413 |
| 9               | 5                  | 1                    | 0.9999299   | -0.0046487 |
| 10              | 10                 | 3                    | 0.9999201   | -0.0050382 |
| 11              | 5                  | 1                    | 0.9999191   | -0.0051516 |
| 12              | 5                  | 0                    | 0.9999186   | -0.0052078 |
| 13              | 5                  | 0                    | 0.9999138   | -0.0052903 |
| 14              | 5                  | 0                    | 0.9999247   | -0.0052349 |
| 15              | 3                  | 0                    | 0.9999091   | -0.0054794 |
| 16              | 5                  | 2                    | 0.9999200   | -0.0057649 |
| 17              | 4                  | 0                    | 0.9999083   | -0.0055632 |
| 18              | 4                  | 0                    | 0.9999087   | -0.0055243 |
| 19              | 4                  | 0                    | 0.9999079   | -0.0056141 |
| 20              | 30                 | 6                    | 0.9999169   | -0.0061086 |
| 21              | 0                  | 4                    | 0.9999100   | 0.0050007  |

Figura D.7 – Tensões e ângulos calculados pelo “Flow”

O programa é dotado de duas metodologias de reconfiguração, a primeira é denominada de “Modo Básico” e a segunda de “Modo Avançado”, a diferença entre as duas é o algoritmo de busca implementado em cada uma. E ainda, conta com um sistema exemplo chamado “Exemplo.sbl” o qual é apresentado na Figura D.8.

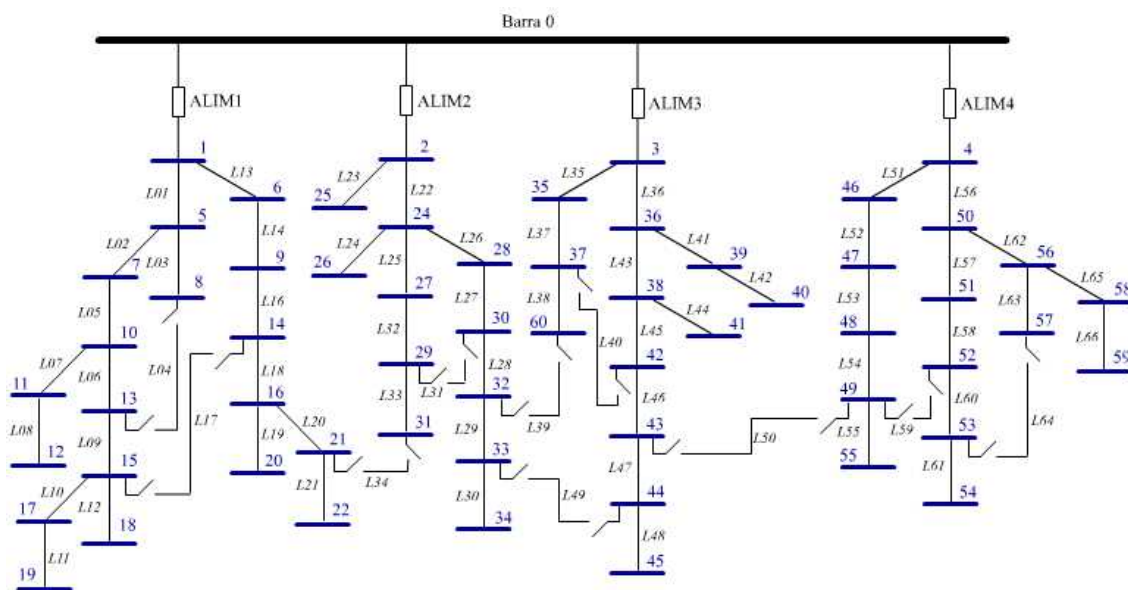


Figura D.8 – Sistema exemplo

Todas as simulações irão apresentar suas soluções em uma janela conforme Figura D.9.

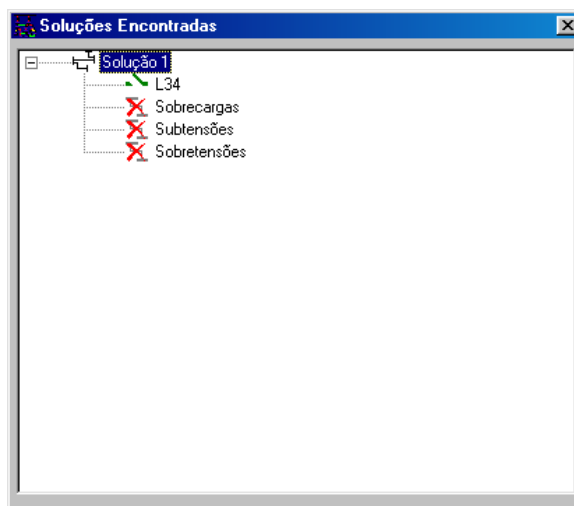


Figura D.9 – Resposta da simulação

### D.2.1 Simulando um Sistema no Modo Básico

Na primeira metodologia, “Modo Básico”, o usuário precisará informar ao programa apenas a barra que ele deseja reenergizar, e o programa irá apresentar apenas a manobra necessária.

O programa além de apresentar quais as chaves normalmente abertas deverão ser fechadas, ainda analisa se essa manobra não causará nenhuma sobrecarga, subtensão ou sobretensão no sistema.

Após carregar o arquivo exemplo e executar o fluxo de carga, o usuário deverá então entrar com os dados da falha, ou seja, selecionar apropriadamente as linhas que tiveram seu estado alterado para “Falha”, por exemplo, colocando o alimentador ALIM1 em falha, Figura D.10. Para tal, basta selecionar o ALIM1 na lista de linhas e clicar no botão “Falha”.

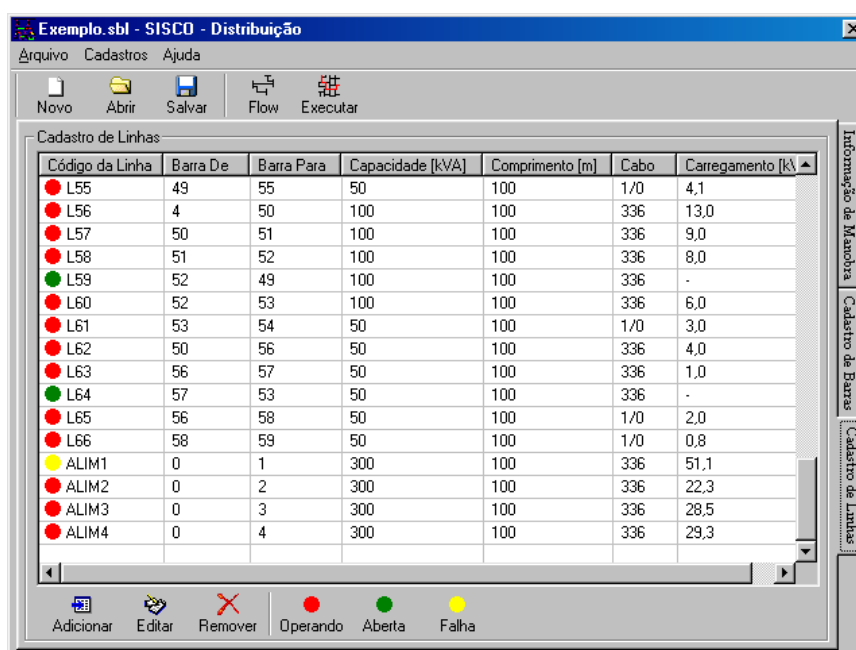


Figura D.10 – Alterando o estado da linha para “Falha”

Nesse exemplo essa metodologia será aplicada solicitando ao programa que reenergize a barra 1, Figura D.11.

