

TESE

1181

Universidade Federal de Itajubá
na de Pós-Graduação em Engenharia Elétrica

**Um Sistema Baseado em Conhecimento para
Diagnóstico, Solução e Aprendizado de Falhas em
Controladores Lógicos Programáveis**

Dissertação apresentada
à Universidade Federal
de Itajubá para obtenção
do Título de Mestre em
Engenharia Elétrica.

Marcelo Pinheiro Werneck

Itajubá

Agosto/2002

**Universidade Federal de Itajubá
Programa de Pós – Graduação em Engenharia Elétrica**

**Um Sistema Baseado em Conhecimento para
Diagnóstico, Solução e Aprendizado de Falhas em
Controladores Lógicos Programáveis**

Dissertação apresentada
à Universidade Federal
de Itajubá para obtenção
do Título de Mestre em
Engenharia Elétrica.

Marcelo Pinheiro Werneck

Itajubá

Agosto/2002



CLASS. 681.5.01 (043.2)
CUTTER 11491.2
TOMBO. 1191



Universidade Federal de Itajubá
Programa de Pós – Graduação em Engenharia Elétrica

**Um Sistema Baseado em Conhecimento para
Diagnóstico, Solução e Aprendizado de Falhas em
Controladores Lógicos Programáveis**

Dissertação apresentada
à Universidade Federal
de Itajubá para obtenção
do Título de Mestre em
Engenharia Elétrica

Área de Concentração:
Automação de Sistemas
Elétricos Industriais.

Autor:
Marcelo Pinheiro
Werneck.

Orientador:
Luiz Edival de Souza.

Itajubá

Agosto/2002

Agradecimentos

Resumo

Abstract

Lista de Figuras

Lista de Tabelas

Lista de
Abreviações

Capítulo 1 - Introdução

Capítulo 2 - Revisão de Conceitos

2.1 Inteligência Artificial

2.2 Sistemas Orientados por Regras

2.3 Fatores e Regras em Conexão

2.4 Desenvolvimento de Raciocínios

2.5 Sistemas Inteligentes e Humanos

2.6 Sistemas Móveis com Visão Elétrônica, em Raciocínio Baseado em Conexões

2.7 Sistemas com Raciocínio

2.8 Comunicação entre Programas

Capítulo 3 - Definição do Problema

3.1 A Manutenção de Manutenção

3.1.1 Pessoal de Manutenção

3.1.2 Pessoal de Manutenção Eletrônica

3.1.3 Pessoal de Manutenção no Chão

3.1.4 Manutenção Funcional Física ou Funcionalista

A meus pais, Paulo, Marcia e Regina, pelo apoio que sempre me deram.

A Liliana, minha alma gêmea, pelo seu amor e carinho.

Sumário

Agradecimentos	ix
Resumo	x
Abstract	xi
Lista de Figuras	xii
Lista de Tabelas	xiv
Lista de Abreviações	xvi
Capítulo 1 - Introdução.....	1
Capítulo 2 - Revisão de Conceitos.....	4
2.1. Inteligência Artificial.....	4
2.2. Sistemas Baseados em Regras.....	5
2.3. Raciocínio Baseado em Casos.....	7
2.4. Aprendizado de Máquinas.....	9
2.5. Sistemas Inteligentes Híbridos.....	10
2.6. Sistemas Híbridos com Sistema Baseados em Regras e Raciocínio Baseado em Casos.....	11
2.7. Sistemas para Diagnóstico.....	11
2.8. Controlador Lógico Programável.....	15
Capítulo 3 - Definição do Problema.....	18
3.1. A Motivação.....	18
3.1.1. Pessoal de Manutenção Inexperiente.....	18
3.1.2. Pessoal de Manutenção Escasso.....	19
3.1.3. Ausência de Metodologia para Resolução de Falhas no CLP.....	19
3.1.4. Mudança de Função ou Perda do Especialista.....	20

3.1.5. Consultas Desnecessárias à Assistência Técnica do Fabricante.....	20
3.2. Problemas com o Controlador Lógico Programável.....	20
3.2.1. Análise da forma atual do Diagnóstico e Solução de Falhas em CLP.....	22
3.3. Problemas com a Aquisição, Tratamento e Estruturação do Conhecimento a respeito do Diagnóstico de Falhas em CLP.....	24
3.4. Problemas de Implementação Computacional do Processo de Diagnóstico, Solução e Aprendizado de Falhas em CLP.....	24
Capítulo 4 - A Solução.....	25
4.1. Uma Proposta de Descrição do Processo Diagnóstico, Solução e Aprendizado de Falhas em CLP.....	25
4.2. Uma Metodologia para Aquisição, Tratamento e Estruturação do Conhecimento referente ao Diagnóstico e Solução de Falhas em CLP.....	26
4.2.1. Aquisição do Conhecimento.....	28
4.2.2. Tratamento do Conhecimento.....	29
4.2.3. Estruturação do Conhecimento.....	30
4.3. Proposta de um Sistema Baseado Regras e Casos para Diagnóstico, Solução e Aprendizado de Falhas em CLP.....	32
4.3.1. Descrição dos Componentes Sistema Proposto.....	32
4.3.1.1. O Sub-sistema Baseado em Regras.....	33
4.3.1.1.1. Regras Tipo 1, ou Regras para Diagnóstico de Falhas	33
4.3.1.1.2. Regras Tipo 2, ou Regras de Determinação de Contexto e Subcontexto de operação do CLP.....	33
4.3.1.1.3. Regras Tipo 3, ou Regras de Determinação da Versão de ocorrência da falha no CLP.....	34

4.3.1.1.4. Regras Tipo 4, ou Regras para Determinação dos Conjuntos de Ações Corretivas e Causas.....	34
4.3.2. O Método de Inferência para Processamento das Regras.....	35
4.4. O Sub-sistema Baseado em Casos.....	37
4.4.1. O Algoritmo de Seleção e Ordenação dos Casos Armazenados.....	38
4.4.2. O Algoritmo de Aprendizado e Reforço do Aprendizado.....	41
4.4.3. Algoritmo de Transição entre Conhecimento Geral (BAC) e Conhecimento Específico (BCO).....	41
4.5. Funcionamento Sistema.....	44
4.5.1. Pré Configuração.....	44
4.5.2. Consultando o Sistema.....	44
4.6. Pré – requisitos para Aplicação da Solução Proposta.....	45
Capítulo 5 - Aplicando a Metodologia Proposta.....	48
5.1. O CLP Selecionado.....	48
5.2. Aplicando a Metodologia para Aquisição, Tratamento e Estruturação do Conhecimento referente ao Diagnóstico e Solução de Falhas em CLP.....	51
5.2.1. Aquisição do Conhecimento.....	51
5.2.1.1. Apresentação da Falha, Causas Prováveis e Ações Recomendas pelo Fabricante.....	51
5.2.1.2. Identificação Visual da Falha.....	53
5.2.1.3. Determinação dos Sintomas Típicos da Falha.....	53
5.2.1.4. Determinação do Sintoma Visual Complementar..	54
5.2.1.5. Pré Tratamento do Conhecimento referente a Causas e Ações Corretivas.....	54
5.2.2. Tratamento do Conhecimento.....	55

5.2.2.1. Determinação da Hierarquia de Diagnóstico de Falhas no CLP.....	55
5.2.2.2. Determinação dos Contextos, Subcontextos e Versões de Ocorrência da Falha.....	55
5.2.2.3. Determinação dos Recursos Aplicáveis para Solução da Falha.....	55
5.2.2.4. Determinação das Combinações de Recursos Aplicáveis aos Contextos, Subcontextos e Versões.....	57
5.2.2.5. Seleção das Combinações de Recursos Aplicáveis de acordo com os Contextos, Subcontextos e Versões.....	59
5.2.3. Estruturação do Conhecimento.....	60
5.2.3.1. Classificação e Seqüência de Aplicação das Ações Corretivas aplicáveis à Solução de Falhas.....	60
5.2.3.2. Determinação dos Conjuntos de Ações Corretivas e Causas.....	60
5.2.3.3. Associação dos Conjuntos de Ações Corretivas e Causas aos Contextos/Subcontextos/Versões e Combinações de Recursos Aplicáveis.....	61
5.2.3.4. Criação da Árvore de Decisões.....	63
5.2.3.5. Conversão da Árvore de Decisões em Regras de Produção.....	63
5.2.3.6. Determinação dos Estados Indicadores de Falha Resolvida.....	63
Capítulo 6 - A Implementação Computacional – O CLPSolve.....	65
6.1. A Implementação Computacional: O CLP Solve.....	65
6.1.1. Visão Geral do CLPSolve.....	65
6.1.2. Módulo de Configuração do CLP.....	68
6.1.3. Módulo de Entrada dos Sintomas de Falha.....	68
6.1.4. Módulo de Apresentação do Diagnóstico, Ações Corretivas e Causas da Falha.....	70
6.1.5. Módulo de Edição de Bases (BAC e/ou BCO).....	72

6.1.6. Módulo de Acompanhamento de Desempenho.....	73
6.2. Um exemplo de consulta ao CLPSolve.....	73
6.2.1. A Configuração.....	74
6.2.2. Entrada dos Sintomas.....	74
6.2.3. Consulta ao CLPSolve.....	75
6.2.4. Encerramento da Consulta.....	76
Capítulo 7 - Conclusão	77
Referências Bibliográficas	79
Anexo 1 - Algoritmos do Capítulo 4 representados em Pseudocódigo.....	81
Anexo 2 - Regras para Diagnóstico e Solução da Falha de Comunicação no Canal 1.....	89
Anexo 3 - Conjuntos de Ações e Causas aplicáveis a Falha de Comunicação no Canal 1.....	109
Anexo 4 - Árvore de decisões para resolução da falha de comunicação no canal 1.....	117
Anexo 5 - Busca realizada na árvore de decisões até a escolha da BAC ou BCO 42, no exemplo apresentado no Cap. 6 – Item 6.2.....	123

Agradecimentos

Ao Professor Luiz Edival de Souza, pela orientação, dedicação e paciência.

Ao Professor Wilton Ney, pelos conselhos e ensinamentos que tornaram meu caminho menos tortuoso.

Aos Professores Prado e Grandinetti, pela confiança que sempre demonstraram em mim.

Aos Professores Dinarti, Kassic, Perin, Hans, Assunção, Barreto e Zanchin, da UFSC, por terem me ensinado o que é ser Engenheiro.

A Universidade de Taubaté, pelo apoio que permitiu o desenvolvimento dessa dissertação.

A todos aqueles que direta ou indiretamente, contribuíram para a realização desse trabalho.

Resumo

Programable logic controller (PLC) control of automated industrial processes. Diagnosis, solution and learning of PLC failures and repair.

O controlador lógico programável (CLP) tem largo emprego no controle de processos industriais automatizados. O diagnóstico, solução e aprendizado de falhas em CLP, constitui um problema relevante, agravado pela escassez ou despreparo da mão de obra empregada para tal objetivo.

Inicialmente essa dissertação propõe uma metodologia para aquisição, tratamento e estruturação do conhecimento disponível sobre o diagnóstico de falhas, ações corretivas aplicáveis e causas da falha. A metodologia proposta tem como objetivo adequar o conhecimento proveniente da experiência do especialista e das diversas fontes de consulta, como manuais de manutenção, a uma implementação computacional.

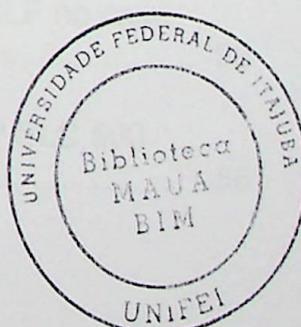
Dado que o conhecimento referente as falhas possíveis no equipamento foi devidamente estabelecido, é proposta uma descrição para o processo de diagnóstico e solução de falha em CLP. Uma vez diagnosticado o Estado Operacional e/ou Falha, tal descrição considera que são propostas ações corretivas baseadas nos recursos para teste e/ou substituição do componente defeituoso disponíveis no contexto da indústria ou setor onde o CLP está instalado. Quando um conjunto de ações corretivas propostas elimina a falha diagnosticada, a causa da falha é apresentada e armazena-se o conjunto Falha – Ações Corretivas – Causa na forma de um Caso. Na repetição da mesma falha, inicialmente são sugeridas ações corretivas baseadas nos casos anteriores, caso esses existam, que obtiveram sucesso em solucioná-la.

A aplicação da metodologia para aquisição, tratamento e estruturação do conhecimento unida à descrição proposta para o processo de diagnóstico são utilizados no desenvolvimento de uma ferramenta computacional baseada em Regras e Casos, para apoio ao diagnóstico, solução e aprendizado de falhas em CLP, o CLPSolve.

Os resultados da aplicação da metodologia proposta em uma falha de comunicação e da utilização do CLPSolve em condições de laboratório, demonstraram a viabilidade da aplicação da ferramenta computacional.

Abstract

Programmable logic controls (PLC's) are widely employed in automatic control of industrial processes. Diagnosis, solution and learning of PLC failures represent an important problem, which is aggravated by the lack of experienced people in PLC maintenance. First, this work presents a systematic approach to deal with acquisition, processing, and structuring of the available knowledge on failure diagnosis, troubleshooting procedures and possible causes of failure. This approach aims at implementing a computational solution, using the experience of technicians and engineers and the available literature (troubleshooting manuals, technical reports, papers, etc.). After establishing the knowledge related to equipment failure, our work provides a complete description of the diagnosis process and failure repair. Having the diagnosis of the *Operational State and/or Failure*, the system describes also the necessary troubleshooting procedures, based on the resources for the test and/or replacement of the defective part. Repaired the failure by the set of troubleshooting actions, the cause of the failure is presented and the set "*Failure-Troubleshooting actions-Cause*" is stored as a "Case". If there is a failure recurrence, repair actions, based on the stored cases that proved to be successful, are suggested. The application of this approach to acquire, process and structure the knowledge, added to the description of the diagnosis process, were used to develop the *CLPSolve*, a computational tool based on Rules and Case, suited for diagnosis, solution and learning of PLC failures. The results obtained from the application of our proposed approach in a communication failure and from the *CLPSolve* utilization in laboratory conditions, proved that the computational tool is completely viable.



Lista de Figuras

2.1 - Blocos componentes de um SBR.....	7
2.2 - Processo de diagnóstico e aprendizado no RBC.....	9
2.3 - Exemplo de uma árvore de decisões para diagnóstico de uma falha de um veículo motorizado.....	14
2.4 - Partes componentes de um CLP.....	17
4.1 - Passos para implementação da metodologia proposta.....	27
4.2 - Fluxo de avaliação das regras para determinação de uma falha, seu contexto/subcontexto/versão e escolha da base de conhecimento adequada.....	36
4.3 - Fluxograma apresentando o algoritmo de seleção e ordenação de casos armazenados.....	39
4.4 - Fluxograma apresentando o algoritmo de aprendizado e reforço do aprendizado de casos.....	42
4.5 - Fluxograma apresentando o algoritmo de transição entre consulta a BAC e consulta a BCO.....	43
4.6 - Funcionamento geral do sistema proposto.....	46
5.1 - O CLP Rockwell Automation – Allen Bradley SLC 500 UCP 03 tipo 2 (SLC 5/03) – Painel Frontal.....	48
5.2 - Detalhe dos indicadores luminosos do painel da UCP SLC 5/03 tipo 2...	49
5.3 - O conjunto ferramenta de programação, módulo PIC, módulo AIC e CLP, assim como a conexão entre esses elementos, utilizando os cabos adequados.....	53
5.4 - A configuração visual assumida pelos LEDs do painel do CLP no caso da falha de comunicação em questão.....	53
5.5 - Fluxograma da hierarquia de diagnóstico de falhas na UCP SLC 500 5/03.....	56

5.6 - Estado assumido pelos indicadores luminosos do painel do CLP indicando falha resolvida.....	64
6.1 - Interface principal do CLPSolve.....	66
6.2 - A interface visual do Módulo de Configuração do CLPSolve.....	69
6.3 - Detalhe da interface visual do Módulo de Configuração com as informações relevantes à falha de comunicação no CN 1.....	69
6.4 - A interface visual do Módulo de Entrada dos Sintomas de Falha do CLPSolve.....	70
6.5 - Exemplo da interface para alteração de estados assumidos pela interface visual de entrada de Sintomas da Falha.....	70
6.6 - A interface visual do Módulo de Apresentação do Diagnóstico, Ações Corretivas e Causas do CLPSolve.....	71
6.7 - Interface visual do Módulo de Edição de Bases do CLPSolve.....	72
6.8 - A interface visual do Módulo de Acompanhamento de Desempenho do CLPSolve.....	73
6.9 - Estado final da interface visual após a alteração dos estados dos LEDs relevantes ao diagnóstico da falha e a pergunta feita ao usuário durante a consulta ao CLPSolve.....	75

Lista de Tabelas

2.1 - Exemplo de formato de um caso num sistema de RBC.....	8
3.1 - Um exemplo de como o fabricante pode apresentar um estado de falha... ..	22
4.1 - Formato de um caso presente numa BCO.....	39
5.1 - Características técnicas da UCP escolhida.....	49
5.2 - Significado do estados assumidos pelos indicadores luminosos do painel frontal da UCP SLC 5/03 tipo 2.....	49
5.3 - Estado do indicador luminoso da fonte de alimentação do CLP.....	49
5.4 - Reprodução da forma como o fabricante do CLP SLC 500 UCP 5/03 tipo 2 apresenta a falha de comunicação no canal 1 (DH 485), causas prováveis e ações corretivas.....	52
5.5 - Sintomas típicos de falha de comunicação no canal 1, fornecidos pelo fabricante.....	53
5.6 - Contextos, Subcontextos e Versões de ocorrência da falha de comunicação no CN 1.....	57
5.7 - Combinações de Recursos Aplicáveis ao Contexto 1, Subcontextos 1.2 e 1.3 da falha de comunicação no CN 1.....	58
5.8 - Combinações de Recursos Aplicáveis ao Contexto 2, Subcontextos 2.1 da falha de comunicação no CN 1.....	59
5.9 - Seleção das Combinações de Recursos Aplicáveis de acordo com os Contextos e Subcontextos e Versões da falha de comunicação no CN 1..	60
5.10 - Seleção das ações corretivas de acordo com as combinações de recursos e sua associação as BACs.....	62
6.1 - Configurações relativas ao agrupamento “Ferramenta de Programação e Interfaces de Comunicação” do módulo de Configuração do CLP.....	74
6.2 - Configurações relativas ao agrupamento “Comunicação em Rede” do módulo de Configuração do CLP.....	74

6.3 - Estado assumido pelos LEDs relevantes ao diagnóstico de falha de comunicação no canal 1.....	75
--	----

SAC - Módulo de Automação e Controle

SAC - Placa de Automação

SAC - Placa de Interface de Comunicação

SAC - Base de Caso de Comunicação

SAC - Base de Caso de Interface de Comunicação

SAC - Base de Caso de Interface de Comunicação

SAC - Placa de Interface de Comunicação

SDA - Sistemas de Automação

SDA - Módulo conversor de meio-fio de comunicação

SDA - Recocador de automação

SDA - Recocador de automação

SDA - Unidade Central do Processamento

Lista de Abreviaturas

AIC	- Módulo isolador para conexão do CLP em rede.
BAC	- Bases de Ações e Causas.
BCC	- Base de Casos Consultados.
BCO	- Base de Casos Ocorridos.
BCOO	- Base de Casos Ocorridos Ordenada.
BDC	- Base de Dados de Configuração.
BFE	- Bases de Falha e Estado.
Bit	- Dígito Binário.
CLP	- Controlador Lógico Programável.
CN	- Canal de Comunicação.
FMEA	- Técnicas de Análise de Falha e Efeito.
IA	- Inteligência Artificial.
IAC	- Inteligência Artificial Conexionalista.
IAE	- Inteligência Artificial Evolucionária.
IAS	- Inteligência Artificial Simbólica.
LED	- Díodo Emissor de Luz.
PC	- Computador Pessoal.
PIC	- Módulo conversor de meio físico de comunicação.
RBC	- Raciocínio Baseado em Casos.
SBR	- Sistemas Baseados em Regras.
UCP	- Unidade Central de Processamento.

Capítulo 1

Introdução

O controlador lógico programável (CLP) tem largo emprego no controle de processos industriais automatizados. O diagnóstico, a solução e o aprendizado de falhas em CLP constitui um problema relevante, agravado pela escassez ou despreparo da mão de obra empregada para tal objetivo.

A questão da formação de pessoal encarregado da manutenção em plantas industriais é abordado por Alves e Gebral [1]. Nesse trabalho, os autores atentam para o aumento na complexidade dos equipamentos industriais e para a diminuição nos quadros de funcionários, o que gera a necessidade de criação de meios para garantir o armazenamento, acesso e gerenciamento do conhecimento relativo ao processo industrial.