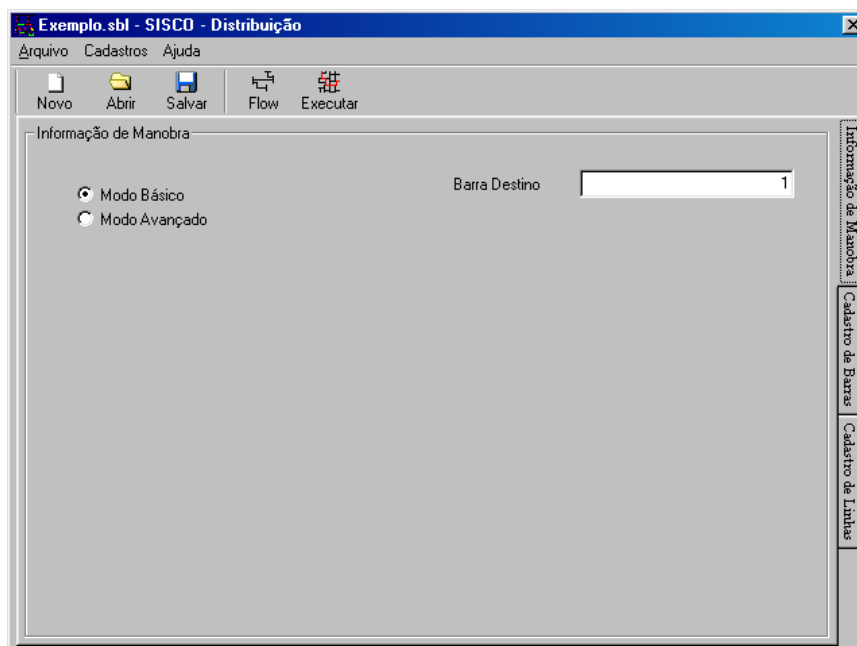


Figura D.11 – Simulando em “Modo Básico”

A resposta apresentada pelo programa está mostrada na Figura D.12. Para este exemplo foi determinada a necessidade de chavear a linha L34 e não houve ocorrências de sobrecargas, subtensões e sobretensões.

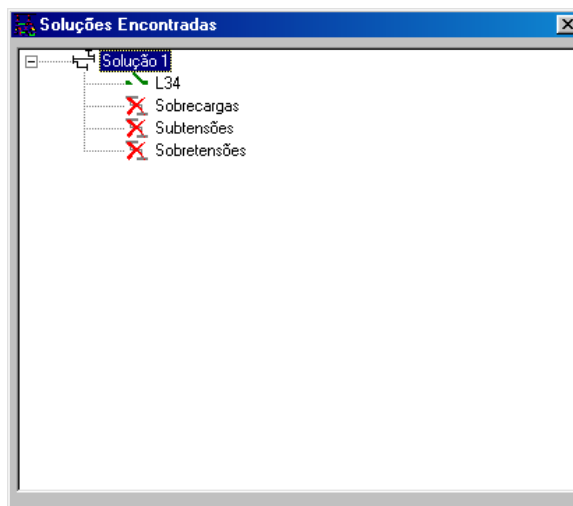


Figura D.12 – Resposta da simulação em “Modo Básico”

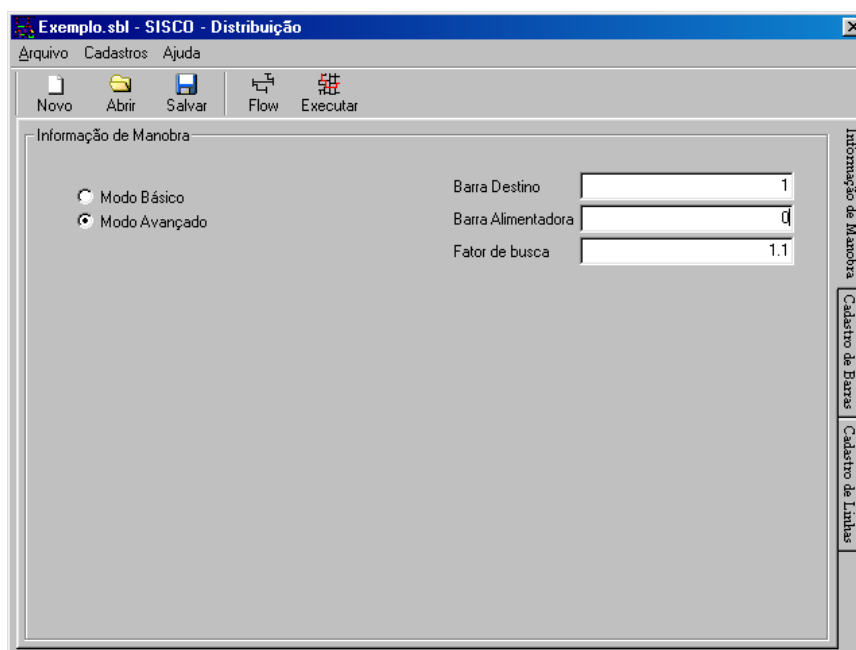
## D.2.2 Simulando um Sistema no Modo Avançado

Já no “Modo Avançado” o usuário deverá informar a barra que deseja energizar, bem como a barra de onde ele pretende partir a busca, e ainda um

terceiro parâmetro que é um fator que influencia na quantidade de caminhos analisados pelo programa, este fator age como um multiplicador aumentando o espaço de busca do programa.

Neste tipo de simulação o programa apresenta não só as chaves que precisam ser fechadas bem como todo o caminho percorrido para completar a manobra, além de analisar se essa manobra não causará nenhuma sobrecarga, subtensão ou sobretensão no sistema.

Nesse exemplo pode-se fazer, a reconfiguração da barra 1 partindo da barra 0 (subestação) com um fator de busca de 1.1 (aumento de 10% no espaço de busca), conforme Figura D.13.



**Figura D.13** – Simulando em “Modo Avançado”

A resposta apresentada pelo programa está mostrada na Figura D.14. Para essa simulação foram encontradas duas soluções, sendo que a primeira é a mesma solução encontrada na simulação em modo básico. Já a segunda solução consta além do chaveamento da linha L34, o chaveamento da linha L31, sem que haja eventos de sobrecarga, subtensão e sobretensões.



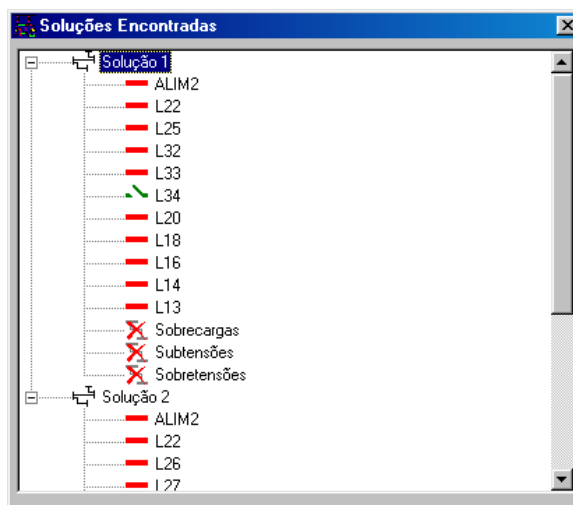


Figura D.14 – Resposta da simulação em “Modo Avançado”

### D.2.3 Simulando Falhas Simultâneas

Caso haja a necessidade de se simular falhas simultâneas, podem-se ter dois procedimentos dependendo do tipo da falha. Se as falhas não se sobrepuserem, ou seja, a segunda falha desenergiza uma área externa à que foi desenergizada pela primeira, pode-se utilizar o “Modo Básico” do programa.

Por exemplo, considere uma falha na linha L09 que tenha ocorrido quase ao mesmo tempo em que uma falha na linha L20. Para se obter o resultado desejado basta que o usuário ajuste o sistema com essas duas linhas em falha (Figura D.15) e simule a reconfiguração colocando como barra de destino a barra 0. Como a barra 0 já está energizada o programa entende que todas as soluções encontradas são possíveis apresentando a seguinte resposta, Figura D.16.

Nessa resposta o programa apresenta as duas manobras necessárias para reenergizar as regiões afetadas pelas falhas: deve-se chavear a linha L17 para reenergizar as barras 15, 17, 18 e 19; e chavear a linha L34 para reenergizar as barras 21 e 22. Nenhuma manobra causa sobrecargas, subtensões ou sobretensões no sistema.

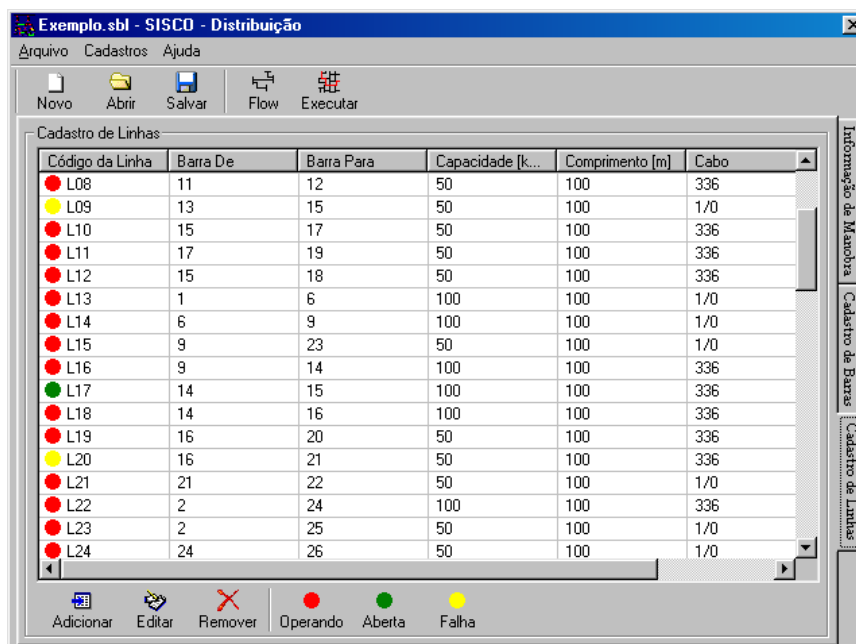


Figura D.15 – Ajustando os dados para duas falhas simultâneas

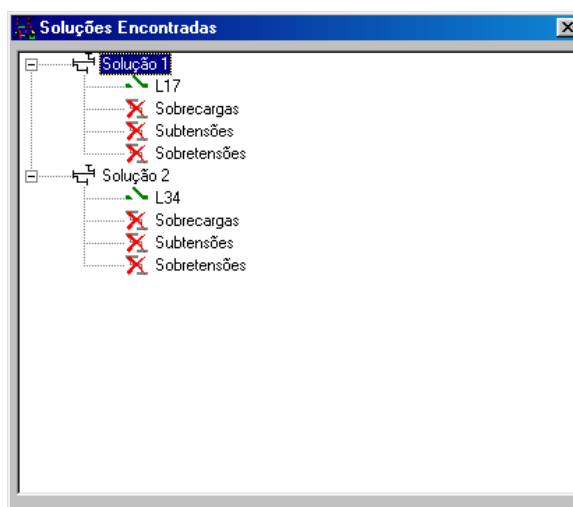


Figura D.16 – Resposta obtida para a reconfiguração do sistema

Agora, caso haja uma falha do tipo, alimentador ALIM1 e ALIM2 ao mesmo tempo (Figura D.17), nesse caso a simulação deverá ser feita através do algoritmo do “Modo Avançado”. Esse tipo de falha indica uma sobreposição de áreas afetadas, pois quando o ALIM2 entra em falha ele isola a área desenergizada pela falha do ALIM1.

Nesse caso o usuário deverá colocar como barra de partida a barra 0 (barra da subestação), e a barra de destino deverá ser a barra 1, que é a barra mais

interna das áreas em falha. Já o fator de busca será mantido em 1.1. A solução encontrada (Figura D.18) parte do alimentador ALIM3, reenergizando primeiramente a área afetada do alimentador ALIM2, através do chaveamento da linha L39, e depois a área afetada do alimentador ALIM1, através do chaveamento das linhas L31 e L34. Não há ocorrências de sobrecarga, subtensões e sobretensões para essa manobra.

The screenshot shows a software window titled 'Exemplo.sbl - SISCO - Distribuição'. It features a menu bar with 'Arquivo', 'Cadastros', and 'Ajuda'. Below the menu is a toolbar with icons for 'Novo', 'Abrir', 'Salvar', 'Flow', and 'Executar'. The main area is a table titled 'Cadastro de Linhas' with the following data:

| Código da Linha | Barra De | Barra Para | Capacidade [k... | Comprimento [m] | Cabo |
|-----------------|----------|------------|------------------|-----------------|------|
| L55             | 49       | 55         | 50               | 100             | 1/0  |
| L56             | 4        | 50         | 100              | 100             | 336  |
| L57             | 50       | 51         | 100              | 100             | 336  |
| L58             | 51       | 52         | 100              | 100             | 336  |
| L59             | 52       | 49         | 100              | 100             | 336  |
| L60             | 52       | 53         | 100              | 100             | 336  |
| L61             | 53       | 54         | 50               | 100             | 1/0  |
| L62             | 50       | 56         | 50               | 100             | 336  |
| L63             | 56       | 57         | 50               | 100             | 336  |
| L64             | 57       | 53         | 50               | 100             | 336  |
| L65             | 56       | 58         | 50               | 100             | 1/0  |
| L66             | 58       | 59         | 50               | 100             | 1/0  |
| ALIM1           | 0        | 1          | 300              | 100             | 336  |
| ALIM2           | 0        | 2          | 300              | 100             | 336  |
| ALIM3           | 0        | 3          | 300              | 100             | 336  |
| ALIM4           | 0        | 4          | 300              | 100             | 336  |

At the bottom of the table, there are icons for 'Adicionar', 'Editar', 'Remover', and status indicators for 'Operando' (red dot), 'Aberta' (green dot), and 'Falha' (yellow dot). On the right side, there are vertical tabs for 'Informação de Manobra', 'Cadastro de Barras', and 'Cadastro de Linhas'.

Figura D.17 – Ajustando os dados para duas falhas simultâneas sobrepostas

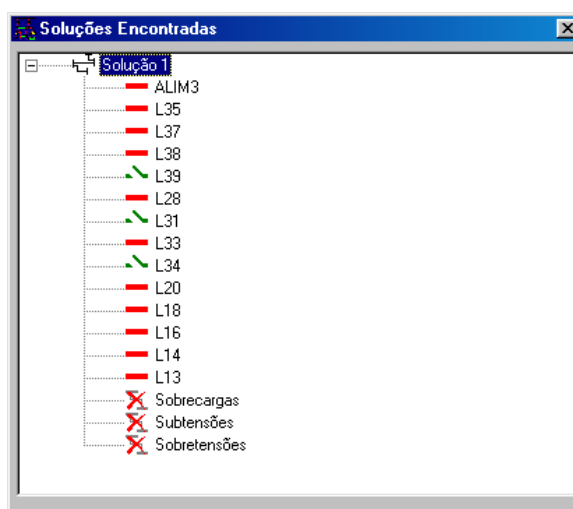


Figura D.18 – Resposta obtida para a reconfiguração do sistema

# ANEXO E

## SISTEMA INTELIGENTE DE LOCOMOÇÃO DE VIATURAS PARA ATENDIMENTO NA REDE DE DISTRIBUIÇÃO

Este projeto foi desenvolvido para a Companhia Energética de Brasília (CEB), no qual foram implementados dois programas, sendo um denominado por “Programa de Desenvolvimento” e o outro denominado por “Programa Operacional”.

A principal diferença entre os dois *softwares* é que o “Programa de Desenvolvimento” constitui a plataforma onde se desenvolveram todos os métodos de busca, análise e classificação dos dados. Nesse programa é possível acessar e configurar todas as bases de dados, bem como visualizar os trajetos encontrados.

Já o “Programa Operacional” sintetiza todas essas funções em uma plataforma desenvolvida especialmente para o uso no dia-a-dia da operação preparada para ser integrada no Sistema de Apoio a Operação (SAO) da CEB. Todas as funções e recursos do “Programa de Desenvolvimento” foram incorporados de maneira explícita ou implícita ao mesmo.

Estes aplicativos foram desenvolvidos em Visual Basic 6.0 utilizando diversas ferramentas adicionais para o manuseio dos bancos de dados, das lógicas de busca e das interfaces de apresentação do problema.

Os principais objetivos almejados com o uso desse pacote de programas são:

- Determinar a melhor viatura para atender a um ponto de defeito na rede de distribuição de energia elétrica baseado na análise do melhor caminho que esta deve efetuar. O melhor caminho entre a localização da viatura e o ponto de defeito é determinado levando-se em consideração aspectos como: vias

disponíveis na cidade, tráfego, horários de pico e engarrafamentos, entre outros;

- Criar a possibilidade de alternativas possíveis para locomoção dos veículos dentro da área de concessão da CEB, diminuindo seu tempo de deslocamento até o local do problema;
- Contribuir para a melhoria dos índices de continuidade dos serviços;
- Identificar outras possíveis viaturas que possam dar suporte à solução do problema da rede elétrica.

O desenvolvimento e o uso desses aplicativos visam alcançar vários benefícios, dos quais podem-se citar:

- A melhor utilização da capacidade da frota de veículos da CEB visando corrigir defeitos na rede;
- O estabelecimento de uma metodologia que permita uma análise da melhor locomoção dos veículos, reduzindo custos operacionais;
- O aumento da qualidade dos serviços prestados aos clientes, com um atendimento mais eficiente aos problemas existentes na rede de distribuição.
- A diminuição do tempo no processo de tomada de decisão.