O desenvolvimento de sistemas inteligentes para apoio ao diagnóstico e proposição de ações tem sido foco de diversos trabalhos. Ballakrishan e Honavar [2] analisam diversas formas de aplicação de sistemas inteligentes em diagnóstico de falhas. Wilson [3] propõe um sistema baseado em regras para auxilio na configuração e montagem de minicomputadores comerciais da série DPS6 da Honeywell – Bull Ltd. , chamado HONE ("Honeywell configuring Expert"). Nesse trabalho, destacam-se a estruturação das regras utilizadas pelo sistema baseado em regras e a forma de seleção das regras a serem aplicadas. Para cada passo da montagem do minicomputador são determinados contextos e subcontextos (que correspondem a fase atual do processo de montagem). Uma vez determinado o contexto e o subcontexto atual, um conjunto de ações de montagem e configuração são apresentadas ao responsável pela montagem, que as aplica, levando a uma alteração do contexto e subcontexto, sendo ordenadas novas ações, até que a montagem termine.

Kamel, McCaffrey e Metzler [4] descrevem todo o ciclo de desenvolvimento de um sistema baseado em regras para auxilio ao diagnóstico de falhas em sistemas eletrônicos de controle armas de fragatas da Marinha dos Estados Unidos. Destaca-se nesse artigo toda análise das falhas atuais no processo de diagnóstico, como ausência do especialista na manutenção do sistema, necessidade freqüentes e dispendiosas de contato com a assistência técnica do fabricante do sistema, dificuldade de treinamento do especialista, além de outras.

Aamodt e Plaza [5] apresentam as diferenças fundamentais entre os sistemas baseados em casos das demais técnicas de IA aplicadas a solução de problemas. Segundo os autores, sistemas inteligentes baseados em casos, ao invés de basearem-se apenas no conhecimento geral sobre o domínio de um problema, são capazes de utilizar o conhecimento específico sobre ocorrências do problema já experimentadas no passado. Outra característica é o caráter incremental do aprendizado de novas soluções a medida que novos problemas ocorram, sendo que tais soluções tornam-se imediatamente disponíveis para aplicação em problemas futuros.

A aprendizagem e seus modelos tem sido abordados em diversos estudos psicológicos. O trabalho de Roos [6] analisa e apresenta evidências empíricas a respeito do domínio do conhecimento específico e de situações previamente experimentadas na solução de problemas por seres humanos e na forma de memorização desses casos. Adicionalmente, o trabalho de Kolodner [7,8] indica o uso de casos passados como o método de solução de problemas predominante entre especialistas em determinado domínio.

O Sistema Baseado em Conhecimento para Diagnóstico, Solução e Aprendizado de Falhas em CLP, o CLP Solve, objeto dessa dissertação, foi desenvolvido com o objetivo de diagnosticar Estados de Operação e/ou Falha em CLPs, propondo ações corretivas de forma a orientar o operador do equipamento controlado por um CLP, na alteração do estado operacional ou solução da falha apresentada. Uma vez que as ações propostas obtiveram sucesso em resolver a falha, são armazenadas na forma de casos, apresentando-se como solução na repetição da mesma falha.

O Capítulo 2 apresenta uma revisão de conceitos relevantes ao entendimento do trabalho.

O Capítulo 3 apresenta a motivação e descreve os problemas envolvidos no diagnóstico, solução e aprendizado de falhas em CLP.

O Capítulo 4 inicialmente propõe uma descrição para o processo de diagnóstico e solução de falha em CLP, que será implementada pela ferramenta computacional objeto do trabalho. É apresentada a seguir uma metodologia para a aquisição, tratamento e estruturação do conhecimento referente ao diagnóstico e solução de falhas em CLP. Um sistema baseado em regras e casos que implementa a descrição propostas para o processo de diagnóstico, solução e aprendizado de falhas em CLP, utilizando o conhecimento tratado pela metodologia é então proposto.

O Capítulo 5 aplica o método proposto para a aquisição, tratamento e estruturação do conhecimento referente ao diagnóstico e solução de falhas em CLP a um CLP real.

O Capítulo 6 apresenta a implementação computacional realizada, descrevendo o CLPSolve, uma ferramenta computacional para diagnóstico, solução e aprendizado de falhas em CLP.

O Capítulo 7 apresenta as conclusões e desenvolvimentos futuros que podem orientar novas pesquisas e aprimorar os resultados obtidos nesse trabalho.

O capítulo 8 é sobre previsão: uma visão das possibilidades e desafios da inteligência artificial no campo da previsão.

1. Introdução à IA

A inteligência Artificial (IA) busca prover máquinas com a capacidade de realizar algumas habilidades mentais do ser humano. São essas habilidades como: a capacidade de aprendizagem, a de variedades de aquisições, que permitem a realização de tarefas não necessariamente algorítmicas. As ações das tecnologias de IA modernas podem envolver a sensopercepção (como leitura e processamento de dados sensoriais), a linguagem (como as verbais e categóricas) e a razão (que consiste de conceitos abstratos, generalizações e hipóteses).

O estudo da IA é importante para o aperfeiçoamento das técnicas de trabalho para a melhoria das condições de vida e produtividade humana.

Em termos de tecnologia, a IA pode auxiliar na realização de tarefas complexas no lugar do operador humano, liberando-o de atividades rotineiras ou estagnadas. Também pode aumentar a eficiência do trabalho em processos que envolvem coletâneas sofisticadas. Permite, ainda, que determinados processos sejam realizados por muitas pessoas, sem que haja necessidade de grande especialização.

A IA possui diversos domínios de aplicação. No caso dessa tese, é utilizada para o campo industrial.

Descreve-se é o processo de detectar anomalias ou mal funcionamento, causado num conjunto de sistemas. A informação de anomalias separadas são interpretadas e a causa dessas situações é identificada.

As anomalias podem ser divididas em três tipos principais:

Capítulo 2

Revisão de Conceitos

O capítulo a seguir apresenta uma revisão dos conceitos e técnicas relevantes citados e utilizados nos próximos capítulos dessa dissertação.

2.1 Inteligência Artificial

“A Inteligência Artificial (IA) busca prover máquinas com a capacidade de realizar algumas atividades mentais do ser humano. Em geral são máquinas com algum recurso computacional, de variadas arquiteturas, que permitem a implementação de rotinas não propriamente algorítmicas. As atividades realizadas por essas máquinas podem envolver a sensopercepção (como tato, audição e visão), as capacidades intelectuais (como aprendizado de conceitos e de juízos, raciocínio dedutivo e memória), a linguagem (como as verbais e gráficas), a atenção (decisão no sentido de concentrar atividades sobre um determinado estímulo)”[9].

O estudo de IA contribui não somente para o aperfeiçoamento das máquinas, mas também para a motivação em se conhecer melhor as próprias atividades mentais humanas.

Em termos de tecnologia, IA permite que máquinas possam realizar tarefas complexas no lugar do operador humano, liberando-o de atividades enfadonhas, insalubres ou inseguras. Também pode aumentar a eficiência do humano em sua interação com equipamentos sofisticados. Permite, ainda, que conhecimentos possam ser compartilhados por muitas pessoas, sem que haja necessidade de consultas a especialistas.

A IA possui diversos domínios de aplicação. No caso dessa dissertação, são de especial interesse:

- Diagnóstico: é o processo de determinar uma falha ou mau funcionamento, baseado num conjunto de sintomas. A informação de entrada (os sintomas) são interpretados e a causa desses sintomas é a saída apresentada.

- Reparação: é a correção do defeito encontrado na fase de diagnóstico. Pode se tornar um problema muito difícil, pois freqüentemente implica em uma seqüência de ações, cada uma produzindo um certo efeito sobre o mundo exterior, o que por sua vez modifica a seqüência de ações.

Atualmente distinguem-se três abordagens para IA: a Simbólica (IAS), a Conexionalista (IAC) e a Evolucionária (IAE). No caso dessa dissertação, serão utilizadas técnicas da IAS.

Na IAS, espera-se que o conhecimento sobre um problema particular e das técnicas para a sua solução, possam, através da manipulação de conhecimentos básicos e da imitação do modo de raciocínio usado por humanos, resultar na solução do mesmo. Para alcançar tal objetivo "a IAS deve ser usada quando o problema é bem definido, se tenha uma boa idéia de como ele seria resolvido e o modo de encontrar a solução seja explícito"[10]. Sistemas inteligentes baseados em IAS são chamados normalmente Sistemas Baseados em Conhecimento.

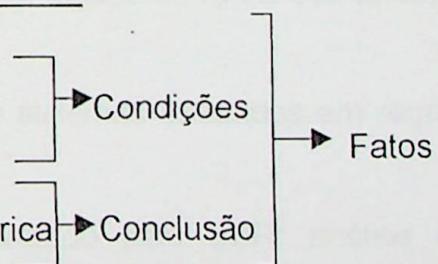
2.2 Sistemas Baseados em Regras

Sistemas Baseados em Regras (SBR) são programas de computador que procuram atingir soluções de determinados problemas do mesmo modo que especialistas humanos, sob as mesmas condições, procedem.

As regras, muitas vezes chamadas de regras de produção, são um conjunto de fatos, condições (ou premissas) e conclusões, organizados na forma " SE <condições> ENTÃO <conclusão> ", com a possibilidade de inclusão de conectivos lógicos (E, OU, NÃO) relacionando as diversas condições do domínio de conhecimento sobre o qual versa o SBR. A seguir apresenta-se um exemplo de regra:

Regra

SE Indicador de CLP Energizado = apagado
 E Estado do Fusível = bom
 OU Temperatura = normal
 ENTÃO
 Ação = Observar nível de tensão da rede elétrica



Tradicionalmente, um sistema baseado em regras é formado pelos seguintes blocos, conforme figura 2.1:

- Base de Conhecimento: representa de forma computacional a informação (fatos e regras) que um especialista utiliza para resolver um problema.

- Máquina de Inferência: é a parte do SBR responsável pelas deduções sobre a base de conhecimentos.
- Interface com o usuário: responsável pela interação com o usuário do sistema.
- Base de Dados de Fatos: armazena os fatos fornecidos pelo usuário na consulta em curso. Difere da Base de Conhecimento por, normalmente, conter informações transitórias, dependentes do contexto e momento no qual realiza-se a consulta ao sistema.

Uma consulta à um SBR inicia-se com a aquisição de fatos atuais a respeito do problema a ser resolvido. Tais fatos podem ser obtidos de várias formas: através de perguntas diretas ao usuário, consultas a bancos de dados, observação de valores atribuídos a variáveis de um programa, etc... Tais fatos são armazenados de forma temporária no SBR. A máquina de inferência compara os fatos atuais com os fatos contidos nas condições das diversas regras contidas na base de conhecimento. Quando um conjunto de fatos atuais coincide com um conjunto de fatos contidos nas condições de uma regra, essa é considerada verdadeira (é aceita ou disparada) fazendo com que o sistema considere verdadeira a conclusão da regra. A conclusão de uma regra gera um novo fato, que pode ser a condição de uma outra regra e assim por diante. Várias regras podem ser aceitas durante uma consulta ao sistema, podendo o mesmo chegar a diversas conclusões numa mesma consulta.

A máquina de inferência pode acessar a base de conhecimento utilizando duas estratégias de busca: o encadeamento para frente (encadeamento guiado por fatos) ou o encadeamento para traz (encadeamento guiado por objetivos). No caso dessa dissertação será utilizado o encadeamento para frente.

No encadeamento para frente, as regras são aceitas baseadas nos fatos atuais apresentados ao sistema, independente de objetivos predeterminados, ou seja, dado que os fatos atuais coincidem com as condições da regra sob análise, essa é aceita, obtendo-se assim uma conclusão.

Segundo [10] o ciclo de desenvolvimento de sistemas baseados em regras pode ser separado nas seguintes etapas:

- Estudo da relevância do problema a ser tratado pelo SBR: análise de oportunidade.
- Análise Funcional: definição das funcionalidades desejadas do SBR.
- Conceituação: criação de um modelo capaz de resolver o problema, incluindo a definição das ferramentas a serem usadas.

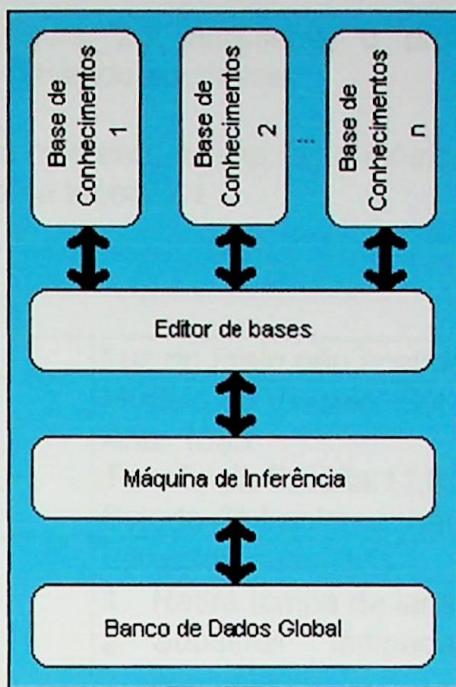


Figura 2.1 – Blocos componentes de um SBR.

- Aquisição do conhecimento: obtenção do conhecimento necessário à implementação da base de conhecimento. Segundo [4] o conhecimento pode vir de várias fontes, como entrevistas com o especialista, consulta a manuais, relatórios de manutenção, árvores de diagnóstico, etc...
- Implementação: consiste na confecção do código que exercerá o papel da máquina de inferência, criação das bases de conhecimento e das interfaces necessárias com o processo sob análise e o usuário.
- Teste: consiste em conferir se as conclusões alcançadas pelo SBR, dado um conjunto de fatos, coincidem com as alcançadas por um especialista no domínio de aplicação do SBR.
- Manutenção: atualização da base de conhecimento, de forma a incluir novos conhecimentos ou retirar conhecimentos desnecessários ou redundantes.

2.3 Raciocínio Baseado em Casos

"Raciocínio Baseado em Casos (RBC) é uma forma de usar soluções utilizadas no passado em um tipo particular de problema para resolver novos problemas." [11] Pode ser considerado análogo à forma como um especialista humano resolve um problema através do uso de sua experiência passada.

O RBC é aplicável para a solução de diversos problemas, mas tem provado ser extremamente popular na área de diagnóstico, especialmente em aplicações

de suporte técnico. A figura 2.2 apresenta o processo de diagnóstico e aprendizado no raciocínio baseado em casos.

Um exemplo de caso referente a uma falha elétrica num veículo automotivo teria formato apresentado na tabela 2.1.

Caso nº: 10	
Número de ocorrências: 3	
Sintomas:	Luz do Freio não acende
Informações Complementares:	Modelo do Veículo: Gol 1000 Ano: 1999 Tensão da Bateria: 11,8V Estado da Lanterna: não danificada
Diagnóstico:	Lâmpada queimada.
Ações Corretivas:	1. Retira tampa da lanterna traseira. 2. Substituir lâmpada queimada por uma equivalente. 3. Recolocar tampa da lanterna traseira.

Tabela 2.1 – Exemplo de formato de um caso num sistema de RBC.

Ao ocorrer uma falha num modelo diferente de carro (um Golf, por exemplo) cujos sintomas fossem os mesmos (Luz de freio não acende), a base de casos seria consultada e o caso acima apresentado como uma provável solução, mesmo sem o equipamento ser exatamente igual. Se houvessem outros casos semelhantes (ou mesmo iguais) esses também seriam apresentados como possíveis soluções.

Segundo [11] um sistema de diagnóstico utilizando o RBC tem um ciclo de desenvolvimento composto das seguintes etapas:

- levantar os casos passados disponíveis;
- escolher uma forma de representação para os casos;
- determinar como os sintomas e características do caso atual devem ser comparados com os dos casos passados;
- determinar como os casos passados, considerados parecidos com o atual, devem ser ordenados para serem verificados;
- determinar como verificar se o caso escolhido após a verificação dos diversos candidatos é o verdadeiro diagnóstico;

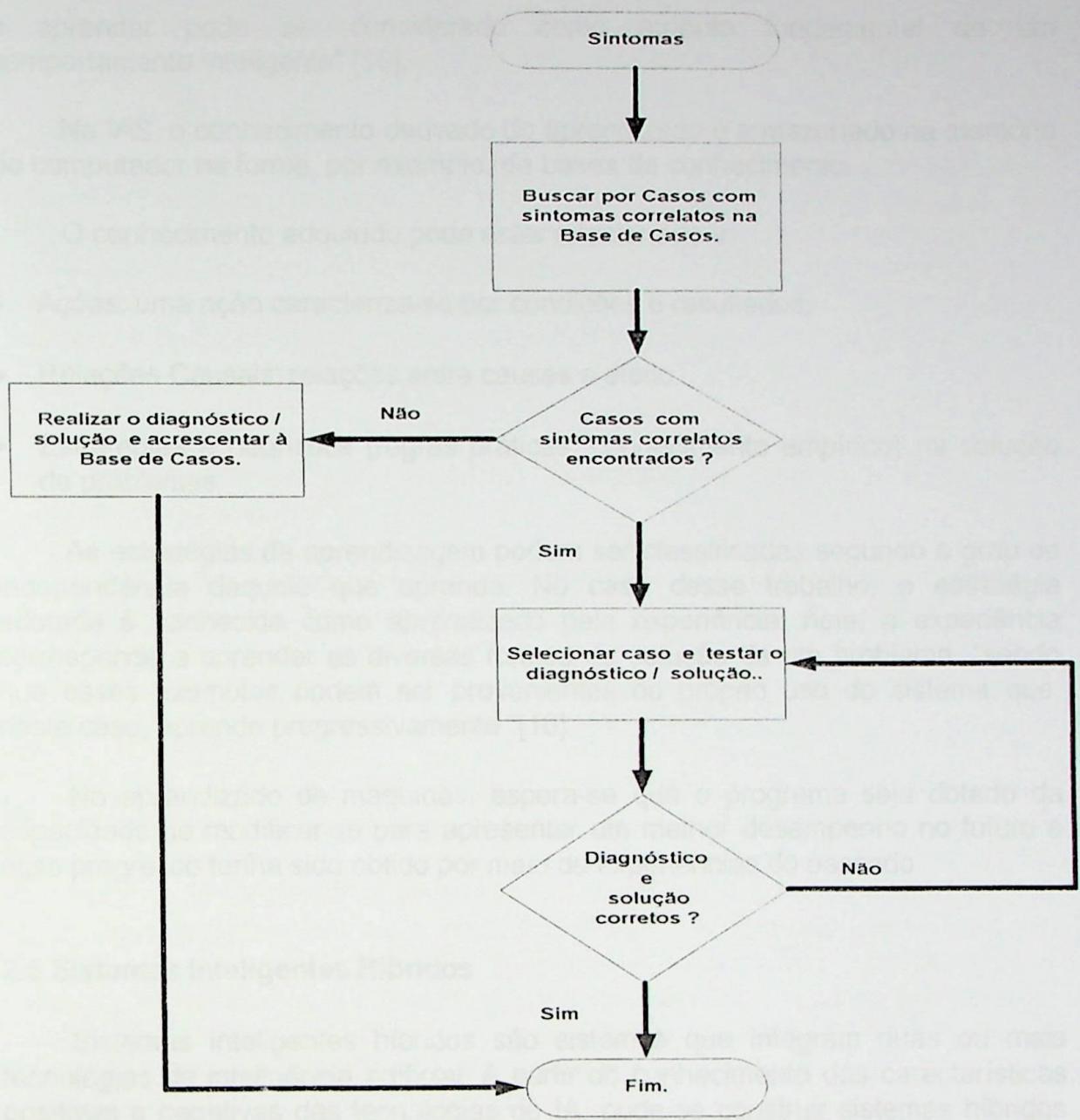


Figura 2.2 - Processo de diagnóstico e aprendizado no RBC.

- determinar como um novo caso é adicionado a base de casos;
- determinar a forma que a interface com o usuário deve assumir;



2.4 Aprendizado de Máquinas

"Aprender é o ato que produz um comportamento diferente a um estímulo externo devido à excitações recebidas no passado e é de certa forma sinônimo de aquisição de conhecimento. Em IA é comum se falar de aprendizado pela máquina

e aprender pode ser considerado como atributo fundamental de um comportamento inteligente” [10].

Na IAS, o conhecimento derivado do aprendizado é armazenado na memória do computador na forma, por exemplo, de bases de conhecimento.

O conhecimento adquirido pode estar relacionado a:

- Ações: uma ação caracteriza-se por condições e resultados.
- Relações Causais: relações entre causas e efeito.
- Estratégias e heurística (regras práticas, conhecimento empírico) na solução de problemas.

As estratégias de aprendizagem podem ser classificadas segundo o grau de independência daquele que aprende. No caso desse trabalho, a estratégia adotada é conhecida como aprendizado pela experiência; nele, a experiência corresponde a aprender as diversas formas de solução de um problema, “sendo que esses exemplos podem ser provenientes do próprio uso do sistema que, neste caso, aprende progressivamente.”[10]

No aprendizado de máquinas, espera-se que o programa seja dotado da capacidade de modificar-se para apresentar um melhor desempenho no futuro e esse progresso tenha sido obtido por meio de experiências do passado.