## **E.1 Fluxograma Global de Operação**

Na Figura E.1 será apresentado o fluxograma global de operação do sistema multi-agente integrado ao SAO, neste será explicitado o caminho da informação, bem como o processamento empregado em cada parcela do conjunto de dados para a obtenção do resultado.

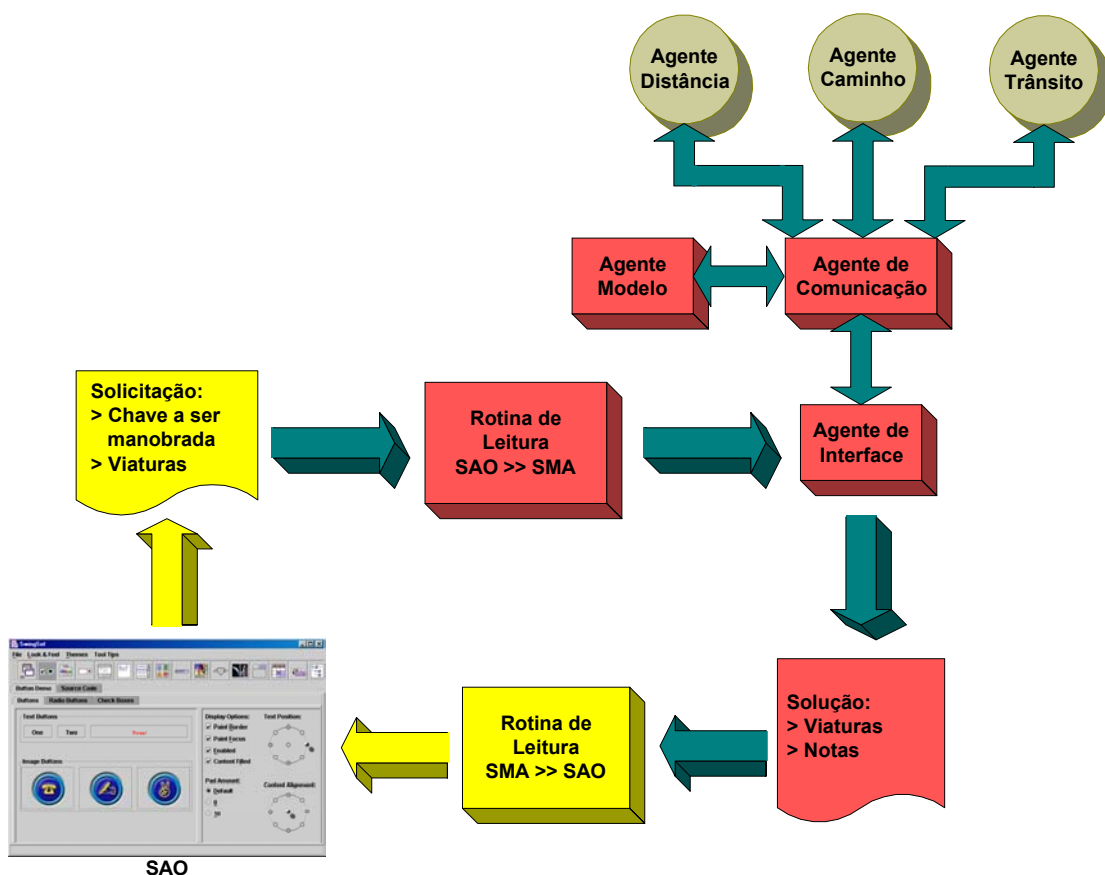


Figura E.1 – Fluxograma global de operação

O operador irá gerar uma solicitação através de uma interface apropriada no programa SAO, para isto o usuário deverá selecionar qual chave ele deseja operar numa lista de chaves e solicitar uma análise. Junto com a informação da chave é enviada uma lista da posição atual de todas as viaturas da CEB obtidas do sistema de localização via satélite deles.

De posse desses dados o programa SMA inicia o processo de busca, conforme explicitado no item 3.3.3. Foi implementada também uma rotina de controle de concorrência para esse programa nos mesmos moldes da rotina apresentada no item 5.6.

Terminada a análise o SMA escreve um arquivo SAIDA.TXT para o SAO com as informações das melhores viaturas para atender aquela ocorrência.

## E.2 Modelo de Dados

### E.2.1 Estrutura do Banco de Dados do Distrito Federal

Primeiramente, buscou-se organizar o mapa do Distrito Federal segundo o padrão adotado pela Secretaria do Estado de Desenvolvimento Urbano e Habitação (SEDUH) de Brasília, a qual divide o Distrito Federal (DF) em dezenove regiões administrativas: RA I – Brasília, RA II – Gama, RA III – Taguatinga, RA IV – Brazlândia, RA V – Sobradinho, RA VI – Planaltina, RA VII – Paranoá, RA VIII – Núcleo Bandeirante, RA IX – Ceilândia, RA X – Guará, RA XI – Cruzeiro, RA XII – Samambaia, RA XIII – Santa Maria, RA XIV – São Sebastião, RA XV – Recanto das Emas, RA XVI – Lago Sul, RA XVII – Riacho Fundo, RA XVIII – Lago Norte e RA XIX – Candangolândia.

Em seguida, para cada região o mapa do (DF) foi digitalizado a partir de um software específico, e foi desenvolvido um interpretador dos dados digitalizados para um banco de dados Access que contém as seguintes tabelas:

- Bairros – nessa tabela está contida o identificador do bairro (“IdBairro”) e o nome do Bairro (“Nome”), Figura E.2;
- Pontos – nessa tabela estão contidas todas as esquinas encontradas no mapa, armazenadas através de um identificador do ponto (“IdPonto”), meridiano de longitude WGr (“X”), paralelo de latitude sul (“Y”), ambos em graus, e um identificador se esse ponto é de conexão entre bairros (“Ligacao”), Figura E.3;
- Ruas – nessa tabela estão contidas todas as ruas encontradas no mapa, armazenadas através de um identificador da rua (“IdRua”), o nome da rua (“Nome”) e o código de endereçamento postal (“CEP”), Figura E.4;
- Quadras – nessa tabela estão contidas todas as quadras encontradas no mapa, armazenadas através de um identificador da quadra (“IdQuadra”), um identificador do bairro (“IdBairro”), um identificador da rua (“IdRua”), um identificador da esquina de origem (“IdDe”), um identificador da esquina de destino

(“IdPara”), o tamanho da quadra em metros (“Lenght”), e por fim o sentido de trânsito (“Sentido”), Figura E.5.

| IdBairro | Nome                                               |
|----------|----------------------------------------------------|
| 1        | Asa Norte                                          |
| 2        | Asa Sul                                            |
| 3        | Eixo Monumental                                    |
| 4        | Futuro Setor de Comercio                           |
| 5        | Setor de Armazenamento e Abastecimento Norte       |
| 6        | Setor de Autarquia e Setor de Embaixada Norte      |
| 7        | Setor de Clubes Esportivos Norte                   |
| 8        | Setor de Clubes Esportivos Norte'                  |
| 9        | Setor de Clubes Esportivos Sul                     |
| 10       | Setor Comercial, Setor Bancário e Setor de Autarqu |
| 11       | Setor Cultural Norte                               |
| 12       | Setor Cultural Sul                                 |
| 13       | Setor de Embaixada Sul                             |
| 14       | Areas Isoladas e Setor de Garagens Oficiais        |
| 15       | Areas Octogonais e Setor de Garagens e Veiculos    |
| 16       | Setor Comercial e Setor Bancário Norte             |
| 17       | Setor Hoteleiro Norte                              |
| 18       | Setor Hoteleiro Sul                                |
| 19       | Setor de Indústrias Gráficas                       |
| 20       | Setor de Mansões Isoladas                          |