2.5 Sistemas Inteligentes Híbridos

Sistemas inteligentes híbridos são sistemas que integram duas ou mais tecnologias de inteligência artificial. A partir do conhecimento das características positivas e negativas das tecnologias de IA, pode-se construir sistemas híbridos que diminuem as limitações e tiram proveito das características positivas de cada tecnologia, permitindo produzir sistemas mais poderosos.

Existem cinco diferentes modelos de implementação de sistemas híbridos: modelos individuais, transformacionais, acoplamento fraco, acoplamento forte e integração total. No caso dessa dissertação, são interessantes os modelos transformacionais e de integração total.

Modelos transformacionais começam como um tipo de sistema inteligente e terminam como outro tipo. Apresentam a vantagem de serem desenvolvidos rapidamente e requererem, no final, manutenção de apenas um sistema. A forma de transformar uma tecnologia em outra ainda é objeto de estudo. Uma mudança significativa no sistema pode entretanto requerer um novo esforço de desenvolvimento, novo processo de transformação, e o sistema final tem todas as

limitações da tecnologia resultante. Determinar qual técnica de IA será usada para inicio e qual a desejada ao final será função do problema a ser resolvido [12].

Modelos totalmente integrados compartilham estruturas de dados e formas de representação do conhecimento. O raciocínio pode ser realizado cooperativamente ou por um dos componentes, nesse caso chamado de controlador. As principais vantagens vislumbradas são o aumento na capacidade de solução de problemas e a possibilidade de criação de sistemas com capacidades de adaptação, generalização, tolerância a ruído, justificativa de resultados e dedução lógica; porém resultam em sistemas de maior complexidade [12].

2.6 Sistemas Híbridos com Sistemas Baseados em Regras e Raciocínio Baseado em Casos

O RBC vem sendo utilizado com SBR de diversas formas. Devido a similaridade das duas tecnologias no tocante ao domínio de problemas, são alternativas naturais quer como sistemas independentes para o aprendizado a respeito de um problema, quer como uma forma para adquirir conhecimento para implementação do outro, sendo um deles o sistema final.

Sistemas utilizando o RBC superam uma série de limitações observadas pelos engenheiros do conhecimento nos SBR, tais como custo e demora na aquisição do conhecimento e o fato de sistemas especialistas não disporem de um mecanismo automático de aprendizado ou de melhoria do seu desempenho a medida que são utilizados; outra limitação refere-se ao fato que cada vez que o SBR encontra o mesmo problema, o mesmo processo de busca de fatos e avaliação de regras deve ser refeito. Um sistema baseado em casos apresentaria casos avaliados como semelhantes, até que esses esgotem-se ou seu grau de semelhança com o problema em tratamento atinja seu limite inferior.

2.7 Sistemas para Diagnóstico

Sistemas para diagnóstico destinam-se a atividades de busca e correção de falhas a partir da análise de sintomas do problema apresentado pelos equipamentos ou processos no domínio do sistema.

A medida da qualidade de um sistema para diagnóstico pode ser feita através da avaliação de alguns itens [11]:

- construção eficiente: o sistema de diagnóstico deve ser construído com o menor custo e no mínimo tempo necessário.
- facilidade de manutenção: o sistema deve ser construído de forma a permitir manutenções com baixo custo, dado que o equipamento objeto do sistema

pode ser alterado ao longo do tempo e novos problemas podem ser descobertos ao longo do uso do equipamento.

- abrangência adequada: o sistema deve abranger a maior quantidade de falhas passíveis de ocorrência no equipamento.
- respostas corretas: o sistema deve emitir diagnósticos corretos para a maior parte das falhas cobertas.
- mínimo esforço para o usuário: o sistema deve ser construído de forma a evitar que o usuário realize testes repetidos ou desnecessários. Além disso, esse deve ser capaz de diagnosticar o problema com o menor número de informações possível.
- tempo de diagnóstico apropriado: o sistema deve ser capaz de fornecer o diagnóstico correto no menor tempo possível.
- relação custo / benefício: o sistema deve apresentar uma relação custo benefício favorável quando comparado aos métodos de diagnóstico existentes. Essa relação pode ser medida na forma de economia em horas de engenharia para manutenção do sistema ou redução de tempo do equipamento parado”[11].

Toda atividade de diagnóstico segue essencialmente as tarefas descritas a seguir:

- identificação do problema: consiste em notar que algo não está funcionando da maneira devida. De acordo com o equipamento isso pode ser notado através de ruídos estranhos, leituras de grandezas fora das especificações ou mesmo alarmes sonoros e visuais.
- localização da falha: uma vez detectado que há um problema, é necessário que o local onde esse esteja ocorrendo seja localizado.
- identificação da falha: equipamentos são normalmente formados por partes mecânicas, elétricas/eletrônicas e, em alguns casos, programas que exercem funções de controle, logo, é necessário identificarmos qual das partes está apresentando a falha, de forma a restringirmos o universo de diagnóstico.
- diagnóstico da falha: uma vez identificado a parte defeituosa, é necessário descobrir o componente defeituoso.
- reparo: são as ações a serem tomadas, numa ordem coerente, para que o componente causador da falha seja substituído, trocado ou reconfigurado, de forma a restituir o funcionamento normal do equipamento.

A principal forma de representar uma estratégia de diagnóstico é através do uso de árvores de diagnóstico, sendo essa uma representação gráfica na forma de fluxograma, que contém: os sintomas apresentados, testes que devem ser aplicados ao equipamento defeituoso e diagnósticos (conclusões) provenientes dos resultados dos testes. A figura 2.3 apresenta um exemplo de uma árvore de decisões para diagnóstico de uma falha de um veículo motorizado.

As vantagens das árvores de diagnóstico são mostradas a seguir [11] :

- podem ser facilmente entendidas.
- são usadas em engenharia para descrever procedimentos para busca e solução de falhas num formato comum aos engenheiros que realizam testes em sistemas. Além disso são uma forma simples de compartilhamento de conhecimento entre o projetista de sistema e o responsável por sua manutenção.
- aplicação em diversos tipos de diagnóstico, independente do sistema ou equipamento sob análise, a árvore de diagnóstico é geral o suficiente para representar a estratégia de solução a ser adotada.

Apesar das qualidades das árvores de diagnóstico, em diversos casos seu uso se torna inadequado [11]:

- a árvore de diagnóstico torna-se muito grande: quando isso ocorre, sua descrição tradicional (em papel ou quadro sinótico) começa a tornar-se difícil de seguir, levando o usuário a perder-se na leitura da árvore ou pior, realizar testes pertencentes a outros ramos da árvore, o que pode levar a resultados completamente incoerentes com a falha presente. A árvore também pode tornar-se repetitiva o que ocorre quando os mesmos testes ou subtestes repetem-se várias vezes para o mesmo equipamento.
- a árvore de diagnóstico muda freqüentemente: alterar a árvore de diagnóstico pode ser um processo lento e exposto a erros, pois demanda alterações na estrutura gráfica da árvore. Além disso, quando se trata de diagnóstico de equipamentos dependentes de configuração feita pelo usuário, a árvore precisa ser particularizada para cada configuração, o que demanda tempo e custo relativo a horas de engenharia.
- a árvore é uma “floresta de pequenas árvores”: ocorre quando diferentes tipos de problemas podem ocorrer com um mesmo equipamento e esses não estão interrelacionados. O grande desafio nesse caso é descobrir em qual das pequenas árvores está a falha. Esse problema é comum em equipamentos de informática. Nesse podemos ter:

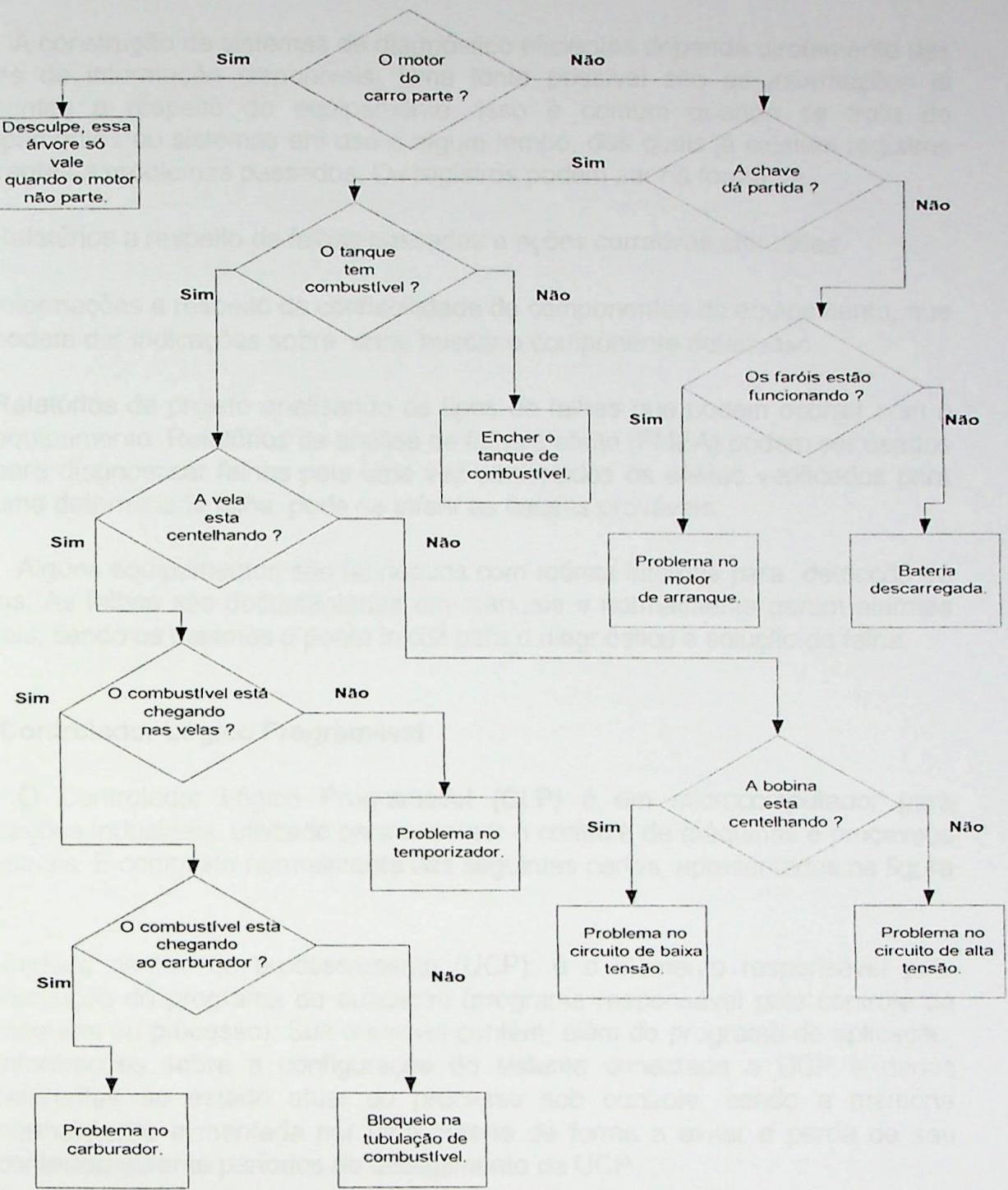


Figura 2.3 - Exemplo de uma árvore de decisões para diagnóstico de uma falha de um veículo motorizado.

1. Problemas com componentes: placas defeituosas, equipamentos desconectados.
2. Problemas de comunicação em rede: o computador local está desconectado, o roteador inativo, o repetidor defeituoso, o servidor de dados inativo.

A construção de sistemas de diagnóstico eficientes depende diretamente das fontes de informação disponíveis. Uma fonte possível são as informações já existentes a respeito do equipamento. Isso é comum quando se trata de equipamentos ou sistemas em uso a algum tempo, dos quais já existem registros referentes a problemas passados. Os registros podem ser na forma de:

- Relatórios a respeito de falhas passadas e ações corretivas efetuadas.
- Informações a respeito da confiabilidade de componentes do equipamento, que podem dar indicações sobre onde buscar o componente defeituoso.
- Relatórios de projeto analisando os tipos de falhas que podem ocorrer com o equipamento. Relatórios de análise de falha e efeito (FMEA) podem ser usados para diagnosticar falhas pois uma vez conhecidos os efeitos verificados para uma determinada falha, pode-se inferir as causas prováveis.

Alguns equipamentos são fabricados com rotinas internas para detecção de falhas. As falhas são documentadas em manuais e normalmente geram alarmes visuais, sendo os mesmos o ponto inicial para o diagnóstico e solução da falha.

2.8 Controlador Lógico Programável

O Controlador Lógico Programável (CLP) é um microcomputador para aplicações industriais, utilizado para executar o controle de máquinas e processos industriais. É composto normalmente das seguintes partes, apresentadas na figura 2.4:

- unidade central de processamento (UCP): é o elemento responsável pela execução do programa de aplicação (programa responsável pelo controle da máquina ou processo). Sua memória contém, além do programa de aplicação, informações sobre a configuração do sistema conectado a UCP e dados referentes ao estado atual do processo sob controle, sendo a memória normalmente alimentada por uma bateria de forma a evitar a perda de seu conteúdo durante períodos de desligamento da UCP.
- portas de comunicação: são os elementos responsáveis pela troca de dados entre o CLP, estações de programação e interfaces homem – máquina. Apresentam características físicas e elétricas bem definidas e formas de trocar dados definidas por protocolos de comunicação disponibilizados pelo fabricante. Podem estar integradas a UCP ou separadas em equipamentos dedicados.
- fonte de alimentação: elemento responsável por adequar a natureza e nível de tensão da rede de alimentação à natureza e níveis de tensão adequados ao funcionamento do CLP.

- cartões de entrada: são os elementos responsáveis por adquirir informações provenientes do estado da máquina ou processo sob controle. Uma vez adquirida a informação, essa é armazenada de forma digital na memória da UCP do CLP, que utiliza esses dados para tomar decisões.
- cartões de saída: são equipamentos responsáveis por transformar as decisões tomadas pelo programa contido na UCP em ações, de forma a alterar condições de operação da máquina ou processo.
- chassis: estrutura física na qual a UCP, portas de comunicação, fonte de alimentação e cartões de entrada e saída são montados.
- ferramenta de programação: microcomputador , normalmente portátil, que contém um programa que permite monitoração do estado do CLP e da execução do programa de aplicação.

O CLP executa o programa de aplicação de forma cíclica. Um ciclo de operação do CLP é composto, a grosso modo, de uma etapa dedicada a leitura dos valores presentes nos cartões de entrada e sua armazenagem na forma de bits ou palavras binárias na memória do CLP. Terminada a etapa de leitura das entradas, inicia-se a etapa de execução do programa de aplicação, que, dadas os estados das entradas armazenados na memória do CLP e a lógica do programa, toma decisões que são armazenadas na memória do CLP. Encerrada a execução do programa, todas as decisões tomadas são transformadas em ações que são aplicadas ao processo, mantendo ou alterando o estado desse.

O controlador lógico programável apresenta os seguintes estados de operação:

- Modo de Programação: modo de operação no qual o CLP não realiza a leitura das entradas, não executa o programa de aplicação e não aplica ao processo as decisões tomadas pela avaliação das condições lógicas contidas no programa de aplicação .
- Modo de Execução: modo de operação no qual o CLP realiza a leitura das entradas, executa o programa de aplicação e aplica ao processo as decisões tomadas pela avaliação das condições lógicas contidas no programa de aplicação.
- Modo Remoto: permite que, via ferramenta de programação, o estado de operação do controlador seja alterado para modo de Programação ou modo de Execução.

Os estados de operação são definidos pela posição de uma chave colocada no painel da CPU do controlador.

O CLP é largamente utilizado no parque industrial brasileiro, sendo a detecção e solução de falhas em seus componentes a atividade de interesse nessa dissertação.

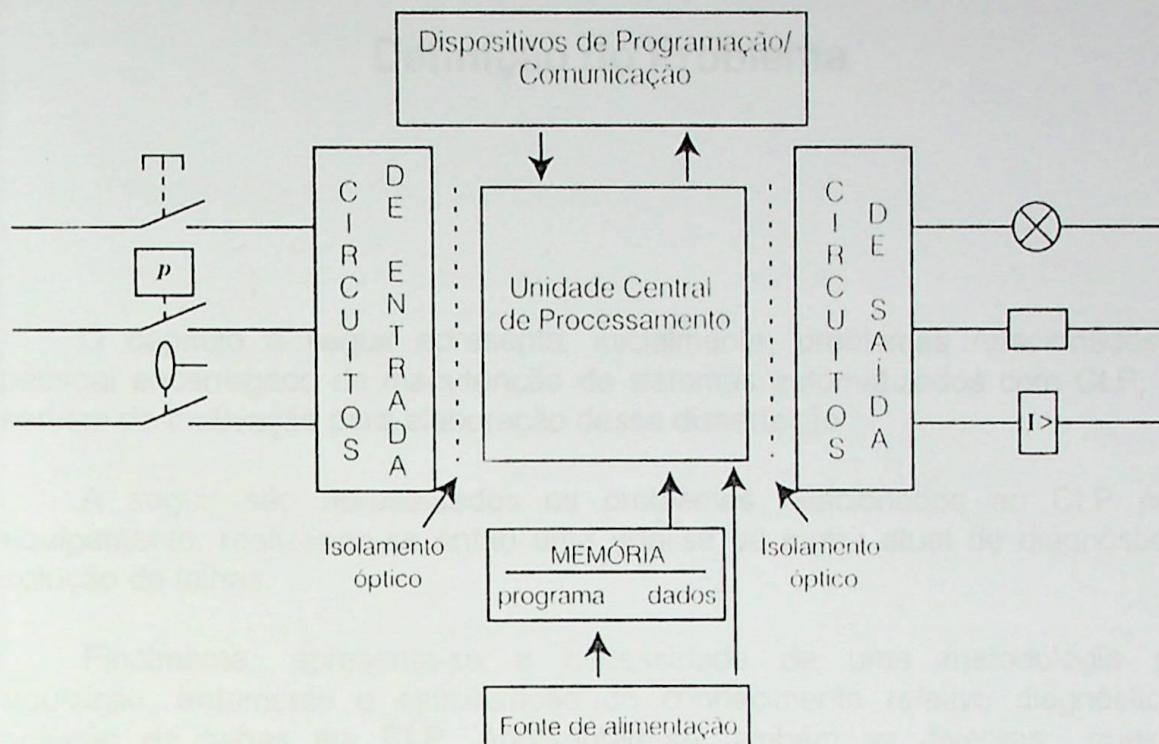


Figura 2.4 – Partes componentes de um CLP.

As falhas no CLP podem ser dos seguintes tipos:

- Falhas de Equipamento: algum componente físico de CLP com defeito em sua eletrônica, como a UCP e Fontes de Alimentação, Portas e acessórios de comunicação (como conversores e meio físico de transmissão da dados), Cartões de Entrada e Saída, Chassis e Ferramenta de Programação. Nesse tipo de falha, inicialmente identifica-se, através de testes, o elemento defeituoso e realiza-se sua substituição por um equivalente em bom estado.
- Falhas de Configuração: algum dos componentes de CLP configurado de forma imprópria para o contexto no qual irá operar. Nesse tipo de falha posições de microchaves ou dados presentes na memória do CLP são alterados.
- Falhas de Programa de Aplicação: falhas na forma como o programa executado pelo CLP foi desenvolvido.

O trabalho objeto dessa dissertação visa o diagnóstico e solução de falhas referentes a Equipamentos e Configuração (restrita a falhas de comunicação de dados). Falhas referentes a programa de aplicação não são consideradas.

Capítulo 3

Definição do Problema

O capítulo a seguir apresenta, inicialmente, problemas relacionados ao pessoal encarregado da manutenção de sistemas automatizados com CLP, que servem de motivação para elaboração dessa dissertação.

A seguir são apresentados os problemas relacionados ao CLP como equipamento, realizando-se então uma análise da forma atual de diagnóstico e solução de falhas.

Finalmente, apresenta-se a necessidade de uma metodologia para aquisição, tratamento e estruturação do conhecimento relativo diagnóstico e solução de falhas em CLP. Apresentam-se também as diversas questões relativas a uma implementação computacional do processo de diagnóstico, solução e aprendizado de falhas em CLP.

3.1 A Motivação

A seguir são apresentados diversos problemas relativos ao pessoal encarregado da manutenção de sistemas automatizados com CLPs, cuja solução motiva o desenvolvimento dessa dissertação.