Figura E.2 – Tabela de bairros do banco de dados da RA I – Brasília

| IdPonto | X          | Y          | Ligacao |
|---------|------------|------------|---------|
| 1       | -47,870341 | -15,782885 | 0       |
| 2       | -47,871275 | -15,781954 | 0       |
| 3       | -47,872061 | -15,782539 | 0       |
| 4       | -47,873021 | -15,783382 | 0       |
| 5       | -47,873527 | -15,785255 | 0       |
| 6       | -47,873746 | -15,783813 | 0       |
| 7       | -47,873318 | -15,783523 | 0       |
| 8       | -47,873059 | -15,782986 | 0       |
| 9       | -47,872474 | -15,782301 | 0       |
| 10      | -47,871607 | -15,781234 | 0       |
| 11      | -47,871083 | -15,780796 | 0       |
| 12      | -47,869926 | -15,779742 | 0       |
| 13      | -47,866536 | -15,777809 | 0       |
| 14      | -47,874611 | -15,785316 | 0       |
| 15      | -47,873468 | -15,786874 | 0       |
| 16      | -47,887594 | -15,783592 | 0       |
| 17      | -47,884332 | -15,783661 | 0       |
| 18      | -47,884048 | -15,783623 | 0       |
| 19      | -47,882361 | -15,783682 | 0       |
| 20      | -47,881168 | -15,78334  | 0       |
| 21      | -47,88117  | -15,783673 | 0       |
| 22      | -47,881203 | -15,784046 | 0       |
| 23      | -47,88052  | -15,783735 | 0       |
| 24      | -47,879839 | -15,783723 | 0       |
| 25      | -47,879004 | -15,783758 | 0       |
| 26      | -47,876844 | -15,783854 | 0       |
| 27      | -47,876849 | -15,782964 | 0       |
| 28      | -47,876846 | -15,7827   | 0       |
| 29      | -47,876991 | -15,780736 | 0       |
| 30      | -47,880755 | -15,786655 | 0       |

Figura E.3 – Tabela de pontos do banco de dados da RA I – Brasília



| IdRua | Nome                          | CEP      |
|-------|-------------------------------|----------|
| 1     | EIXO OESTE-NORTE              |          |
| 2     | EIXO RODOVIARIO NORTE         |          |
| 3     | L2-NORTE                      |          |
| 4     | EIXO LESTE-NORTE              |          |
| 5     | AV W-2 NORTE                  |          |
| 6     | AV W-3 NORTE                  |          |
| 7     | AV DAS NACOES                 |          |
| 8     | EIXO LESTE-SUL                |          |
| 9     | EIXO OESTE-SUL                |          |
| 10    | EIXO RODOVIARIO SUL           |          |
| 11    | L2-SUL                        |          |
| 12    | AV W-2 SUL                    |          |
| 13    | AV W-3 SUL                    |          |
| 14    | EST INDUSTRIAS GRAFICAS       |          |
| 15    | SES 813                       | 70200130 |
| 16    | SES 809                       | 70200090 |
| 17    | EIXO MONUMENTAL SI            |          |
| 18    | RODOVIA EPIA                  |          |
| 19    | RODOVIA EPTT                  |          |
| 20    | SEN                           | 70800400 |
| 21    | PONTE PRES COSTA E SILVA      |          |
| 22    | VIA HCE UM                    |          |
| 23    | R D                           |          |
| 24    | EST PARQUE CONTORNO DO BOSQUE |          |
| 25    | SRES Q 02 BLOCO H             | 70648080 |
| 26    | SRES Q 02 BLOCO D             | 70648040 |
| 27    | SRES Q 02 BLOCO T             | 70648200 |

Figura E.4 – Tabela de ruas do banco de dados da RA I – Brasília

| IdQuadra | IdBairro | IdRua | IdDe | IdPara | Lenght       | Sentido |
|----------|----------|-------|------|--------|--------------|---------|
| 1        | 1        | 0     | 1    | 2      | 143,89273310 | 2       |
| 2        | 1        | 0     | 2    | 3      | 106,32479092 | 2       |
| 3        | 1        | 0     | 3    | 4      | 139,06354182 | 2       |
| 4        | 1        | 0     | 4    | 5      | 215,19099753 | 2       |
| 5        | 1        | 0     | 5    | 6      | 329,07390922 | 2       |
| 6        | 1        | 0     | 6    | 7      | 56,010822993 | 2       |
| 7        | 1        | 0     | 7    | 4      | 35,436838441 | 2       |
| 8        | 1        | 0     | 7    | 8      | 65,829636284 | 2       |
| 9        | 1        | 0     | 8    | 9      | 98,590125299 | 2       |
| 10       | 1        | 0     | 9    | 10     | 150,60980295 | 2       |
| 11       | 1        | 0     | 10   | 11     | 74,268927637 | 2       |
| 12       | 1        | 0     | 11   | 12     | 170,47921324 | 2       |
| 13       | 1        | 0     | 12   | 13     | 421,64529649 | 2       |
| 14       | 1        | 0     | 13   | 2      | 685,25661380 | 2       |
| 15       | 1        | 0     | 14   | 15     | 215,77683737 | 2       |
| 16       | 1        | 0     | 16   | 17     | 349,1265323  | 2       |
| 17       | 1        | 0     | 17   | 18     | 30,681068806 | 2       |
| 18       | 1        | 0     | 18   | 19     | 180,6324032  | 2       |
| 19       | 1        | 0     | 19   | 20     | 133,19814544 | 2       |

Figura E.5 – Tabela de quadras do banco de dados da RA I – Brasília

As tabelas do banco de dados se relacionam entre si conforme mostrado na Figura E.6 abaixo:

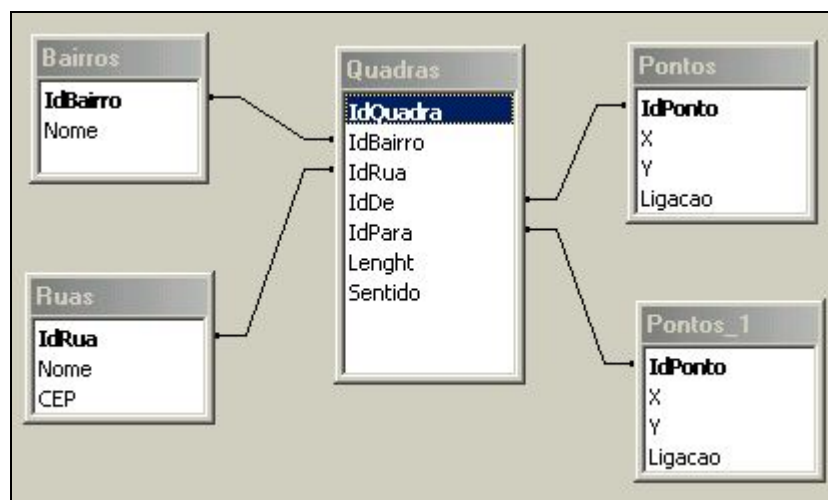


Figura E.6 – Estrutura de relacionamento das tabelas

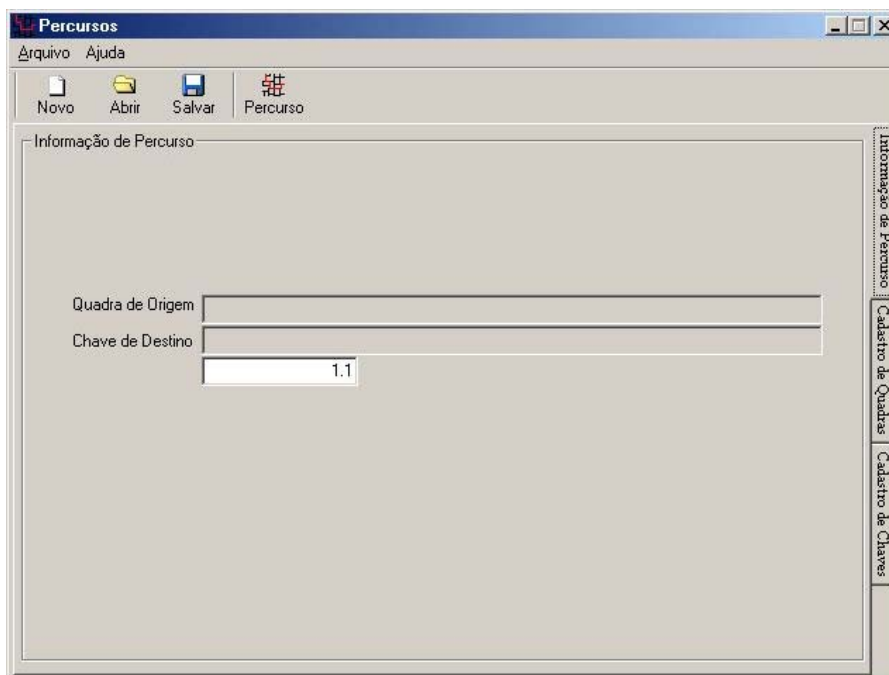
## E.2.2 Estrutura do Banco de Dados de Chaves

O banco de dados de chaves é adicionado à base de dados do DF e contém as seguintes informações

- Chave – Código da chave;
- Categoria ou Tipo – sendo “0” – NA de Recurso, “1” – NA de Socorro Externo, “2” – NA de Socorro Interno, 3 – NA e 4 – NF;
- Estado – Estado operativo atual da chave;
- Bairro – Bairro onde se localiza a quadra;
- Rua – Rua onde se localiza a chave;
- Quadra – Quadra onde se localiza a chave.

## E.3 Procedimentos Operacionais

O programa “Percurso” apresenta a seguinte janela principal para o usuário, Figura E.7.



**Figura E.7** – Janela principal do programa “Percurso”

O programa possui um menu com as seguintes opções típicas dos programas atuais: Novo, Abrir, Salvar, Sair, Tópicos e Sobre. Na janela principal o usuário ainda possui uma barra de ferramentas com botões para acessar algumas funções do programa: Novo, Abrir, Salvar e Percurso, o qual executa a análise do percurso necessário para executar a operação na chave desejada.

### **E.3.1 Cadastrando uma Cidade**

Para cadastrar uma cidade o usuário deverá cadastrar as quadras e as chaves separadamente acessando as suas tabelas “Cadastro de Quadras” e “Cadastro de Chaves”.

Ao acessar o cadastro de quadras o usuário encontrará uma nova barra de ferramentas abaixo da área de trabalho (Figura E.8) com as seguintes opções:

- Nova Quadra – Adiciona uma quadra na cidade existente;
- Editar Quadra – Edita a quadra selecionada da cidade existente;
- Remover – Remove a quadra selecionada da cidade existente.

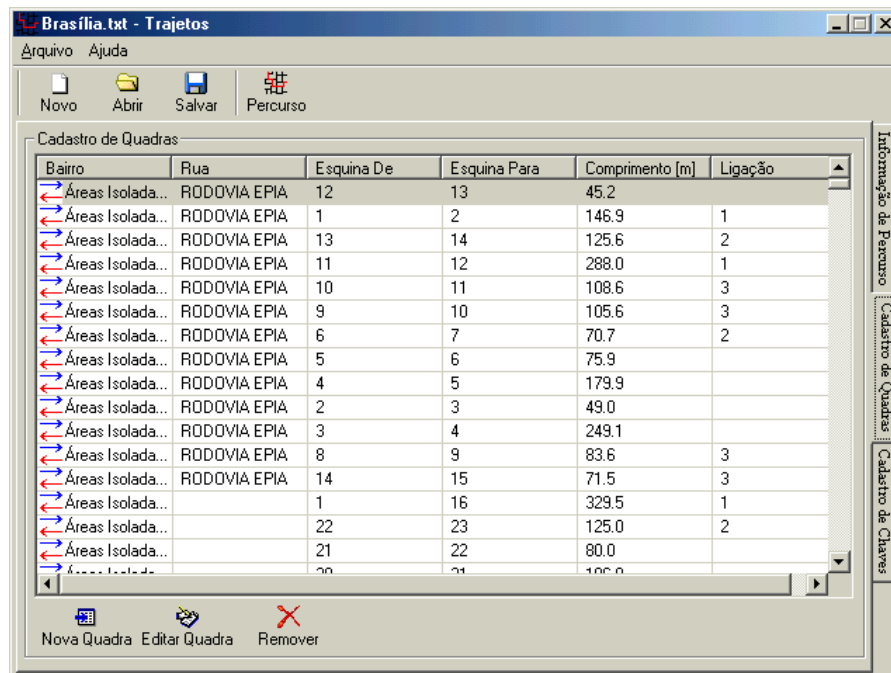


Figura E.8 – Área de cadastro de quadras

O usuário deverá inserir todos os dados das quadras pertencentes à cidade, clicando no botão “Nova Quadra” aparecerá a janela da Figura E.9.

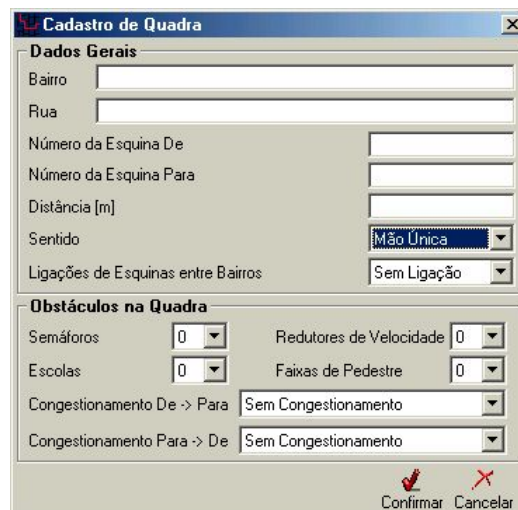


Figura E.9 – Janela para cadastro de quadras

Os dados gerais necessários para cadastrar uma quadra são:

- Bairro – Nome do bairro da cidade onde a quadra da rua se encontra;
- Rua – Nome da rua da quadra que está sendo adicionada;

- Número da Esquina De – Número da esquina de origem da quadra;
- Número da Esquina Para – Número da esquina de destino da quadra;
- Distância [m] – Distância da quadra em m;
- Sentido – Sentido de trânsito na quadra: “Mão Única” ou “Mão Dupla”;
- Ligações de Esquinas entre bairros – Indica se alguma esquina da quadra é uma ligação preferencial entre bairros: “Sem Ligação”, “Esquina De”, “Esquina Para”, “Ambas as Esquinas”.

Os dados de obstáculos na quadra necessários para cadastrar uma quadra são:

- Semáforos – Quantidade de semáforos existentes na quadra;
- Escolas – Quantidade de escolas existentes na quadra;
- Faixas de Pedestres – Quantidade de faixas de pedestres existentes na quadra;
- Redutores de Velocidade – Quantidade de redutores de velocidade existentes na quadra;
- Congestionamento De -> Para – Indica se há congestionamento na quadra no sentido “Esquina De” para “Esquina Para” no período: “Sem Congestionamento”, “Matutino”, “Vespertino”, “Noturno”, “Horários de Pico” e “Dia Todo”.

Esses dados serão utilizados como parâmetros de penalização para caminhos que contenham muitos desses itens. Dessa forma, o algoritmo de busca terá como avaliar não só a menor distância, bem como o caminho mais rápido.

Ao acessar o cadastro de chaves o usuário também encontrará uma barra de ferramentas abaixo da área de trabalho (Figura E.10) com as seguintes opções:

- Nova Chave – Adiciona uma quadra na cidade existente;
- Editar Chave – Edita a quadra selecionada da cidade existente;
- Remover – Remove a quadra selecionada da cidade existente.

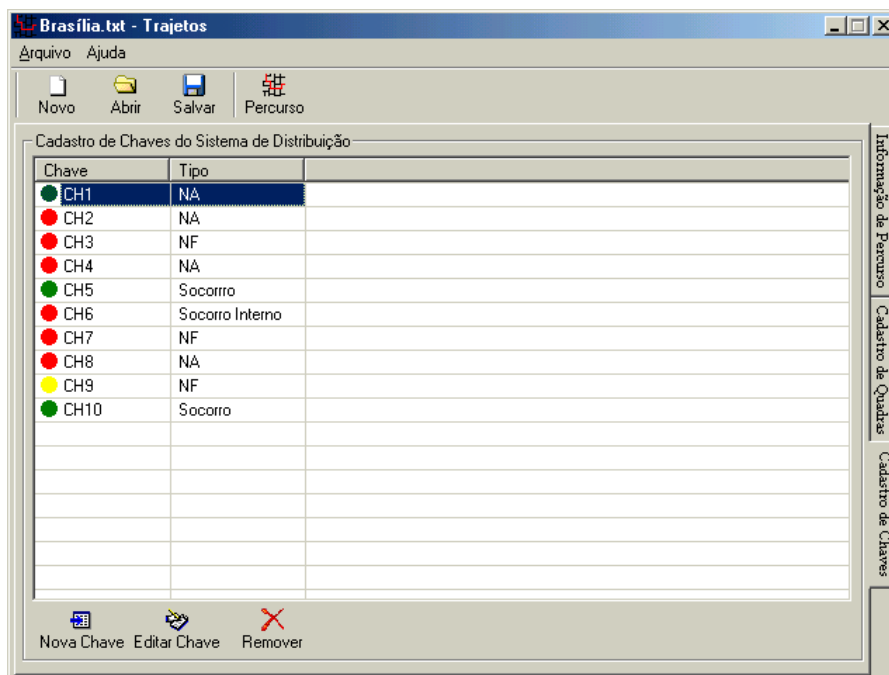


Figura E.10 – Área de cadastro de chaves

O usuário deverá inserir todos os dados das chaves pertencentes ao sistema de distribuição da cidade, clicando no botão “Nova Chave” aparecerá a janela da Figura E.11.