3.1.1 Pessoal de Manutenção Inexperiente

Pessoal de manutenção inexperiente leva tempo até reagir com rapidez à falhas apresentadas pelos CLPs. Tal fato pode dár-se pelas falhas serem pouco freqüentes no contexto de operação do sistema automatizado ou demorarem a ocorrer pela primeira vez. Assim o pessoal de manutenção além de demorar a ser exposto a uma quantidade razoável de ocorrências de falhas diferentes, tende a esquecer o procedimento de resolução da falha, principalmente quando a ocorrência das falhas é pouco freqüente e mal documentada.

Além disso, muitas industrias não investem no treinamento de seus profissionais de manutenção para conviverem com novas tecnologias e equipamentos de maior complexidade.

O problema torna-se mais grave quando a documentação do equipamento (manuais) não está disponível, é incompleta ou está em idioma estrangeiro, o que une a situação de inexperiência e falta de treinamento adequado, com a ausência de fontes adequadas de consulta, resultando em tentativas de solução das falhas apresentadas pelo CLP por um processo aleatório de aplicação de ações corretivas.

3.1.2 Pessoal de Manutenção Escasso

A necessidade de redução dos custos de produção tem levado as indústrias a manter constante ou até mesmo reduzir o pessoal encarregado de manutenção, mesmo com níveis crescentes de produção.

A escassez de pessoal de manutenção resulta na diminuição da capacidade de resolução de falhas em equipamentos, principalmente quando essas ocorrem de forma simultânea, implicando no aumento do tempo de máquina parada e/ou jornadas de trabalho estendidas, o que por sua vez reduz o faturamento e o lucro.

Como forma de amenizar a redução nos quadros de profissionais ligados a manutenção tem-se investido em tentativas de treinamento do pessoal encarregado da operação das máquinas para realização de tarefas limitadas de manutenção. A idéia é que falhas de baixa e média complexidade (falhas elétricas, substituição de equipamentos defeituosos e eliminação de erros devido a estados de falha gerados por configurações incorretas) poderiam ser resolvidos sem a necessidade de pessoal técnico mais especializado (técnicos ou engenheiros), o que reduziria a ocupação desses na resolução falhas simples, liberando o tempo do pessoal técnico mais especializado para solução das falhas mais complexas.

Na implantação de programas de manutenção de equipamentos por operadores de máquinas, as empresas esbarram no problema de adequar um profissional acostumado a realização de tarefas repetitivas e de baixo grau de complexidade para a realização de tarefas de manutenção que exigem raciocínio lógico e dedutivo. Tal adequação é lenta e nem sempre encontra material humano capaz ou disposto a incrementar suas funções.

3.1.3 Ausência de Metodologia para Resolução de Falhas no CLP

Procedimento comum, no qual não se emprega a seqüência correta de testes para o diagnóstico e solução da falha apresentada pelo CLP, o que leva as falhas a serem solucionadas quase por acaso, com desperdício de tempo.

Mesmo quando há uma procedimento estipulado para resolução das falhas conhecidas, como uma árvore de diagnóstico, esse tende a ser ignorado,

normalmente porque não sofre atualização constante, é confuso ou de consulta difícil.

3.1.4 Mudança de Função ou Perda do Especialista

A mudança de função ou perda do especialista (técnico ou engenheiro) responsável pela manutenção do(s) CLP(s) leva a perda no conhecimento relativo ao diagnóstico e solução de falhas, o que torna a industria dependente do conhecimento acumulado ao longo do tempo por poucos profissionais.

Outro aspecto é que o especialista pode ser profundo conhecedor de um equipamento, porém, quando esse é trocado, sofre algum tipo de atualização tecnológica ou o especialista passa a cuidar de outros equipamentos, novamente tem-se que adquirir o conhecimento necessário ao diagnóstico e solução de falhas.

A perda do especialista torna-se um problema mais grave quando a transmissão do conhecimento acumulado ao longo do tempo não ocorre, ou por falta de documentação dos casos de falhas já ocorridos e das ações realizadas para a sua solução, por "má vontade" do especialista em dividir o conhecimento acumulado ou por dificuldade do especialista em organizar de forma coerente o raciocínio empregado na solução das falhas ocorridas.

3.1.5 Consultas Desnecessárias à Assistência Técnica do Fabricante

Na ocorrência de qualquer falha, procura-se a assistência técnica do fornecedor do CLP. Além do custo elevado, essa nem sempre está disponível para pronto atendimento, resultando em esperas que poderiam ser evitadas. Mesmo com a assistência via telefone, imprecisões na descrição da falha e do contexto na qual essa ocorre dificultam o diagnóstico correto e consequente solução da falha.

3.2 Problemas com o Controlador Lógico Programável

O controlador lógico programável pode apresentar dois tipos de problemas em seu funcionamento:

- Problemas de funcionamento, configuração e instalação: partes integrantes do CLP defeituosas, configuradas de forma errada ou expostas a condições de operação que impeçam seu funcionamento normal.
- Problemas relativos ao programa de aplicação: erros na lógica programada para o controle da máquina ou processo.

Além dos problemas apresentados, existem também os Estados de Operacionais , que não constituem problemas, mas que definem condições operacionais relevantes à definição do estado geral do CLP.

Essa dissertação aborda o diagnóstico e solução dos problemas relativos ao controlador, doravante chamados de falhas, e a determinação e alteração dos estados operacionais presentes.

As falhas possíveis são apresentadas a seguir:

- Falhas do Sistema de Energia: estão associadas a falhas no fornecimento de energia elétrica ao controlador, como falta de energia, níveis de tensão da rede elétrica fora da faixa de valores tolerados, fonte de energia do CLP configurada para receber níveis de tensão incompatíveis com os disponíveis na rede elétrica, fusível da fonte de alimentação queimado, defeito na eletrônica da fonte e defeitos na instalação da fonte. Tal falha é de solução prioritária, pois uma vez que o CLP não recebe e manipula a energia de forma correta, toda a sua operação é comprometida.
- Falhas de UCP: causam a suspensão da execução do programa de aplicação. Podem ser resultado de configurações erradas dos dispositivos que compõe o CLP, operações não permitidas (divisão por zero, por exemplo) executadas pelo programa de aplicação. Tal falha suspende a execução do programa de aplicação, porém não desabilita a operação dos canais de comunicação, o que permite o diagnóstico da falha através do acesso, via ferramenta de programação, ao arquivo de estado do CLP.
- Falhas Graves de UCP: causam a suspensão da execução do programa de aplicação e desabilitam a operação dos canais de comunicação, sendo resolvidas apenas com a volta da UCP as condições iniciais de fábricas (sem programa de aplicação e desconfigurada). As falhas graves de UCP são geradas por defeitos na eletrônica da UCP ou em algum elemento associado a essa (como cartão de cópia de segurança do conteúdo da memória).
- Falhas de Comunicação: estão associadas a incapacidade da UCP trocar dados com outros dispositivos conectados a suas portas de comunicação (ou canais da comunicação). Podem ocorrer devido a falhas de configuração dos parâmetros necessários ao estabelecimento da comunicação entre dispositivos, defeito na eletrônica do canal de comunicação, cabos de comunicação defeituosos, etc... Não resultam em parada na execução do programa de aplicação.

Dentro dos estados operacionais, podemos citar:

- Diagnóstico da condição atual da UCP no que toca a execução do programa de aplicação.

- Presença de entradas ou saídas com estado forçados.
- Bateria de manutenção do programa armazenado na memória ausente ou com carga baixa.

3.2.1 Análise da forma atual do Diagnóstico e Solução de Falhas em CLP

A identificação das falhas e estados operacionais é realizada de forma visual, através da observação da configuração assumida por indicadores luminosos presentes no painel da UCP e da posição da chave que determina o estado de operação da UCP no tocante a execução ou não do programa da aplicação.

Uma vez observada a configuração assumida pelos indicadores luminosos presentes no painel da UCP, essa é comparada com um conjunto definido de configurações presentes na documentação disponibilizada pelo fabricante. Caso a configuração atual dos indicadores luminosos coincida com alguma presente na documentação, a falha ou estado, as causas prováveis e as ações corretivas referentes a cada causa provável são apresentadas. Um exemplo de como um fabricante [13] pode apresentar um estado de falha é mostrado na tabela 3.1.

Estados dos Indicadores Luminosos da UCP	Falha	Causas Prováveis	Ações Corretivas
POWER = ACESO RUN = NÃO IMPORTA FAULT = PISCANDO OU APAGADO BATT = NÃO IMPORTA FORCE = NÃO IMPORTA DH485 = APAGADO RS232 = NÃO IMPORTA	Falha Fatal e falta de comunicação pelo canal DH485.	Sistema de Energia Inadequado.	Conferir tensão da rede. Observar chave seletora de 120V/240V na fonte de alimentação. Observar as demais ações aplicáveis a falhas de sistema de energia.
		Canal de comunicação desativado ("shutdown").	Conferir a configuração do canal de comunicação com a ferramenta de programação. Também ver "retornando o processador as condições iniciais de fábrica".
		Canal de Comunicação danificado.	Trocar processador.

Tabela 3.1 - Um exemplo de como o fabricante pode apresentar um estado de falha.

No exemplo apresentado na tabela 3.1 notam-se algumas características do método de diagnóstico e solução de falhas no CLP apresentadas pelo fabricante:

- Cada configuração de indicadores luminosos da UCP conduz a uma falha.
- Cada falha conduz a um conjunto de causas prováveis.
- Para cada uma das causas prováveis, um conjunto de ações corretivas é proposto.

Esse método apresenta os seguintes problemas:

- As diversas falhas e estados são apresentadas de forma separada , porém esses podem ocorrer simultaneamente o que demanda uma hierarquia na solução das mesmas, fato não verificado nos procedimentos de solução de falha propostos pelo fabricante.
- Dado que a falha é detectada, não há uma hierarquia de causas e ações para a solução da mesma. Muitas vezes as ações corretivas de maior grau de dificuldade de aplicação são propostas antes das ações de menor grau de dificuldade. Entende-se como ações corretivas de maior grau de dificuldade aquelas que exigem desmontagem e troca de equipamentos ou reconfiguração de parâmetros. Entende-se como ações corretivas de menor grau de dificuldade aquelas que requerem inspeção visual, verificação de conexões e medição de níveis de grandezas elétricas.
- As ações corretivas não levam em conta os recursos para manutenção disponíveis no local onde o CLP está instalado, tais como: peças de reposição, ferramenta de programação, cópia atualizada do programa da aplicação e etc...
- As ações corretivas propostas carecem de detalhamento e muitas vezes omitem procedimentos.
- A documentação que contém os procedimentos de diagnóstico e solução de falhas nem sempre está disponível ou é de livre acesso, estando normalmente em língua estrangeira.
- Não há um mecanismo de aprendizado e armazenamento dos casos de falhas já ocorridos, sendo que fica-se dependente da experiência do profissional de manutenção.

3.3 Problemas com a Aquisição, Tratamento e Estruturação do Conhecimento a respeito do Diagnóstico e Solução de Falhas em CLP

O conhecimento disponível sobre o diagnóstico e solução de falhas em CLP nem sempre encontra-se disponível de forma clara e sistemática em suas fontes. Além disso, esse pode conter ambigüidades, repetições desnecessárias de procedimentos, omissão de dados relevantes e imprecisões nos testes e ações corretivas propostas.

Assim, necessita-se de um método que permita, de forma sistemática, adquirir, tratar e estruturar o conhecimento que será empregados para a implementação computacional do processo de diagnóstico de solução de falhas em CLP.

3.4 Problemas de Implementação Computacional do Processo de Diagnóstico, Solução e Aprendizado de Falhas em CLP

Qualquer sistema computacional cujo objetivo seja servir como ferramenta de auxílio ao diagnóstico e solução de falhas em CLP deve implementar uma descrição proposta para o processo de diagnóstico e solução de falhas no CLP.

O passos necessários à implementação computacional são:

- Propor uma descrição do processo de diagnóstico e solução de falhas em CLP, de forma a poder-se definir o funcionamento da implementação computacional a ser desenvolvida.
- Considerar os fatores relevantes para tomadas de decisão, como recursos disponíveis e contextos nos quais a falha ocorre.
- Propor uma forma de determinar a ordem de escolha das ações corretivas aplicadas.
- Propor uma forma de aprendizado de soluções de sucesso para aplicação futura.
- Escolher as técnicas de inteligência artificial que implementem o processo descrito de diagnóstico, solução e aprendizado de falha.

O Capítulo 4 decreve a solução proposta para os problemas apresentados.

Capítulo 4

A Solução

Este capítulo apresenta a solução adotada para os problemas de manutenção de sistemas automatizados com CLP apresentados no Capítulo 3.

Inicialmente, propõe-se uma descrição para o processo de diagnóstico e solução de falhas em CLP, assim como a forma como ocorre o aprendizado das soluções.

A seguir, propõe-se uma metodologia para a aquisição, tratamento e estruturação do conhecimento a ser utilizado na solução de falhas e detecção de estados operacionais no CLP.

Finalmente, propõe-se uma ferramenta computacional baseada em regras e casos que implementa a descrição proposta, aplicando a metodologia proposta para a aquisição, tratamento e estruturação do conhecimento, de forma a diagnosticar, solucionar e possibilitar o aprendizado de falhas em CLP.

4.1 Uma Proposta de Descrição do Processo Diagnóstico, Solução e Aprendizado de Falhas em CLP

O primeiro passo para a solução dos problemas apresentados no capítulo 3 é a proposição de uma descrição para processo diagnóstico, solução e aprendizado de falhas no CLP. Tal descrição será a base do desenvolvimento da ferramenta computacional apresentada nessa dissertação.

Para a proposição de uma descrição do processo de diagnóstico e solução de falhas em CLP, considera-se a existência de um técnico ou engenheiro de manutenção inexperiente encarregado da manutenção de sistemas automatizados com CLP. Considera-se que inicialmente esse profissional possui um conhecimento generalizado a respeito das diversas falhas que podem ocorrer, suas causas e ações corretivas, proveniente das diversas fontes de consulta disponíveis, como manuais, relatórios de manutenção e treinamentos focados na manutenção do CLP utilizado. Ou seja, considera-se que o conhecimento existe, porém a inexperiência do mantenedor leva a sua aplicação de forma sistemática até que uma das ações corretivas logre resolver a situação de falha. Uma vez

resolvida a falha, sua causa é inferida e o processo de diagnóstico e solução completo é memorizado. Na repetição da ocorrência da mesma falha, o considera-se que o mantenedor tende inicialmente a resolver o problema usando as mesmas ações corretivas que resolvem a falha na sua última ocorrência. Caso as ações corretivas logrem sucesso, ocorre um aumento heurístico da certeza (ou reforço do conhecimento) de que sempre que tal falha ocorre, tal ação corretiva a resolve e a causa é conhecida. Quando as ações corretivas adotadas não logram sucesso na resolução da falha, o mantenedor retorna ao conhecimento obtido nos manuais e treinamentos, voltando a aplicar ações corretivas de forma sistemática até aplicar o conjunto de ações que resolva a falha atual.

Conforme descrito no Capítulo 3, o problema de solução de falhas em CLPs apresenta a característica peculiar de que para uma dada falha, diversas causas e ações corretivas são possíveis. A princípio apenas as ações corretivas são conhecidas, sendo a causa do problema inferida a partir do sucesso do conjunto de ações aplicadas.

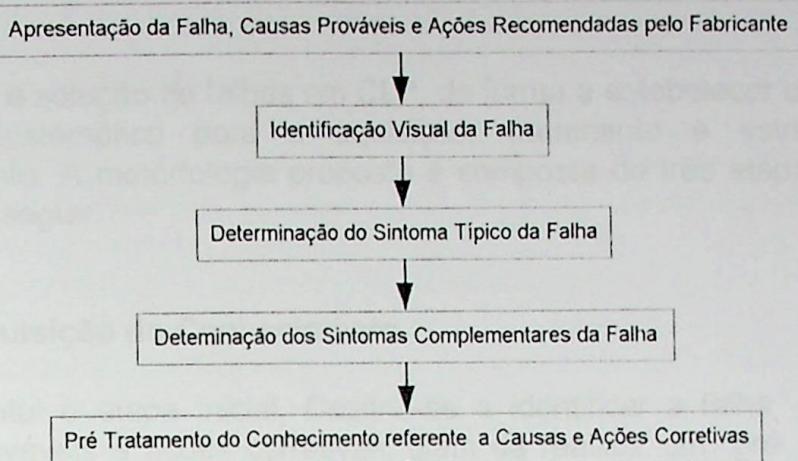
Assim, considera-se para efeito da implementação computacional, que o mantenedor inexperiente, porém treinado, conhece um conjunto de falhas, ações corretivas e causas, porém carece da heurística que leve-o a tomar determinada ação corretiva em detrimento de outra, sendo essas aplicadas inicialmente de forma sistemática, normalmente das ações de menor grau de complexidade para as de maior grau de complexidade, do menor uso de recursos para o maior uso de recursos. Ao longo do tempo, com a ocorrência de falhas, casos vão sendo memorizados pelo mantenedor, fazendo com que a busca de ações corretivas para as falhas deixe de ser uma ação sistemática e passe a basear-se na experiência acumulada ao longo do tempo. Infere-se então que após algum tempo, o mantenedor terá vivenciado praticamente todas as situações de falha, tendo constituído assim uma coleção de casos que serão analisados, e levarão a tomada de ações corretivas, normalmente baseadas no número de vezes que tal ação corrigiu a falha em ocasiões anteriores e na complexidade da aplicação das ações.

Além disso, considera-se que o mantenedor é limitado pelo recursos disponíveis para levar a cabo as ações corretivas possíveis. Ou seja, dado que existem diversas ações corretivas que podem ser aplicadas, serão aplicadas as que estão de acordo os recursos disponíveis. O mantenedor sabe que não é possível aplicar uma ação corretiva que exija o teste da UCP instalada no CLP atual em outro chassis (um chassis de teste) se esse simplesmente não está disponível no contexto da planta onde encontra-se instalado o CLP.

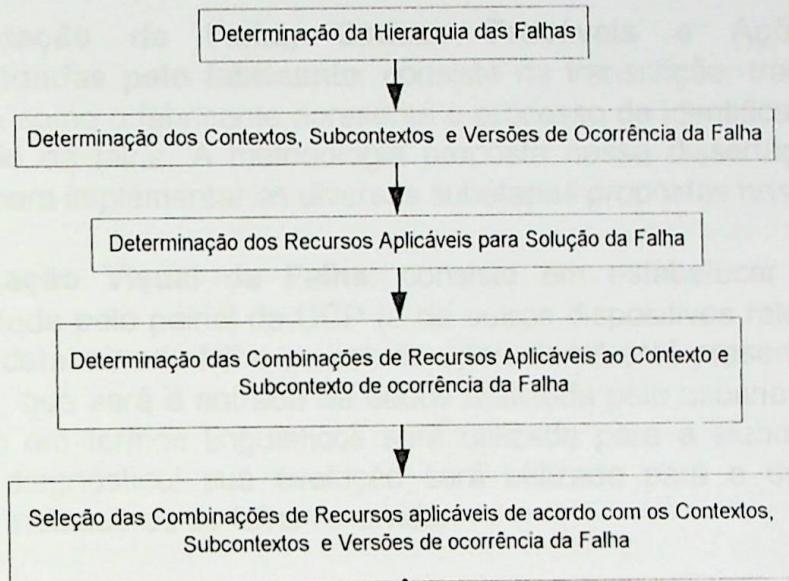
4.2 Uma Metodologia para Aquisição, Tratamento e Estruturação do Conhecimento referente ao Diagnóstico e Solução de Falhas em CLP

A metodologia proposta, figura 4.1, visa adequar a uma implementação computacional o conhecimento disponibilizado pelo fabricante a respeito do

Aquisição do Conhecimento



Tratamento do Conhecimento



Estruturação do Conhecimento

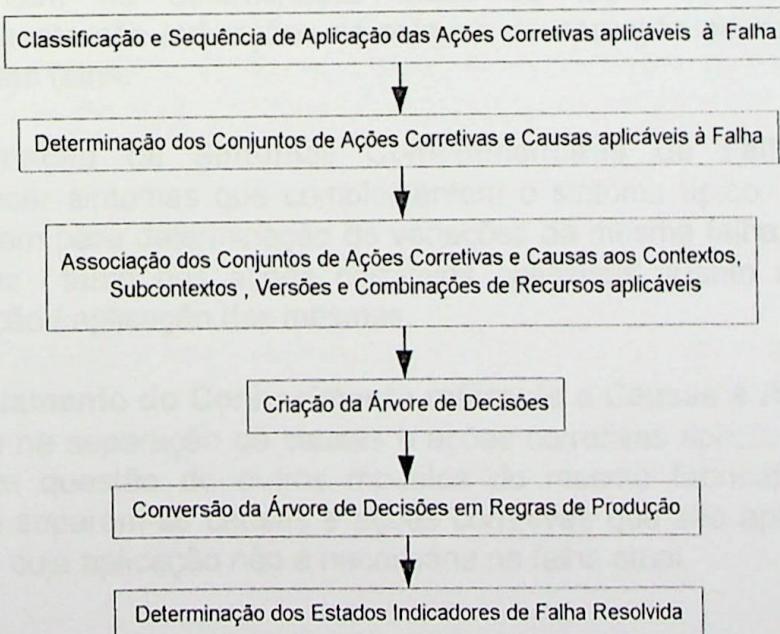


Figura 4.1 Passos para implementação da metodologia proposta.

diagnóstico e solução de falhas em CLP, de forma a estabelecer um procedimento formal e sistemático para a aquisição, tratamento e estruturação desse conhecimento. A metodologia proposta é composta de três etapas, sendo essas descritas a seguir.