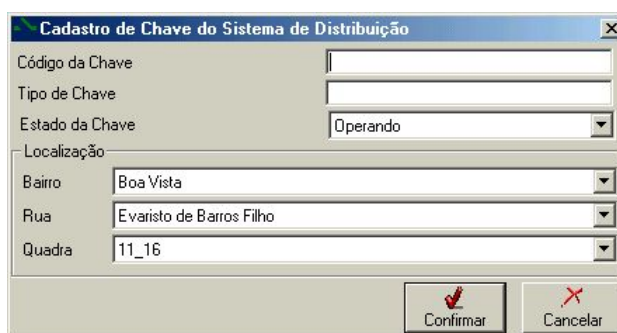


Figura E.11 – Janela para cadastro de chaves

Os dados necessários para cadastrar uma chave são:

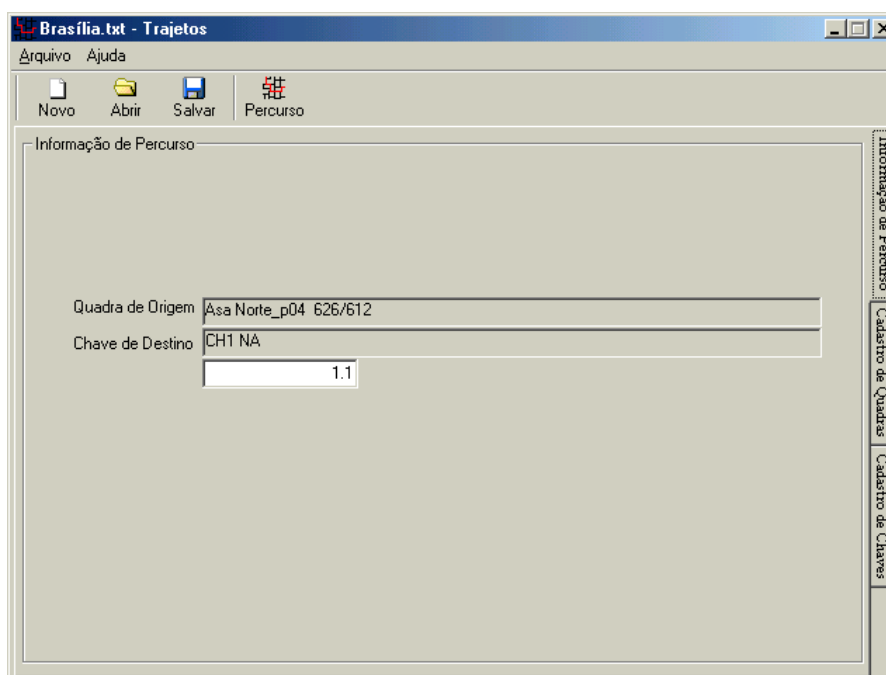
- Código da Chave – Código utilizado pela Concessionária para denominar a chave;
- Tipo da Chave – Tipo da chave;
- Estado da Chave – Estado operativo da chave: “Operando”, “Aberta” ou “Falha”;

- A localização da chave através da seleção do bairro, rua e quadra onde ela se encontra.

### E.3.2 Buscando um Percurso

O primeiro passo para buscar um percurso é selecionar no cadastro de quadras a quadra de origem, isto é feito com um clique em cima da linha desejada. Em seguida, o usuário deverá selecionar a chave de destino, isto é feito com o mesmo procedimento anterior.

O programa então apresentará esses dados na janela de dados de busca, Figura E.12.



**Figura E.12** – Dados para busca

Feito isso, o usuário deverá então acionar o programa, clicando no botão “Percurso”, para buscar os melhores percursos para locomover uma viatura da quadra de origem até a chave.

A resposta apresentada pelo programa está mostrada na Figura E.13.

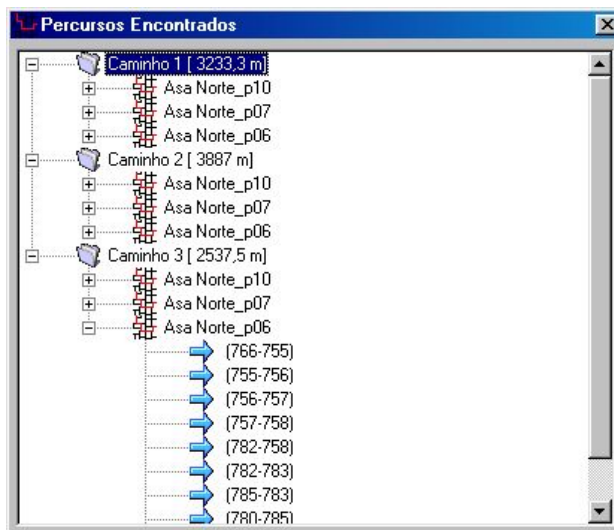


Figura E.13 – Resposta da busca de percurso

## E.4 Metodologia de Busca

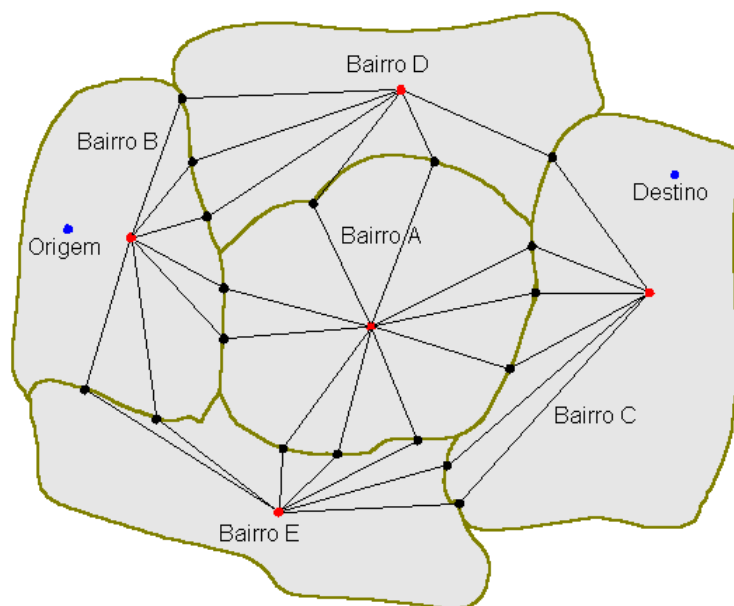
Como as bases de dados do DF ficaram grandes foi desenvolvida uma metodologia de busca para auxiliar o sistema especialista contido no “Agente Caminho”, apresentado no item 3.3.2, a qual é responsável por otimizar o tempo de busca das soluções na base de dados. Essa metodologia será explicada a seguir:

### PASSO 1

Dada a configuração inicial do problema, a quadra de origem (localização da viatura) e a quadra de destino (localização da chave), o programa irá buscar na base de dados os bairros onde essas quadras estão, Figura E.14.

Para tal o “Agente Caminho” busca todas as esquinas preferenciais (marcadas com ●) e cria um ponto central no bairro (marcado com ●) e constrói quadras que ligam esses pontos, como se fosse um novo mapa.





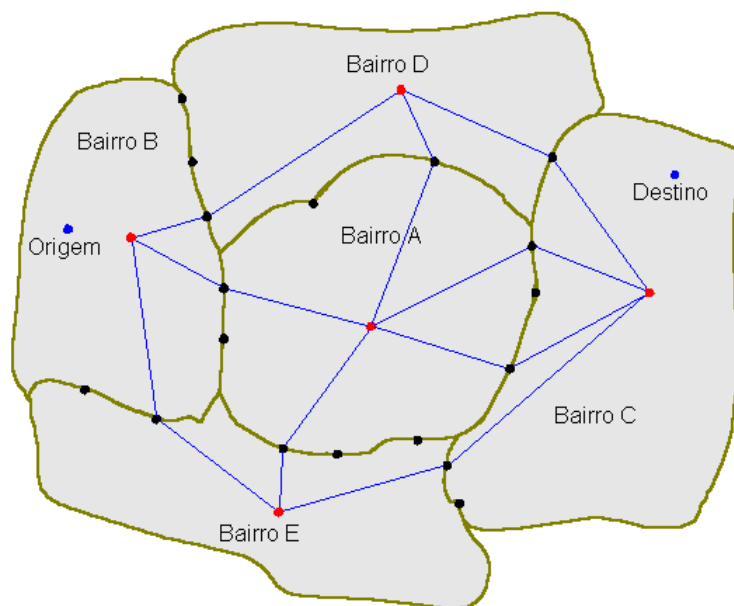
**Figura E.14** – Buscando o caminho entre bairros para a solução

## PASSO 2

Então o “Agente Caminhos” executa uma análise de todas as possibilidades e determina os melhores caminhos para cruzar os bairros (marcados com —), Figura E.15. O critério empregado leva em conta a distância.