4.2.1 Aquisição do Conhecimento

Constitui a etapa inicial. Destina-se a identificar a falha, os sintomas, as causas prováveis e ações corretivas, além de realizar um pré - tratamento das causas e ações corretivas recomendadas. O conhecimento básico está normalmente contido em manuais de manutenção, na forma de texto ou árvores de diagnóstico. Esta etapa é dividida em diversas subetapas, descritas a seguir.

- **Apresentação da Falha, Causas Prováveis e Ações Corretivas recomendadas pelo fabricante:** consiste na transcrição, tradução e análise da forma como o fabricante apresenta o processo de identificação, diagnóstico e solução da falha. A metodologia proposta nessa dissertação parte dessa análise para implementar as diversas subetapas propostas nos próximos itens.
- **Identificação Visual da Falha:** consiste em estabelecer a forma visual apresentada pelo painel da UCP (e de outros dispositivos relevantes) do CLP quando determinada falha ou estado operacional está presente. É o **sintoma** da falha, que será a entrada de dados realizada pelo usuário do sistema; sua tradução em termos lingüísticos será utilizada para a elaboração de regras para o diagnóstico; sua evolução será utilizada para a determinação dos estados indicadores de falha resolvida.
- **Determinação do Sintoma Típico da Falha:** consiste em estabelecer qual é o sintoma que realmente caracteriza a falha e a diferencia de outras falhas que podem ocorrer simultaneamente, de forma a separar os sintomas que nada acrescentam na determinação exata da falha ou de sintomas que simplesmente são indicações de estados de operação normal da UCP e não configuram falha.
- **Determinação de Sintomas Complementares da Falha:** consiste em estabelecer sintomas que complementam o sintoma típico. Quando existem, contribuem para determinação de variações da mesma falha, que determinam variações tanto nas ações corretivas aplicáveis quanto na seqüência de proposição / aplicação das mesmas.
- **Pré Tratamento do Conhecimento referente a Causas e Ações Corretivas:** consiste na separação de causas e ações corretivas aplicáveis ao modelo da UCP em questão de outros modelos do mesmo fabricante. Nessa etapa também separam-se causas e ações corretivas que são aplicadas em outras falhas e cuja aplicação não é necessária na falha atual.

4.2.2 Tratamento do Conhecimento

Constitui a etapa intermediária. Destina-se a determinação dos contextos, subcontextos e versões de ocorrência da falha, determinação dos recursos utilizáveis, suas combinações e a adequação das combinações de recursos utilizáveis aos contextos, subcontextos e versões de ocorrência da falha.

- **Determinação da Hierarquia do Diagnóstico das Falhas:** consiste em determinar quais falhas são mais e menos graves, de forma a estabelecer uma prioridade no diagnóstico (e solução) de falhas que podem ocorrer simultaneamente e filtrar sintomas de falhas menos graves cuja ocorrência é um efeito colateral da ocorrência de falhas mais graves. O processo de diagnóstico e solução de falhas sempre parte das mais graves para as menos graves.
- **Determinação dos Contextos, Subcontextos e Versões de Ocorrência da Falha:** as situações nas quais a falha pode ocorrer, chamadas aqui de contextos e subcontextos, são importantes por serem utilizadas na definição das ações corretivas aplicáveis. Exemplos dessas situações podem ser: presença ou ausência de um determinado item acessório instalado no CLP, chave que determina o modo de operação da UCP em determinada posição ou uma tentativa de trocar dados com o CLP por uma determinada porta de comunicação em detrimento de outra. Os contextos e subcontextos são obtidos pela análise do conhecimento disponível a respeito da falha em questão. As versões são variações de um mesmo subcontexto que normalmente alteram as ações corretivas e sua ordem de aplicação.
- **Determinação dos Recursos Aplicáveis para Solução da Falha:** nesse item são listados os equipamentos utilizáveis para solução das falhas. Além dos contextos, subcontextos e versões, os recursos disponíveis definem as ações corretivas que podem ser aplicadas. Os recursos são obtidos pela análise do conhecimento disponível a respeito da falha em questão, sendo completados pela experiência do especialista (pois muitas vezes são omitidos pelo fabricante). Alguma vez os recursos são agrupados, pois são aplicáveis apenas se diversos recursos estão presentes ao mesmo tempo.
- **Determinação das Combinações de Recursos Aplicáveis aos Contexto e Subcontexto de Ocorrência da Falha:** as combinações dos diversos recursos utilizáveis são realizadas utilizando uma tabela verdade, onde são explicitadas de forma sistemática todas as combinações possíveis de recursos (linhas da tabela verdade) e os recursos propriamente ditos (colunas da tabela verdade). A presença de um determinado recurso é representada por um SIM (equivalente a um "1" lógico) na tabela verdade e sua ausência por um NÃO (equivalente ao "0" lógico). É criada uma tabela verdade para cada contexto e subcontexto (caso esse exista). Versões de um mesmo subcontexto não alteram a tabela verdade.

- **Seleção das Combinações de Recursos Aplicáveis de acordo com os Contextos, Subcontextos e Versões de Ocorrência da Falha:** uma vez definidas, de forma sistemática, as combinações de recursos aplicáveis a cada contexto, subcontexto e versão nos quais a falha apresenta-se, resta separar quais combinações de recursos podem ser utilizadas para a geração das ações corretiva à serem aplicadas. Assim, são separadas as combinações mínimas de recursos aplicáveis das combinações que não permitem a tomada de algum tipo de ação corretiva. Em alguns casos, mesmo com nenhum recurso disponível, alguma ações corretivas podem ser tomadas por exigirem, por exemplo, apenas testes visuais.

4.2.3 Estruturação do Conhecimento

Constitui a etapa final. Destina-se a classificar as ações corretivas em tipos de ações corretivas e sua seqüência de aplicação, estabelecer (para cada contexto, subcontexto e versão) os conjuntos de ações corretivas a serem tomadas e a causa a ser inferida a partir das ações que lograram sucesso, determinar os diversos conjuntos de ações - causa (chamadas aqui de Base de Ações e Causa, BAC) aplicáveis a cada contexto, subcontexto, versão e combinação de recursos, estabelecer a estrutura da árvore de decisões para escolha das BAC, transformar a árvore de decisões num conjunto de regras de produção cuja avaliação implementa o processo de busca das BAC e, finalmente, determinar os estados assumidos pela UCP que indicam falha resolvida.

- **Classificação e Seqüência de Aplicação das Ações Corretivas aplicáveis à Falha:** a classificação dos conjuntos de ações corretivas dá-se pela complexidade envolvida na sua aplicação. Conjuntos de ações que exigem apenas inspeção visual, medições de grandezas elétricas e correção da falha sem necessidade de troca de itens componentes do CLP são considerados nessa dissertação como de menor complexidade. Conjuntos de ações que requerem substituição de itens componentes do CLP são considerados de complexidade intermediária por requererem algum tipo de reconfiguração do CLP ou teste do item suspeito em outra estrutura. Conjuntos de ações que exigem a conexão da ferramenta de programação e testes / reconfigurações através de alteração de parâmetros de programa de aplicação são consideradas de alta complexidade. Cabe notar que a classificação das ações em baixa, intermediária e alta leva em conta o público alvo ao qual a ferramenta de apoio a manutenção de CLP aplica-se: operadores de máquina, com baixo grau de escolaridade e conhecimento superficial do funcionamento do CLP. A seqüência de aplicação da ações inicia-se nos conjuntos de ações de menor complexidade, passa pelas de complexidade intermediária e termina nas de maior complexidade. Dependendo da falha, os conjuntos de ações não contemplam, necessariamente, os três níveis de complexidade.

- **Determinação dos Conjuntos de Ações Corretivas e Causas aplicáveis à Falha:** cada conjunto de ações contém ações corretivas e testes a serem realizados, apresentados de forma seqüencial e detalhada. Aqui, além das ações corretivas e testes determinados pelo fabricante, podem ser inclusas ações e testes determinados pela experiência do especialista, de forma à complementar os conjuntos. Os conjuntos de ações são criados de acordo com os recursos disponíveis: quanto mais recursos, maior o número de conjuntos aplicáveis, quanto menos recursos, menor o número de conjuntos de ações corretivas e testes aplicáveis, sendo os conjuntos adequados aos contextos, subcontextos e versões possíveis. Cada conjunto de ações é formado considerando uma causa provável para a falha, sendo completo, ou seja, considera-se que possui todas ações e testes necessários a resolução da falha, sem a necessidade de que ações propostas em outros conjuntos já tenham sido aplicadas. Conforme apresentado anteriormente, a solução de falhas no CLP tem a particularidade de não sabermos, a princípio, a causa da falha, assim, cria-se um conjunto de ações corretivas para cada causa provável e caso a aplicação dessas logre a solução da falha, a causa é deduzida, caso não, parte-se para aplicação de ações contidas em outros conjuntos, repetindo-se o processo até que um conjunto de ações resolva a falha ou até todos os conjuntos disponíveis para o contexto/subcontexto/versão/recursos disponíveis terem sido aplicados.
- **Associação dos Conjuntos de Ações Corretivas e Causas aos Contextos, Subcontextos, Versões e Combinações de Recursos Aplicáveis à Falha:** nesse item, para cada contexto / subcontexto / versão são associadas as diversas combinações de recursos geradores de ações corretivas e os conjuntos de ações aplicáveis, já explicitando a seqüência de aplicação dos conjuntos de ações. Para cada associação contexto / subcontexto / versão / combinação de recursos aplicáveis / conjuntos de ações, é definida uma base de ações e causas (BAC), que será escolhida como resultado da avaliação da árvore de decisões e das regras que a implementam, além de serem usadas na futura implementação computacional.
- **Criação da Árvore de Decisões:** a árvore de decisões sintetiza, de forma gráfica (fluxograma), o conjunto sintomas / contexto / subcontexto / versão / combinação de recursos disponíveis a serem avaliados e a consequente escolha da base de ações e causas (BAC) aplicável a solução da falha.
- **Conversão da Árvore de Decisões em Regras de Produção:** uma vez que a árvore de decisões é criada, essa deve ser traduzida para um conjunto de regras de produção que a implementam. As regras de produção são utilizadas como base para a implementação computacional que será desenvolvida, pois detém todo o conhecimento a respeito das falhas possíveis, seus sintomas, contextos, subcontextos, versões, combinação de recursos e escolha dos conjuntos de ações corretivas - causas.

- **Determinação dos Estados Indicadores de Falha Resolvida:** uma vez que as ações corretivas tenham sido apresentadas e aplicadas, é necessário apresentar algum tipo de indicação que permita ao usuário do sistema perceber se as ações aplicadas lograram ou não sucesso na solução da falha, assim, torna-se necessário determinar os estados para os quais a UCP (ou outros componentes) do CLP pode evoluir. Tal evolução pode dar-se para um estado de ausência de falha (pois apenas uma falha existia e essa foi resolvida) ou para uma outra falha, diferente da anterior, que estava encoberta por alguma falha mais grave.

No Capítulo 5 é apresentado um exemplo completo da aplicação da metodologia proposta para solução de uma falha de comunicação de um CLP comercial.

4.3 Proposta de um Sistema Baseado em Regras e Casos para Diagnóstico Solução e Aprendizado de Falhas em CLP

O sistema proposto nesse trabalho consiste num sistema de diagnóstico de falhas baseado em regras fortemente integrado a um sistema baseado em casos para o aprendizado das soluções aplicadas às falhas ocorridas. O objetivo do sistema é implementar a descrição proposta para o processo de diagnóstico, solução e aprendizado de falhas em sistemas automatizados por CLP, utilizando a metodologia proposta para aquisição, tratamento e estruturação do conhecimento para o diagnóstico e solução de falhas em CLP.

Sistemas inteligentes baseados puramente em regras tem larga aplicação em sistemas de diagnóstico, principalmente por implementarem de forma direta a estrutura de uma árvore de decisões, porém carecem da característica de aprendizado, além de seguirem sempre o mesmo procedimento de solução, passo por passo, sempre que os mesmos fatos são apresentados. Por outro lado, sistemas inteligentes baseados em casos podem apresentar a característica de aprendizado, porém tem como pré-requisito a existência de um conjunto de casos já ocorridos e devidamente registrados, fato que pode não verificar-se no contexto de operação do sistema. O sistema proposto busca extrair as vantagens tanto dos sistemas baseados em regras quanto dos baseados em casos, minimizando suas desvantagens.

4.3.1 Descrição dos Componentes do Sistema Proposto

A seguir são descritos os componentes do sistema baseado em regras e casos proposto.

4.3.1.1 O Sub-sistema Baseado em Regras

O sub-sistema baseado em regras contém um conjunto de regras extraídas da árvore de decisões. Essas são responsáveis pela determinação da falha a partir dos sintomas apresentados, determinação do contexto/ subcontexto/versão (caso existam) e determinação da base ações – causa (BAC) ou base de casos ocorridos (BCO) de acordo com os recursos disponíveis e do conhecimento acumulado.

As regras contém o conhecimento geral, ou seja, sobre todas as falhas e ações corretivas (contidas nas BAC) que podem ocorrer com o CLP em questão, implementando assim mantenedor inexperiente, porém treinado, mencionado na descrição proposta.

O sistema dispõe de quatro tipos de regras, classificadas de acordo com a função que desempenham, sendo essas apresentadas a seguir.

4.3.1.1.1 Regras Tipo 1, ou Regras para Diagnóstico de Falhas

Comparam os dados fornecidos pelo usuário com as situações de falha possíveis. Cada regra é disparada quando os dados fornecidos pelo usuário coincidem com estados de falha, concluindo assim a falha corrente. Tais regras tem o seguinte formato:

SE (Sintoma_1 = A)
E (Sintoma_2 = B)
E (Sintoma_n = Z)
ENTÃO
 (Falha = K)

No caso do sistema proposto o sistema considera que todas as regras processam dados cujos valores possíveis são apenas verdadeiro ou falsos.

4.3.1.1.2 Regras Tipo 2, ou Regras para Determinação de Contexto e Subcontexto de operação do CLP

Comparam os dados fornecidos ao sistema sobre o contexto atual com os contextos possíveis. A regra na qual os dados disponíveis coincidem com um contexto é disparada, concluindo o contexto corrente . Existem duas fontes de dados para determinação do contexto:

- Busca numa Base de Dados de Configuração (BDC) do CLP: essa contém dados referentes a configuração do CLP, previamente introduzidos na fase de configuração do sistema.

- Pergunta ao usuário: em alguns casos, o sistema necessita de informações sobre as condições de operação do CLP durante a ocorrência da falha, que só podem ser fornecidas quando o sistema é consultado. Procura-se evitar ao máximo essa fonte de dados, porém em alguns casos (falhas de comunicação, por exemplo) isso é necessário.

Tais regras tem seguinte formato:

```

SE (Falha = K)
E (Dado sobre Contexto/Subcontexto_1 = AA)
E (Dado sobre Contexto/Subcontexto_2 = BB)
E (Dado sobre Contexto/Subcontexto_n = CC)
ENTÃO
(Conteúdo/Subcontexto = DD)

```

4.3.1.1.3 Regras Tipo 3, ou Regras para Determinação da Versão de ocorrência da falha no CLP

Uma vez determinado o contexto e o subcontexto de ocorrência da falha, a versão de ocorrência da falha deve ser determinada (caso essa exista).

Tais regras tem o formato:

```

SE (Conteúdo/Subcontexto = DD)
E (Dados sobre Versão = EE)
ENTÃO
(Versão = FF)

```

4.3.1.1.4 Regras Tipo 4, ou Regras para Determinação dos Conjuntos de Ações Corretivas e Causas

Comparam os dados fornecidos pelo sistema, provenientes da BDC, sobre os recursos disponíveis para manutenção, com os recursos necessários à aplicação de determinados conjuntos de ações. A regra na qual os dados disponíveis sobre os recursos necessários coincide com os recursos disponíveis é disparada, concluindo a BAC (ou BCO) aplicável à solução da falha.

Tais regras tem seguinte formato:

```

SE (Conteúdo/Subcontexto = DD)
E (Versão = FF)
E (Recurso_1 = AAA)
E (Recurso_n = ZZZ)
ENTÃO
(BAC = KKK) [ ou BCO = KKK caso não esteja vazia)]

```

O fato das regras Tipo 2, 3 e 4 conterem em si premissas que são conclusões de outras regras cria um encadeamento nas decisões tomadas pelo sistema, o que limita o espaço de busca por soluções, que passa a ser formado apenas pelas regras Tipo 2, 3 e 4 que contém determinada premissa já avaliada como verdadeira (pois é a conclusão de uma regra avaliada como verdadeira), implementando assim uma busca em profundidade no espaço de solução.

O espaço de solução do sistema é formado por todas as regras Tipo 1,2,3 e 4, BACs e BCOs disponíveis para o diagnóstico e solução das falhas.

A figura 4.2 apresenta o fluxo de avaliação das regras para determinação de uma falha, seu contexto/subcontexto/versão e a BAC ou BCO adequada.

É importante notar que o sistema considera que os dados fornecidos para avaliação das regras dos diversos tipos são verdadeiros. No tocante a coerência, existem regras para avaliação da coerência dos dados fornecidos ao sistema apenas nas seguintes fases:

- Configuração: nessa existem dados cuja presença exclui a presença de outros dados, por exemplo: CLP estar usando seu único canal de comunicação, configurável para comunicação em rede, para realizar comunicação ponto a ponto implica na não conexão do CLP em rede.
- Entradas de dados referentes a Sintomas: existem sintomas cuja presença exclui a presença de outros sintomas, por exemplo: se o CLP está desenergizado, é incoerente o sistema receber uma entrada de dados indicando a execução do programa de aplicação.

No tocante a verdade contida nos dados, nada pode-se assegurar, pois pode-se fornecer um conjunto de dados falsos, porém coerentes, que levem o sistema a diagnosticar falhas inexistentes. Cabe notar que o especialista humano, que o sistema tenta “imitar”, pode tomar decisões erradas quando analisa dados falsos.

4.3.2 O Método de Inferência para Processamento das Regras

O encadeamento para frente é o método de inferência utilizado para a avaliação e disparo das regras, pois nesse método as regras são aceitas baseadas nos fatos atuais apresentados ao sistema, independente de objetivos predeterminados, ou seja, dado que os fatos atuais coincidem com as premissas da regra sob análise, essa é aceita, obtendo-se assim uma conclusão.

As regras para diagnóstico de falhas são avaliadas e disparadas de forma seqüencial, sendo que, ao final, todas as falhas presentes foram diagnosticadas, podendo-se partir para a escolha da falha que será resolvida primeiro. Há, porém, uma prioridade na avaliação e disparo das regras. Essa é necessária, pois existem

fallas, como as no sub-sistema de energia do CLP, que quando ocorrem fazem com que o CLP suspenda a operação de outros sub-sistemas, sem que esses apresentem realmente alguma falha real em seu funcionamento. Regras de maior prioridade são avaliadas primeiro.

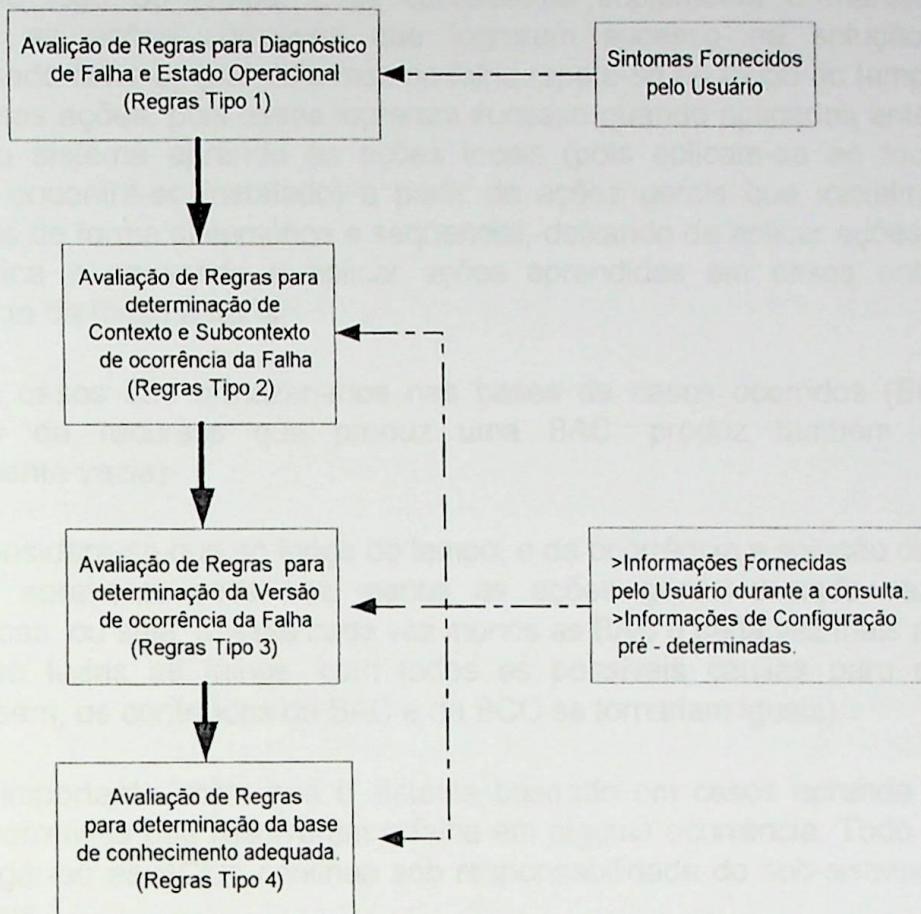


Figura 4.2 - Fluxo de avaliação das regras para determinação de uma falha, seu contexto/subcontexto/versão e escolha da base de conhecimento adequada.

A prioridade na avaliação das regras é implementada por sua posição na seqüência das regras. O resultado da avaliação das regras mais prioritárias passa a ser uma premissa para outras regras que diagnosticam outras falhas, falhas essas que só podem ocorrer realmente se uma determinada falha de maior prioridade não ocorre. O exemplo desse tipo de regra (uma variante da regra Tipo 1) é apresentada a seguir:

```

SE NÃO(Falha = K)
E      (Sintoma_1 = A)
E      (Sintoma_2 = B)
E      (Sintoma_n = Z)
ENTÃO
      (Falha = Y)
  
```

4.4 O Sub-sistema Baseado em Casos

A função do sub-sistema baseado em casos é prover ao sistema proposto uma forma de aprendizado e graduação (na forma de pesos) ao conhecimento obtido ao longo do tempo. Esse sub-sistema implementa o mantenedor que aprende as ações corretivas que lograram sucesso na solução de uma determinada falha e, quando a mesma falha repete-se ao longo do tempo, reutiliza as mesmas ações, pois essas lograram sucesso quando aplicadas anteriormente. Assim, o sistema aprende as ações locais (pois aplicam-se ao local onde o sistema encontra-se instalado) a partir de ações gerais que inicialmente eram aplicadas de forma sistemática e seqüencial, deixando de aplicar ações em ordem sistemática e passando a aplicar ações aprendidas em casos anteriores de ocorrência da mesma falha.

Os casos são armazenados nas bases de casos ocorridos (BCO). Cada conjunto de recursos que produz uma BAC, produz também uma BCO (inicialmente vazia).

Considera-se que ao longo do tempo, e da ocorrência e solução das falhas, o sistema apresente cada vez menos as ações gerais e cada vez mais as específicas, ou seja, acesse cada vez menos as BAC e cada vez mais as BCO (no limite, se todas as falhas, com todas as possíveis causas para cada falha, ocorressem, os conteúdos da BAC e da BCO se tornariam iguais).

É importante notar que o sistema baseado em casos aprende apenas as ações corretivas que resolveram a falha em alguma ocorrência. Todo o processo até chegar-se as BCOs continua sob responsabilidade do sub-sistema baseado em regras.

Um caso aprendido contém os seguintes dados:

- Conjunto ações corretivas que resolveu a falha e a causa da falha: é mostrado ao usuário do sistema como uma possível solução da falha atual.
- Um número de ocorrências (peso) associado a cada conjunto ações – causa: inicialmente todos os conjuntos de ações - causa aprendidos tem o mesmo número de ocorrências (no mínimo uma). A medida que a falha repete-se e o conjunto ações - causa, apresentado como possível solução, logra resolver novamente o problema, esse número de ocorrências (peso) é elevado em uma unidade, indicando que esse conjunto tem tido mais sucesso que os demais já aprendidos. Quando os diversos conjuntos aprendidos tem pesos iguais, o de menor complexidade é apresentado.
- A complexidade relativa à aplicação do conjunto de ações: serve como critério de desempate entre conjuntos ações – causa com pesos iguais. Depende do tipo de ação e dos recursos utilizados, sendo que os casos de

menor complexidade são mostrados primeiro. A ordem de apresentação dos conjuntos de ações – causa presentes na BACs, que originam as BCOs, já contém, de forma implícita (na posição do conjunto de ações dentro dos conjuntos disponíveis), a complexidade das ações.

Um caso presente numa BCO tem o formato apresentado na tabela 4.1.

Complexidade	Nº Ocorrências (Peso)	Conjunto de Ações	Causa
--------------	-----------------------	-------------------	-------

Tabela 4.1 – Formato de um caso presente numa BCO.

4.4.1 O Algoritmo de Seleção e Ordenação dos Casos Armazenados

Uma vez diagnosticada a falha, determinado seu contexto / subcontexto / versão de ocorrência e avaliados os recursos disponíveis para sua solução, é determinada uma base conhecimento para solução da falha. Essa pode ser uma BAC ou uma BCO. O algoritmo de seleção e ordenação de casos armazenados é apresentado no fluxograma da figura 4.3, sendo sua implementação em pseudocódigo apresentada no Anexo 1.

Inicialmente, testa-se a BCO adequada, caso essa tenha casos armazenados (não esteja vazia) essa é ativada, servindo como fornecedora das ações corretivas e causas da falha. Caso não tenha casos armazenados (esteja vazia), a BCO é desativada e a BAC adequada é ativada, servindo como fornecedora das ações corretivas e causas da falha.

Uma vez que uma BCO tenha sido escolhida, existe a necessidade da ordenação dos casos que serão apresentados como prováveis soluções para a falha. Tal ordenação é realizada seguindo os seguintes critérios:

- Casos com maior número de ocorrências (peso) tem precedência sobre casos com menor número de ocorrência.
- Casos com menor complexidade tem precedência sobre casos de maior complexidade.

Cabe notar que vários casos armazenados podem ter o mesmo número de ocorrências, porém cada caso tem complexidade única e pré-definida na BAC que o originou.

A BCO ativa armazena os casos ordenados por complexidade (como as BACs), porém, em tempo de execução (durante a consulta ao sistema), a BCO deve ser ordenada de forma representar o conhecimento adquirido, que pode alterar-se ao final de cada consulta ao sistema. Para tal é criada uma base de casos temporária ou Base de Casos Ocorridos Ordenada (BCOO). Essa, no inicio da consulta ao sistema, é preenchida, com os casos presentes na BCO, seguindo

Operações de Inicialização
 Busca do Caso com maior número de ocorrências na BCO p/ posterior ordenação da BCOO por "Nº de Ocorrências"
 Cópia dos casos da BCO para a BCOO, ordenados na BCOO pelo "Nº de Ocorrências"

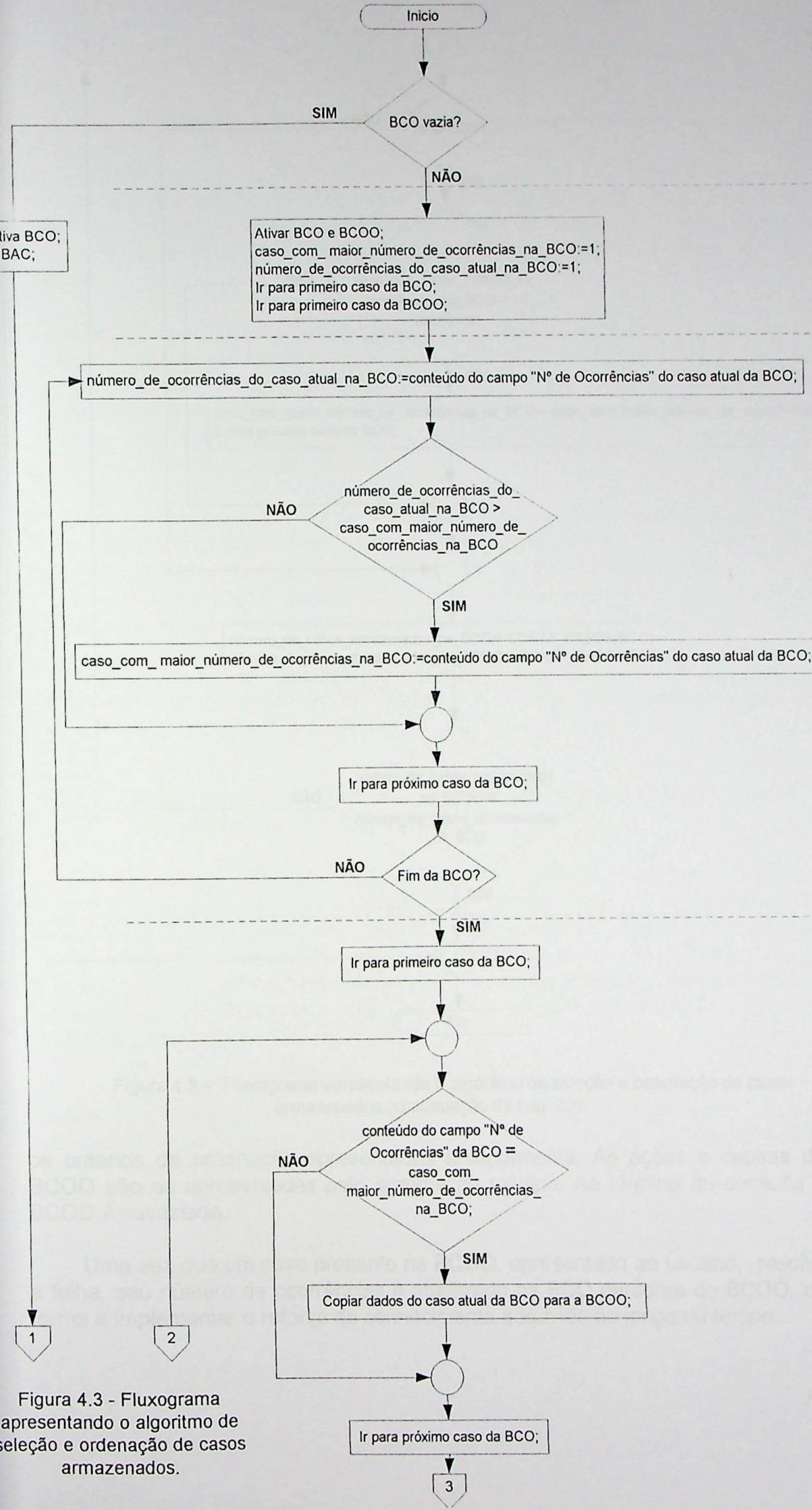


Figura 4.3 - Fluxograma apresentando o algoritmo de seleção e ordenação de casos armazenados.

Continua na página 40

Cópia dos casos da BCO para a BCOO,
ordenados na BCOO pelo "Nº de
Ocorrências".

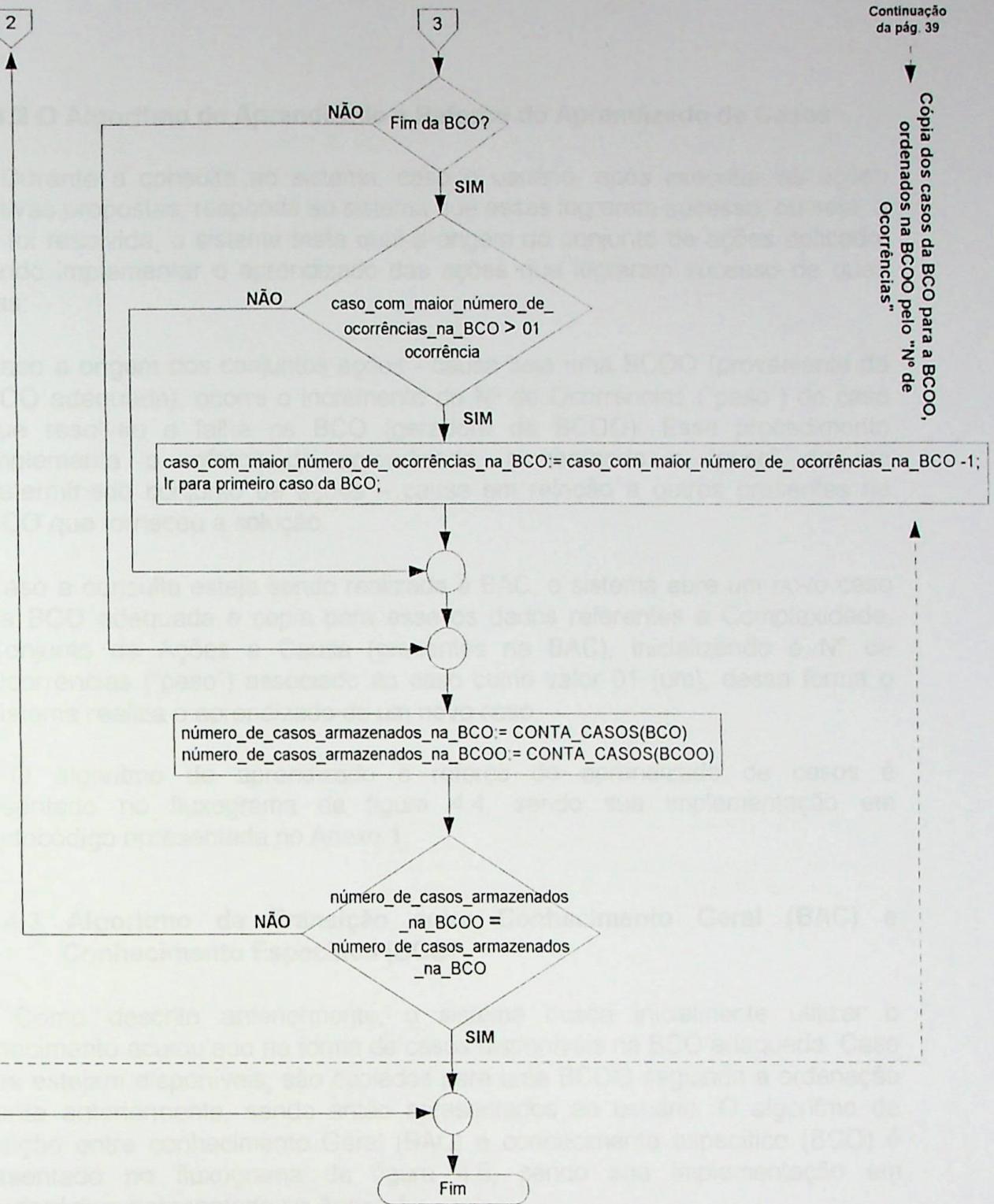


Figura 4.3 – Fluxograma apresentando o algoritmo de seleção e ordenação de casos armazenados (continuação da pág. 39).

os critérios de ordenação apresentados anteriormente. As ações e causas da BCOO são as apresentadas pelo sistema ao usuário. Ao término da consulta a BCOO é esvaziada.

Uma vez que um caso presente na BCOO, apresentado ao usuário, resolva a falha, seu número de ocorrências é atualizado na BCO geradora do BCOO, de forma a implementar o reforço do conhecimento adquirido ao longo do tempo.

4.4.2 O Algoritmo de Aprendizado e Reforço do Aprendizado de Casos

Durante a consulta ao sistema, caso o usuário, após executar as ações corretivas propostas, responda ao sistema que essas lograram sucesso, ou seja, a falha foi resolvida, o sistema testa qual a origem do conjunto de ações aplicado, podendo implementar o aprendizado das ações que lograram sucesso de duas formas:

- Caso a origem dos conjuntos ações - causa seja uma BCOO (proveniente da BCO adequada), ocorre o incremento do Nº de Ocorrências ("peso") do caso que resolveu a falha na BCO (geradora da BCOO). Esse procedimento implementa o reforço do aprendizado, aumentando o "peso" de um determinado conjunto de ações – causa em relação a outros presentes na BCO que forneceu a solução.
- Caso a consulta esteja sendo realizada à BAC, o sistema abre um novo caso na BCO adequada e copia para esse os dados referentes a Complexidade, Conjunto de Ações e Causa (presentes na BAC), inicializando o Nº de Ocorrências ("peso") associado ao caso como valor 01 (um), dessa forma o sistema realiza o aprendizado de um novo caso.

O algoritmo de aprendizado e reforço do aprendizado de casos é apresentado no fluxograma da figura 4.4, sendo sua implementação em pseudocódigo apresentada no Anexo 1.

4.4.3 Algoritmo de Transição entre Conhecimento Geral (BAC) e Conhecimento Específico (BCO)

Como descrito anteriormente, o sistema busca inicialmente utilizar o conhecimento acumulado na forma de casos disponíveis na BCO adequada. Caso esses estejam disponíveis, são copiados para uma BCOO seguindo a ordenação descrita anteriormente, sendo então apresentados ao usuário. O algoritmo de transição entre conhecimento Geral (BAC) e conhecimento específico (BCO) é apresentado no fluxograma da figura 4.5, sendo sua implementação em pseudocódigo apresentada no Anexo 1.

A cada resposta negativa do usuário (insucesso da aplicação do conjunto ações propostas), durante uma consulta a BCOO, a Complexidade do caso aplicado é copiada para uma Base de Casos Consultados (BCC). O objetivo dessa base é manter um histórico dos casos consultados na BCOO, de forma a não apresentá-los novamente ao usuário no caso da consulta mudar dos casos contidos na BCOO para os conjuntos de ações – causa presentes na BAC adequada. A BCC não é permanente, tendo seu conteúdo apagado ao final de cada consulta ao sistema.

Quando os casos presentes na BCOO esgotam-se, sem que a solução da falha tenha sido obtida, o sistema tenta buscar conjuntos de ações – causa na BAC adequada. Caso o número de conjuntos ações - causa presente na BAC adequada seja diferente do número de casos na BCOO, a consulta à BAC prossegue, caso seja igual, o sistema informa ao usuário que o conhecimento disponível no sistema não é capaz de resolver a falha apresentada.

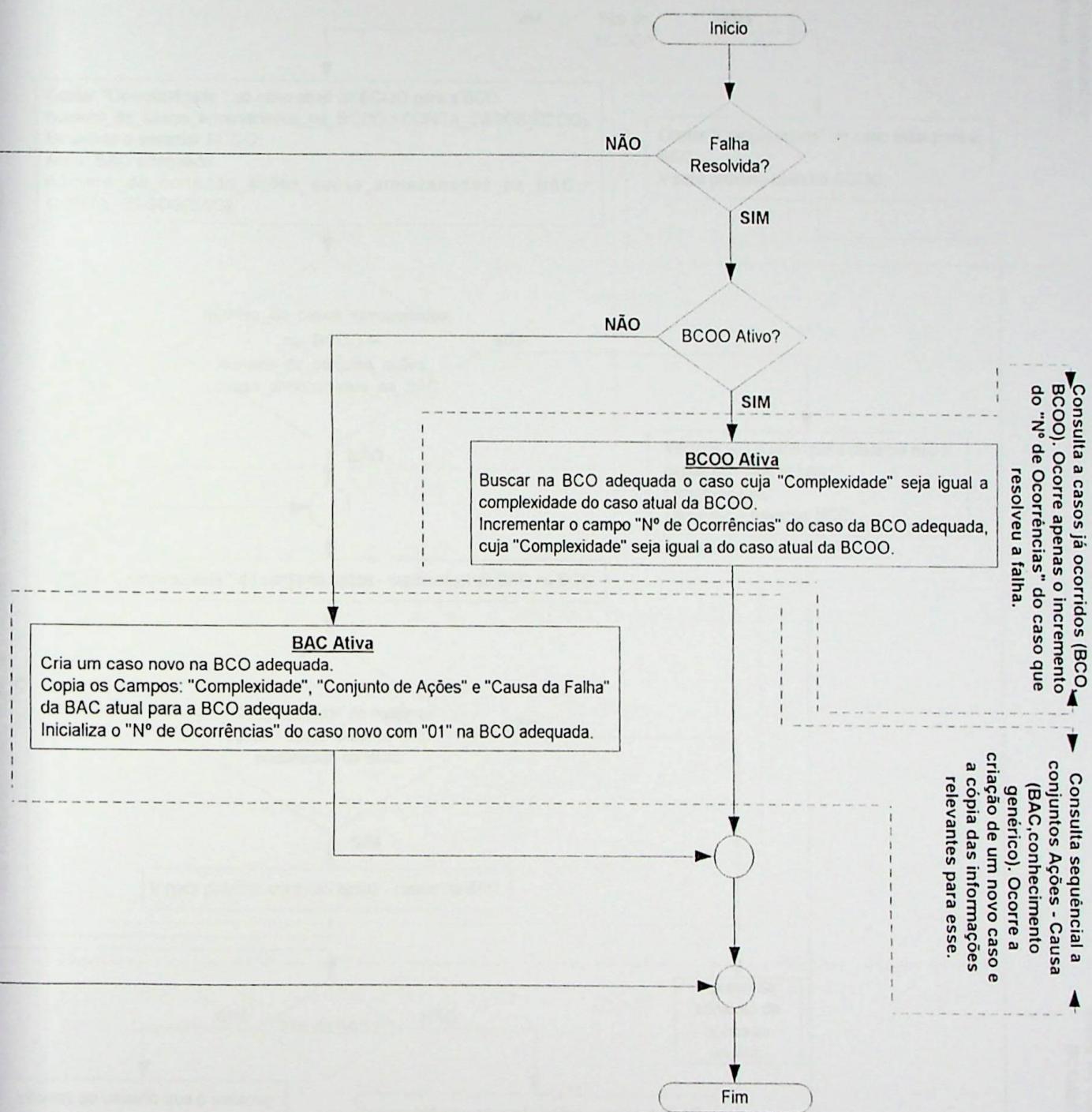


Figura 4.4 – Fluxograma apresentando o algoritmo de aprendizado e reforço do aprendizado de casos.