### Soluções Encontradas:

- Caminho: Bairros B → D → C;
- Caminho: Bairros B → D → A → C;
- Caminho: Bairros B → D → A → C;
- Caminho: Bairros B → D → A → E → C;
- Caminho: Bairros B → A → C;
- Caminho: Bairros B → A → C;
- Caminho: Bairros B → A → D → C;
- Caminho: Bairros B → A → E → C;
- Caminho: Bairros B → E → C;
- Caminho: Bairros B → E → A → C;
- Caminho: Bairros B → E → A → C;
- Caminho: Bairros B → E → D → C.

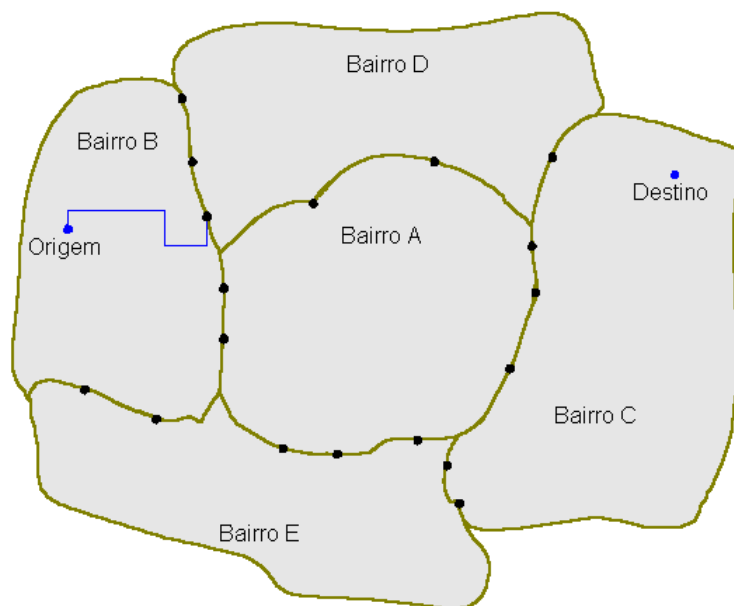


**Figura E.15** – Determinando os melhores caminhos entre bairros

Para cada uma das soluções encontradas o “Agente Caminho” irá ordenar uma busca por solução para o sistema inteligente de busca seguindo os passos abaixo:

**PASSO 3: Caminho: Bairros B → D → C**

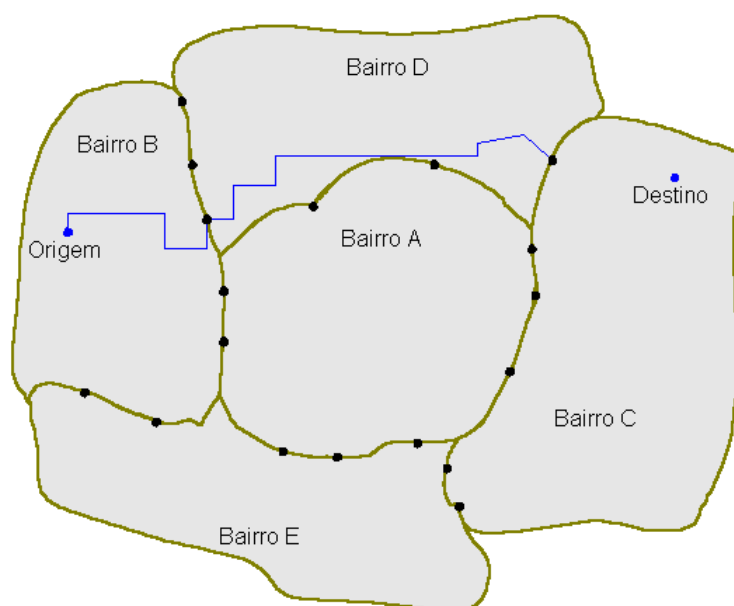
O sistema especialista irá buscar o melhor caminho do ponto de origem até a esquina preferencial entre o bairros B e D da solução em análise, Figura E.16.



**Figura E.16** – Melhor caminho para a primeira etapa da solução

**PASSO 4: Caminho: Bairros B → D → C**

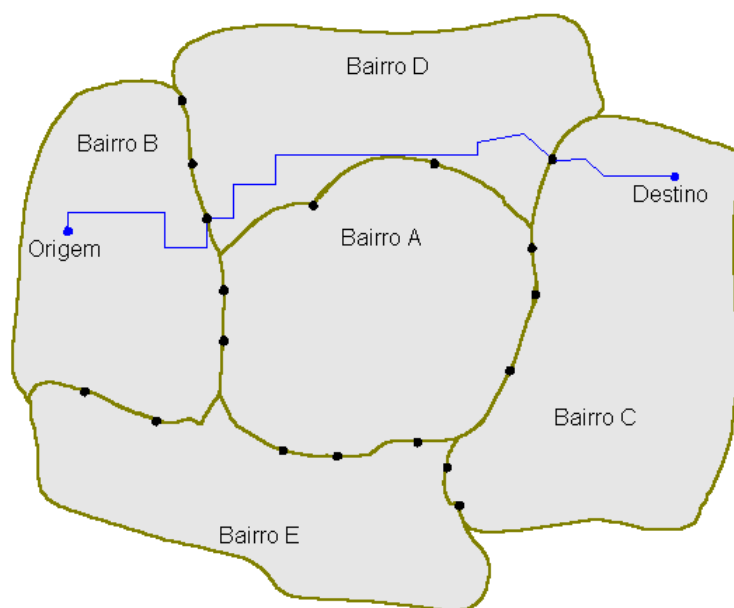
O sistema especialista irá buscar o melhor caminho para cruzar o bairro D, da esquina preferencial entre os bairros B e D até a esquina preferencial entre os bairros D e C, Figura E.17.



**Figura E.17** – Melhor caminho para a segunda etapa da solução

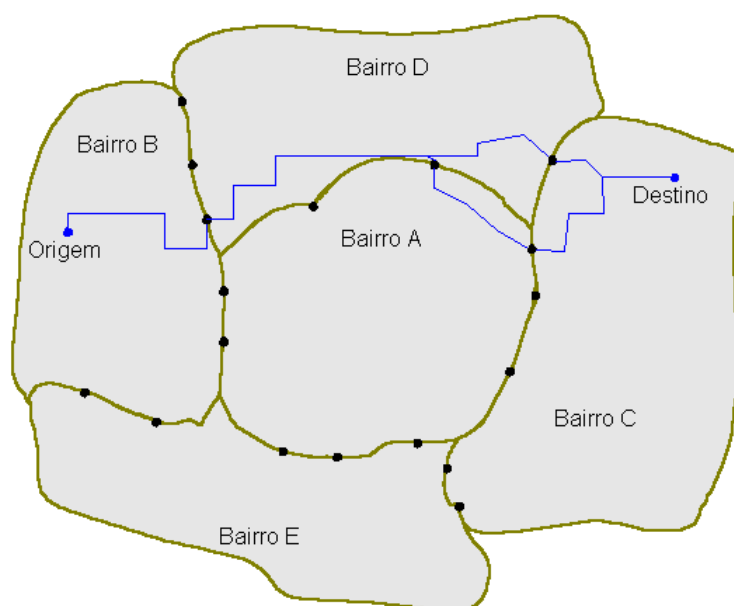
**PASSO 5: Caminho: Bairros B → D → C**

O sistema especialista irá buscar o melhor caminho para ir da esquina preferencial entre os bairros D e C até o ponto de destino, Figura E.18.



**Figura E.18** – Melhor caminho para a terceira etapa da solução

Em seguida ele irá ordenar buscas para encontrar os caminhos para a segunda solução encontrada no passo 2 (Figura E.19) e assim sucessivamente.



**Figura E.19** – Melhor caminho para segunda solução do passo 2