Consulta a BCO/BCOO.
Casos consultados
registrados na BCC.

Consulta a BAC sem
antes ter consultada
BCO/BCOO.

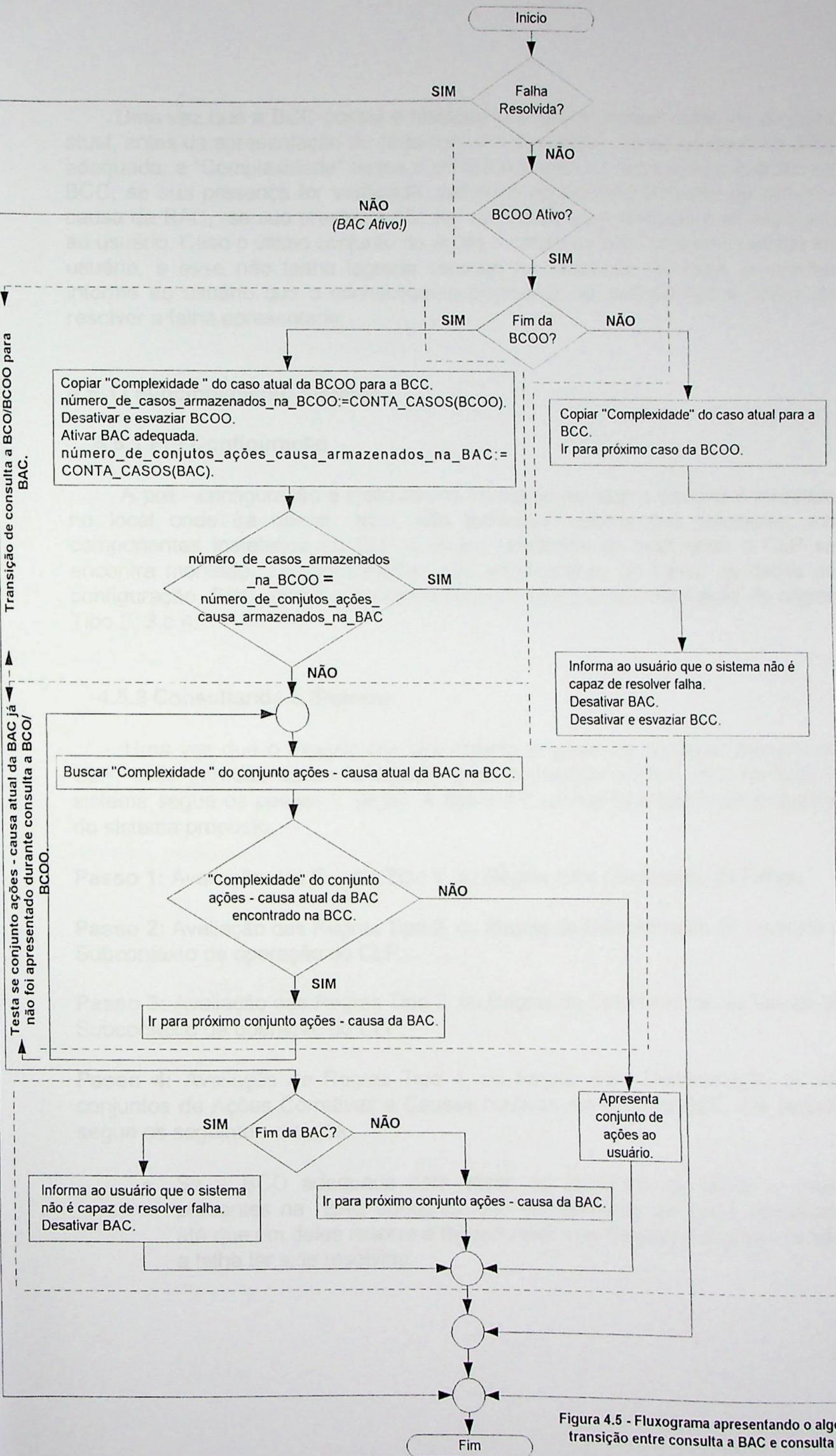


Figura 4.5 - Fluxograma apresentando o algoritmo de transição entre consulta a BAC e consulta a BCO .

Uma vez que a BCC possui o histórico dos casos apresentados na consulta atual, antes da apresentação de cada conjunto de ações – causa presente na BAC adequada, a “Complexidade” desse é comparada com as dos casos presentes na BCC, se sua presença for verificada, passa-se ao próximo conjunto de ações – causa da BAC, se sua presença não for verificada, esse conjunto é apresentado ao usuário. Caso o último conjunto de ações – causa da BAC seja apresentado ao usuário, e esse não tenha logrado sucesso na resolução da falha, o sistema informa ao usuário que o conhecimento disponível no sistema não é capaz de resolver a falha apresentada.

4.5 O Funcionamento Sistema

4.5.1 Pré Configuração

A pré - configuração é realizada no momento em que o sistema é instalado no local onde irá operar. Nela, são fornecidas informações referentes aos componentes instalados no CLP e dados referentes ao local onde o CLP se encontra montado. Tais informações são armazenadas no banco de dados de configuração, BDC, que é uma das fontes de dados para a avaliação de regras Tipo 2, 3 e 4.

4.5.2 Consultando o Sistema

Uma vez que o usuário percebe alguma situação de operação anormal no CLP, esse introduz dados referentes a essa situação e inicia uma consulta, o sistema segue os passos a seguir. A figura 4.6 apresenta o funcionamento geral do sistema proposto.

Passo 1: Avaliação das Regras Tipo 1, ou Regras para Diagnóstico de Falhas.

Passo 2: Avaliação das Regras Tipo 2, ou Regras de Determinação de Contexto e Subcontexto de operação do CLP.

Passo 3: Avaliação das Regras Tipo 3, ou Regras de Determinação da Versão do Subcontexto de operação do CLP.

Passo 4: Avaliação de Regras Tipo 4, ou Regras para Determinação do uso conjuntos de Ações Corretivas e Causas contidos em BAC ou BCO. Tal decisão segue os seguintes critérios.

- Se a BCO adequada esta vazia, os conjuntos de ações – causa presentes na BAC adequada são apresentados de forma seqüencial, até que um deles resolva a falha ou todos os conjuntos esgotem-se sem a falha ter sido resolvida.

- Se a BCO não está vazia, os conjuntos de ações – causa de maior número de ocorrências (“peso”) e menor complexidade contidos nessa são apresentados. Caso todas as ações contidas no BCO sejam aplicadas e a falha não seja solucionada, o sistema retorna à aplicação dos conjuntos ações – causa presentes na BAC adequada, porém, antes de cada conjunto de ações – causa da BAC ser apresentado ao usuário, esse é comparado com os conjuntos apresentados ao usuário durante a consulta a BCO/BCOO, contidos na BCC, para que não haja repetição da apresentação do mesmo conjunto de ações que resultaram em insucesso anteriormente.

Passo 5: A cada conjunto ação – causa apresentado, espera-se que o usuário execute as ações corretivas propostas e observe se, após a aplicação das ações contidas no conjunto corrente, o CLP evoluiu para um estado normal, o que indica falha resolvida. Para que o usuário possa determinar se o CLP evoluiu ou não para um estado normal, o sistema apresenta todos os estados que indicam falha resolvida, e, por comparação entre o estado para o qual o CLP evoluiu e o estado inicial (de falha), determina-se se as ações lograram ou não sucesso.

Passo 6: Uma vez que um conjunto de ações obteve sucesso em resolver a falha, dois procedimentos podem ocorrer:

- Caso a consulta esteja sendo realizada a uma BAC, o conjunto é aprendido pelo sistema, ou seja, passa a fazer parte da BCO adequada.
- Caso a consulta esteja sendo realizada a uma BCO, esse tem seu número de ocorrências (“peso”) aumentado.

4.6 Pré - requisitos para Aplicação da Solução Proposta

Os pré-requisitos para a aplicação da solução proposta são:

- Disponibilidade do conhecimento em relação as falhas que podem ocorrer no CLP para o qual o sistema será desenvolvido, as ações corretivas aplicáveis, os recursos necessários para sua aplicação das ações corretivas e as prováveis causas da falha. Essas podem ser obtidas por consulta a manuais de manutenção, normalmente fornecidos pelo fabricante, podendo ser enriquecidas com a experiência dos profissionais de manutenção envolvidos com o equipamento. A metodologia para aquisição, tratamento e estruturação do conhecimento apresenta uma forma sistemática de tratamento do conhecimento proveniente das fontes propostas.

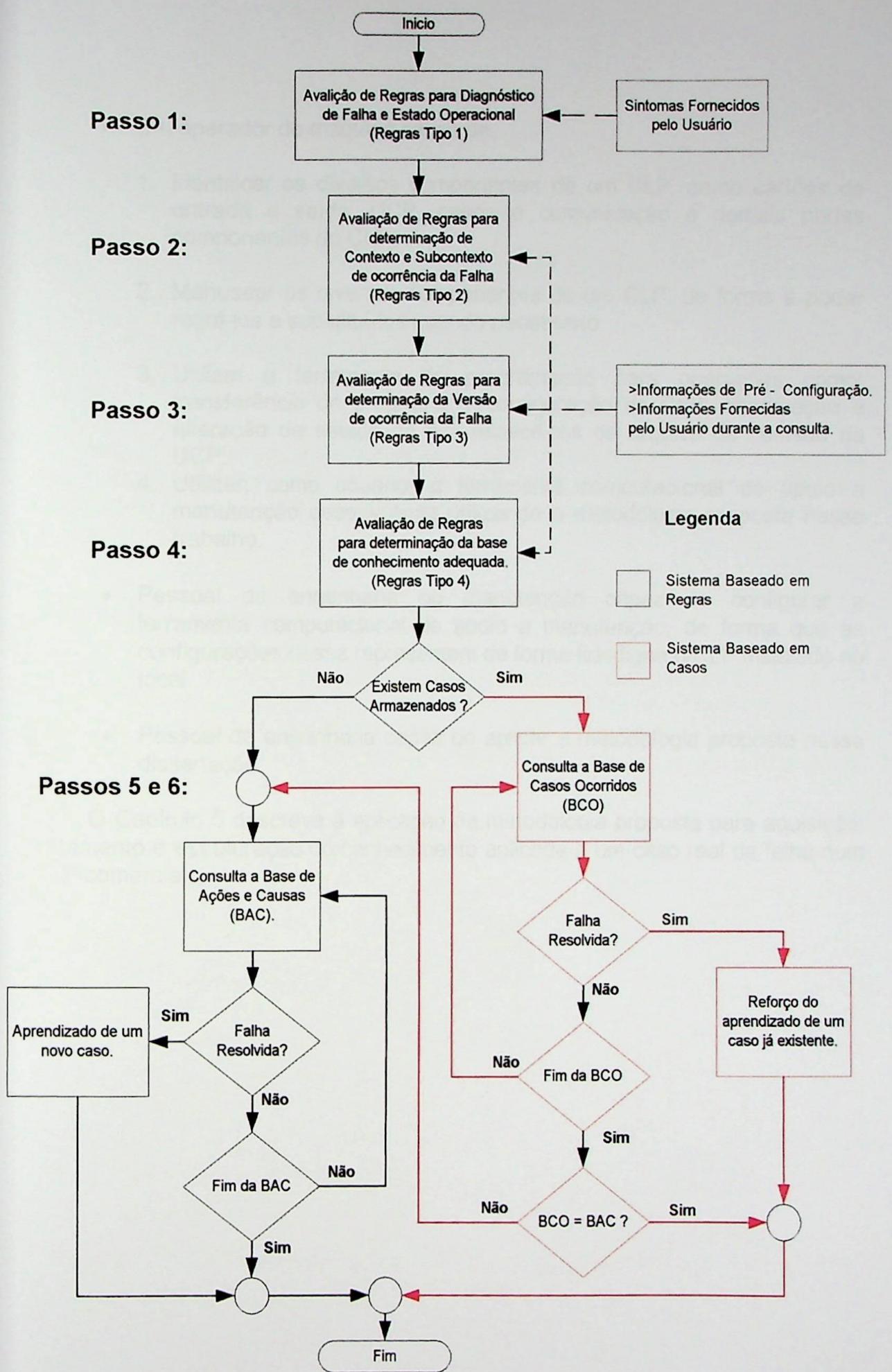


Figura 4.6 – Funcionamento geral do sistema proposto.

- Um operador de máquina capaz de:
 1. Identificar os diversos componentes de um CLP, como cartões de entrada e saída, UCP, cabos e comunicação e demais partes componentes do CLP.
 2. Manusear os diversos componentes de um CLP, de forma a poder retirá-los e substituí-los quando necessário.
 3. Utilizar a ferramenta de programação para operações como: transferência de programa, reconfiguração do CLP, observação e alteração de estado de bits específicos do arquivo de estado da UCP.
 4. Utilizar, como usuário, a ferramenta computacional de apoio a manutenção desenvolvida utilizando a metodologia proposta nesse trabalho.
- Pessoal de engenharia ou manutenção capaz de configurar a ferramenta computacional de apoio a manutenção, de forma que as configurações dessa representem de forma fidedigna o CLP instalado no local.
- Pessoal de engenharia capaz de aplicar a metodologia proposta nessa dissertação.

O Capítulo 5 descreve a aplicação da metodologia proposta para aquisição, tratamento e estruturação do conhecimento aplicada a um caso real de falha num CLP comercial.

Capítulo 5

Aplicando a Metodologia Proposta

Nesse capítulo é apresentado um exemplo completo da aplicação da Metodologia para Aquisição, Tratamento e Estruturação do Conhecimento para Diagnóstico e Solução de Falhas em CLP a uma falha de comunicação de um CLP real.

5.1 O CLP Selecionado

Para exemplificar a aplicação da metodologia proposta no capítulo 4, que é geral (aplica-se a qualquer CLP de qualquer fabricante), escolheu-se um CLP comercial de estrutura modular, o Rockwell Automation – Allen Bradley SLC 500¹ UCP 03 tipo 2 (SLC 5/03). Tal UCP é mostrada na figura 5.1 sendo que a tabela 5.1 apresenta suas características técnicas relevantes.

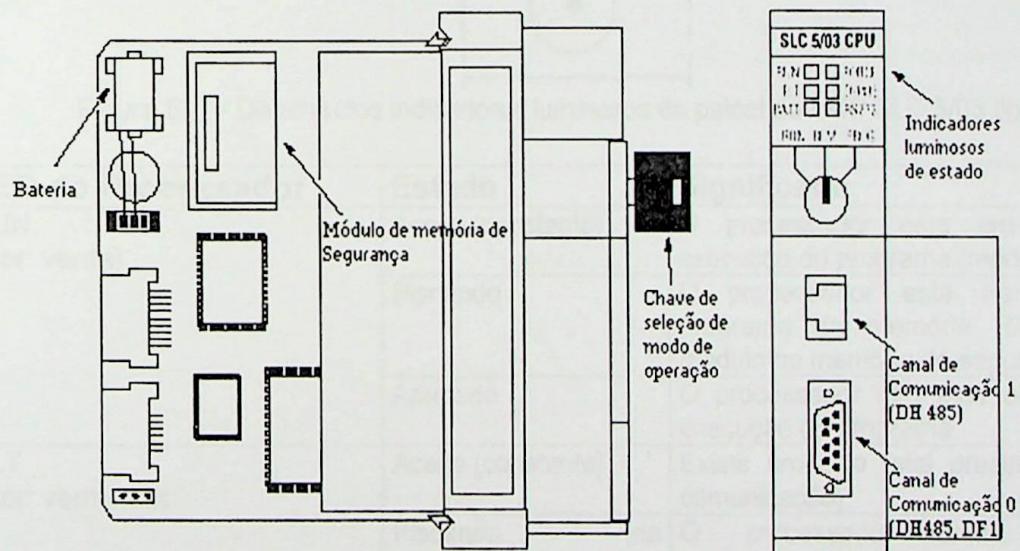


Figura 5.1 – O CLP Rockwell Automation – Allen Bradley SLC 500 UCP 03 tipo 2 (SLC 5/03) – Painel Frontal.

¹ Allen Bradley SLC 500 é marca registrada da Rockwell Automation. Todos os direitos reservados.

UCP (Código de catálogo)	Memória	Nº max. Ptos. E/S local	Canais de Comunicação da UCP (Protocolos)
SLC 5/03 (1747 – L532)	16k palavras	4096	Canal 0 : RS-232 (DF-1, DH485). Canal 1: RS-485 (DH-485).

Tabela 5.1 - Características técnicas da UCP escolhida.

Na UCP em questão, existe um conjunto de indicadores luminosos (chamados de LEDs) presentes no painel frontal, cujos estados podem ser utilizados para diagnosticar o estado de operação atual e a presença de diversos tipos de falhas. A figura 5.2 mostra com detalhes esses indicadores luminosos. A tabela 5.2 descreve o significado de cada estado assumido por cada indicador luminoso.

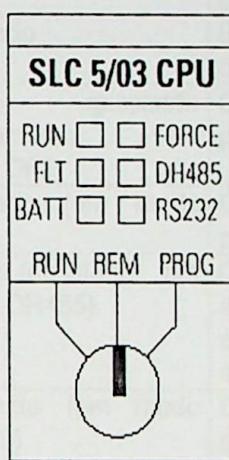


Figura 5.2 – Detalhe dos indicadores luminosos do painel da UCP SLC 5/03 tipo 2.

LED do Processador	Estado	Significado
RUN (Cor: verde)	Aceso (constante)	O processador esta em modo de execução do programa (modo "RUN").
	Piscando	O processador está transferindo o programa da memória RAM para o módulo de memória de segurança.
	Apagado	O processador não esta em modo de execução do programa.
FLT (Cor: vermelho)	Aceso (constante)	Existe um erro fatal presente (não há comunicação).
	Piscando (na energização da UCP)	O processador ainda não foi configurado.
	Piscando (durante a operação normal da UCP)	O processador detectou um erro no processador, chassis de expansão ou na memória.
	Apagado	Não existem erros presentes.

Tabela 5.2 – Significado do estados assumidos pelos indicadores luminosos do painel frontal da UCP SLC 5/03 tipo 2.

BATT (Cor: vermelho)	Aceso (constante)	O nível de tensão da bateria está abaixo do mínimo, a bateria não está instalada ou o conector da bateria está solto.
	Apagado	A bateria está em bom estado.
FORCE (Cor: amarelo)	Aceso (constante)	Os estados forçados foram habilitados.
	Piscando	Um ou mais endereços de entrada e/ou saída física foram forçados em estado ligado/ desligado mas a aplicação do estado forçado não foi habilitada.
	Apagado	Não existem entradas e saídas físicas forçadas e/ou habilitadas.
DH485 (Cor: verde)	Aceso (constante)	O bit de comunicação ativa (S:1/7) do arquivo de estado do sistema está ativo e o processador está comunicando-se em rede.
	Piscando	O processador está tentando estabelecer comunicação, mas não existem outros nós ativos na rede.
	Apagado	Erro fatal presente no canal 1 (não há comunicação).
RS232 (Cor: verde)	Aceso (piscando rapidamente, em modo DF1)	O SLC 5/03 está transmitindo dados para a rede.
	Apagado (em modo DF1)	O SLC 5/03 não está transmitindo dados para a rede.
	Aceso (constante, em modo DH485)	O bit de comunicação ativa (S:1/7) do arquivo de estado do sistema está ativo e o processador está se comunicando em rede.
	Piscando (em modo DH485)	O processador está tentando estabelecer comunicação, mas não existem outros nós ativos na rede.
	Apagado (em modo DH 485)	Erro fatal presente no canal 0 (não há comunicação).

Tabela 5.2 – Significado do estados assumidos pelos indicadores luminosos do painel frontal da UCP SLC 5/03 tipo 2 (Continuação).

A fonte de energia do CLP também possui um indicador luminoso (LED) que indica seu estado de operação, sendo o estado desse relevante para o diagnóstico de falhas no CLP. A tabela 5.3 apresenta seus estados.

LED da Fonte	Estado	Significado
POWER (Cor: vermelho)	Aceso	Fonte de alimentação em operação normal.
	Apagado	Fonte de alimentação com falha.

Tabela 5.3 – Estado do indicador luminoso da fonte de alimentação do CLP.

A escolha do CLP em questão deve-se aos seguintes fatos:

- Disponibilidade da base de conhecimento necessária à confecção das regras e ações corretivas para o sistema proposto, disponíveis no manual do fabricante [13].
- Conhecimento do equipamento por parte do autor, que trabalha a alguns anos com esse CLP, ou seja, a disponibilidade do especialista no domínio abordado.
- Disponibilidade do equipamento para testes.

No CLP especificado acima, escolheu-se um falha de comunicação no canal 1 (CN1, que utiliza um protocolo de comunicação proprietário denominado DH485), pois essa, devido a sua complexidade, permite que todos os tópicos apresentados na metodologia sejam aplicados.

5.2 Aplicando a Metodologia para Aquisição, Tratamento e Estruturação do Conhecimento referente ao Diagnóstico e Solução de Falhas em CLP

5.2.1 Aquisição do Conhecimento

5.2.1.1 Apresentação da Falha, Causas Prováveis e Ações Recomendas pelo Fabricante

A falha ocorre no canal 1 de comunicação da UCP em questão. O fabricante a descreve da seguinte forma: "O processador SLC 5/03 ou 5/04 está tentando estabelecer comunicação, mas não consegue encontrar outros nós ativos."

Na tabela 5.4 é reproduzida a forma (traduzida para o português) como o fabricante apresenta a falha, causas prováveis e ações corretivas aplicáveis à cada causa provável. Essas informações encontram-se disponíveis no manual do fabricante [13] e são o ponto de partida para o desenvolvimento dos demais tópicos da metodologia proposta.

Aqui cabem explicações sobre alguns termos usados pelo fabricante da apresentação da falha:

- **Ferramenta de programação:** é um microcomputador pessoal (PC) que possui instalado os programas para comunicação e programação da UCP em questão. Para se estabelecer comunicação entre o PC e o CLP, normalmente utiliza-se uma das portas seriais do PC (portas RS-232) conectando-se essa a um conversor de sinal (normalmente de padrão RS-232 para padrão RS-485) ou instala-se uma placa de comunicação específica (comercializada pelo fabricante). No caso desse trabalho considera-se que a comunicação será feita utilizando a porta serial do PC e o conversor de sinal.

- Módulo PIC:** é o de conversor sinal citado anteriormente. Converte o sinal proveniente da interface serial padrão RS-232 do PC para um sinal padrão RS-485. O sinal padrão RS-485 é mais adequado a transmissão de dados em ambientes industriais, não só pela maior imunidade a ruído quanto pela possibilidade da construção de redes de CLPs com centenas de metros de comprimento e dezenas de nós. Consiste num cabo que conecta o PC ao conversor, o conversor propriamente dito e um cabo que conecta o conversor ao canal 1 do CLP (porta serial padrão RS-485).

Falha	Causa Provável	Ações Recomendadas
O processador SLC 5/03 ou 5/04 esta tentando estabelecer comunicação mas não consegue encontrar outros nós ativos (o LED está piscando em verde no SLC 5/03 e 5/04).	1) Parâmetros de Comunicação DH485 ou DH+ ajustados de forma imprópria.	1) Conferir os parâmetros de comunicação com a ferramenta de programação. A taxa de transmissão da ferramenta de programação e do processador devem ser iguais. Os endereços da ferramenta de programação e do processador devem ser diferentes. 2) Tentar diferentes combinações de : a) taxa de transmissão (o padrão do processador é 19,2k para DH485 no SLC 5/03 e 57,6k para DH+ no SLC 5/04). b) Endereço do nó (o padrão é 1). 3) Tentar aumentar o máximo endereço de nó (o padrão é 31 para o SLC 5/03). • Observar o manual do usuário do programa de comunicação para informações sobre as configurações do canal.
	2) Dispositivo de comunicação mal conectado.	1) Conferir a continuidade do cabo. 2) Conferir as conexões do cabo entre a ferramenta de programação e o processador. 3) Conferir o dispositivo de comunicação (por exemplo, o PIC).
	3) Nível de tensão baixo ou nulo para o dispositivo de comunicação.	1) Verificar se a fonte está bem dimensionada e a carga consumida pelo barramento (o PIC e o AIC consomem corrente do barramento).

Tabela 5.4 – Reprodução da forma como o fabricante do CLP SLC 500 UCP 5/03 tipo 2 apresenta a falha de comunicação no canal 1 (DH 485), causas prováveis e ações corretivas.

- Módulo AIC:** é utilizado quando o CLP esta conectado numa rede para isolar eletricamente o cabo de rede do canal de comunicação do CLP. O cabo de rede é conectado ao módulo AIC e o mesmo é conectado a porta de comunicação serial (RS-485) CLP por um cabo específico. O AIC recebe alimentação externa ou da própria UCP (quando não esta ligado a nenhuma fonte externa). Possui um indicador visual (LED) que indica se esse está ou não energizado. O módulo AIC não faz parte do chassis do CLP, sendo montado a parte. A figura 5.3 apresenta o conjunto ferramenta de programação, módulo PIC, módulo AIC e CLP, assim como a conexão entre esses elementos, utilizando os cabos adequados.

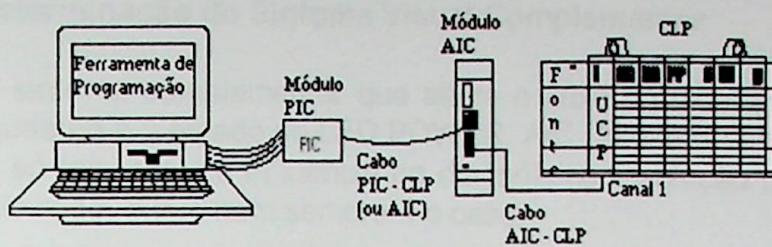


Figura 5.3 – O conjunto ferramenta de programação, módulo PIC, módulo AIC e CLP, assim como a conexão entre esses elementos, utilizando os cabos adequados

5.2.1.2 Identificação Visual da Falha

A configuração visual assumida pelos LEDs no caso da falha em questão é apresentada na figura 5.4.

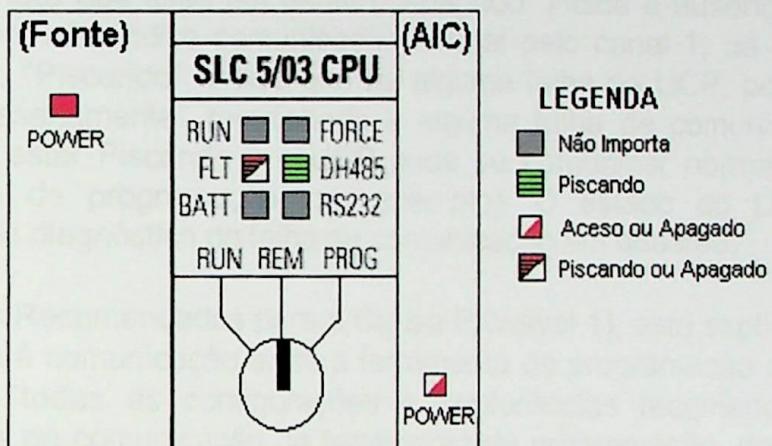


Figura 5.4 - A configuração visual assumida pelos LEDs do painel do CLP no caso da falha de comunicação em questão.

Aqui cabe notar que o estado assumido pelos demais LEDs não é relevante. Apesar de importante para a identificação de falhas de comunicação em rede, o fabricante omite da identificação visual o estado do indicador visual de alimentação do módulo AIC (LED POWER AIC).

5.2.1.3 Determinação dos Sintomas Típicos da Falha

A partir da identificação visual da falha, podemos estabelecer os sintomas típicos da falha, ou seja, aqueles que a diferenciam das outras falhas possíveis. A tabela 5.5 apresenta os sintomas típicos no caso da falha de comunicação em questão.

LED	Estado
POWER (fonte de alimentação):	ACESO
FLT:	PISCANDO ou APAGADO
DH485:	PISCANDO

Tabela 5.5 – Sintomas típicos de falha de comunicação no canal 1, fornecidos pelo fabricante.

5.2.1.4 Determinação do Sintoma Visual Complementar

O único sintoma complementar que altera o fluxo do diagnóstico e solução da falha em questão é o estado do LED POWER AIC (Aceso ou Apagado). Porém esse sintoma só tem alguma influência no diagnóstico e solução da falha caso o CLP seja nó de rede, o que nem sempre é o caso.

5.2.1.5 Pré Tratamento do Conhecimento referente a Causas e Ações Corretivas

- Na identificação visual da falha, o estado do LED FLT não acrescenta absolutamente nenhuma informação relevante ao diagnóstico. Isso dá-se devido ao fato que esse em estado “Apagado” indica a ausência de falha de UCP, porém não indica comunicação normal pelo canal 1; da mesma forma, seu estado “Piscando” indica que há alguma falha de UCP, porém essa não está necessariamente relacionada a alguma falha de comunicação (o LED FLT pode estar Piscando e a UCP pode se comunicar normalmente com a ferramenta de programação, por exemplo). O estado do LED FLT será ignorado no diagnóstico da falha de comunicação em questão.
- Nas Ações Recomendadas para a Causa Provável 1), está explícito que essas aplicam-se à comunicação entre a ferramenta de programação e o canal 1 do CLP, pois todas as configurações e conferências recomendadas alteram parâmetros de comunicação da ferramenta de programação, de forma a fazer com que essa torne-se apta à comunicação com o CLP.
- Na Causa Provável 3), cabem alguns comentários:
 1. Caso a fonte de alimentação do CLP esteja mal dimensionada, essa entra em estado de sobrecarga, o que faz o CLP todo, e não apenas a comunicação pelo canal 1, entrar em colapso. O problema da fonte de alimentação sobrecarregada é tratado como Falha no Sistema de Energia, tendo precedência sobre todas as demais falhas, ou seja, caso essa se manifeste, todas as demais são ignoradas até que a mesma seja resolvida.
 2. Caso a chave seletora do nível de tensão da rede esteja configurada para um nível incompatível com o existente na rede elétrica conectada ao CLP, novamente temos uma Falha no Sistema de Energia.

5.2.2 Tratamento do Conhecimento

5.2.2.1 Determinação da Hierarquia de Diagnóstico de Falhas no CLP

No caso da UCP SLC 500 5/03, a hierarquia no diagnóstico das falhas, na forma de fluxograma, é apresentada na figura 5.5. É importante notar que a detecção correta da falha de comunicação só é realizada se todas as falhas que a precedem, e tem maior gravidade, não são diagnosticadas.

5.2.2.2 Determinação dos Contextos, Subcontextos e Versões de Ocorrência da Falha

A partir da análise do conhecimento disponível e do conhecimento do especialista, pode-se determinar diversos contextos, subcontextos e versões da ocorrência da falha. Esses são mostrados na tabela 5.6.

5.2.2.3 Determinação dos Recursos Aplicáveis para Solução da Falha

Os recursos aplicáveis a solução da falha são:

- **Ferramenta de Programação + Módulo PIC + Cabo PIC – CLP:** para comunicação entre PC e CLP via canal1 (DH485) e eventual reconfiguração de parâmetros do canal 1 incorretamente ajustados.
- **Ferramenta de Programação + Cabo RS-232:** para comunicação entre PC e CLP via canal 0 (protocolos DF1 ou DH485) e eventual reconfiguração de parâmetros do canal 1 incorretamente ajustados, será utilizado quando o Módulo e cabo PIC não estiverem disponíveis para reconfiguração via canal 1 ou quando todas as tentativas de comunicação via canal 1 forem esgotadas.
- **Módulo PIC reserva:** para ações corretivas que exijam a substituição do módulo PIC em uso para determinar-se se esse é o elemento falho.
- **Módulo AIC reserva:** para ações corretivas que exijam a substituição do módulo AIC em uso para determinar-se se esse é o elemento falho.
- **Cabo PIC – CLP reserva:** para ações corretivas que exijam a substituição do cabo PIC - CLP em uso para determinar-se se esse é o elemento falho.
- **Cabo AIC – CLP reserva:** para ações corretivas que exijam a substituição do cabo AIC - CLP em uso para determinar-se se esse é o elemento falho.

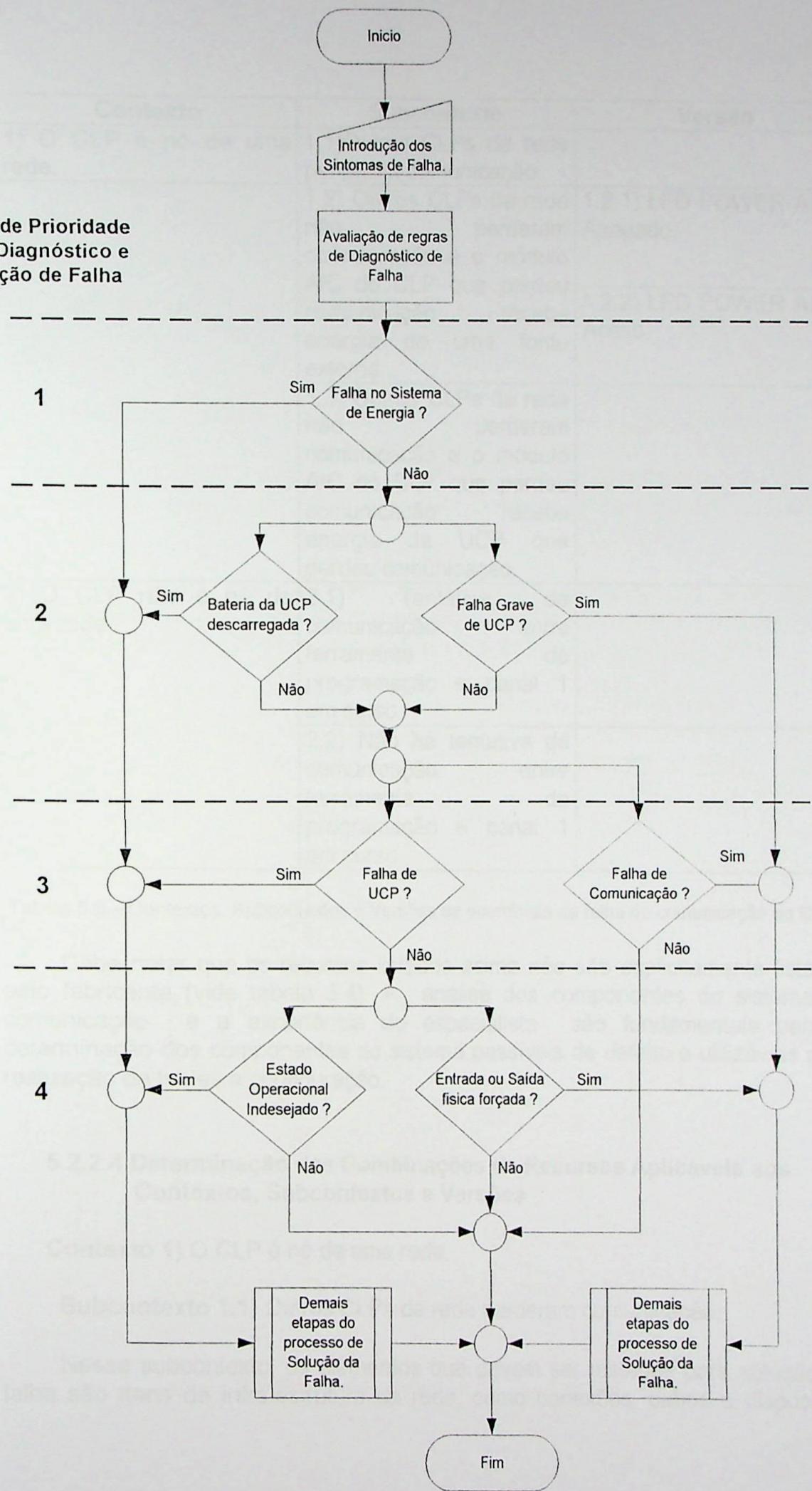


Figura 5.5 - Fluxograma da hierarquia de diagnóstico de falhas na UCP SLC 500 5/03.

Contexto	Subcontexto	Versão
1) O CLP é nó de uma rede.	1.1) Outros CLPs da rede perderam comunicação.	
	1.2) Outros CLPs da rede não perderam comunicação e o módulo AIC do CLP que perdeu comunicação recebe energia de uma fonte externa.	1.2.1) LED POWER AIC = Apagado. 1.2.2) LED POWER AIC = Aceso.
	1.3) Outros CLPs da rede não perderam comunicação e o módulo AIC do CLP que perdeu comunicação recebe energia da UCP que perdeu comunicação.	
2) O CLP não é nó de uma rede.	2.1) Tentativa de comunicação entre ferramenta de programação e canal 1 em curso.	
	2.2) Não há tentativa de comunicação entre ferramenta de programação e canal 1 em curso.	

Tabela 5.6 – Contextos, Subcontextos e Versões de ocorrência da falha de comunicação no CN 1.

Cabe notar que os recursos listados acima não são explicitamente listados pelo fabricante (vide tabela 5.4). A análise dos componentes do sistema de comunicação e a experiência do especialista são fundamentais para a determinação dos componentes do sistema passíveis de defeito e utilizáveis para realização de testes e configuração.

5.2.2.4 Determinação das Combinações de Recursos Aplicáveis aos Contextos, Subcontextos e Versões

Contexto 1) O CLP é nó de uma rede.

Subcontexto 1.1) Outros CLPs da rede perderam comunicação;

Nesse subcontexto, os elementos que devem ser testados para solução da falha são itens de infra-estrutura da rede, como conexões, cabos e disposição

física dos elementos que compõe a rede, assim, não serão definidos recursos necessários, pois dependem das instalações do usuário.

Subcontexto 1.2) Outros CLPs da rede não perderam comunicação e o módulo AIC do CLP que perdeu comunicação recebe energia de uma fonte externa.

Subcontexto 1.3) Outros CLPs da rede não perderam comunicação e o módulo AIC do CLP que perdeu comunicação recebe energia da UCP que perdeu comunicação.

Nos **Subcontextos 1.2 e 1.3**, os recursos aplicáveis são os mesmos, apenas a ordem dos testes ações corretivas altera-se de acordo com a **Versão**, sendo essa função do estado do LED AIC POWER. A **Versão 1** refere-se ao LED AIC POWER = **Apagado**, enquanto a **Versão 2** refere-se ao LED AIC POWER = **Aceso**.

A tabela 5.7 lista sistematicamente todas as combinações possíveis de recursos aplicáveis. As células preenchidas com **Sim** indicam a disponibilidade do recurso, as preenchidas com **Não** indicam a não disponibilidade do recurso.

Nº Combinação	Combinações de Recursos Aplicáveis			
	Ferram. Prog. + Mód. PIC + Cabo PIC – CLP	Ferram. Prog. + Cabo RS-232	Mód. AIC reserva	Cabo AIC – CLP reserva
0	Não	Não	Não	Não
1	Não	Não	Não	Sim
2	Não	Não	Sim	Não
3	Não	Não	Sim	Sim
4	Não	Sim	Não	Não
5	Não	Sim	Não	Sim
6	Não	Sim	Sim	Não
7	Não	Sim	Sim	Sim
8	Sim	Não	Não	Não
9	Sim	Não	Não	Sim
10	Sim	Não	Sim	Não
11	Sim	Não	Sim	Sim
12	Sim	Sim	Não	Não
13	Sim	Sim	Não	Sim
14	Sim	Sim	Sim	Não
15	Sim	Sim	Sim	Sim

Tabela 5.7 - Combinações de Recursos Aplicáveis ao Contexto 1, Subcontextos 1.2 e 1.3 da falha de comunicação no CN 1.

Contexto 2) O CLP não é nó de uma rede.

Subcontexto 2.1) Tentativa de comunicação entre ferramenta de programação e canal 1 em curso.

Nesse subcontexto, os recursos aplicáveis e suas combinações são listados na tabela 5.8.

Nº da Combinação	Combinações de Recursos Aplicáveis			
	Ferram. Prog. + Mód. PIC + Cabo PIC – CLP	Ferram. Prog. + Cabo RS-232	Mód. PIC reserva	Cabo PIC – CLP reserva
0	Não	Não	Não	Não
1	Não	Não	Não	Sim
2	Não	Não	Sim	Não
3	Não	Não	Sim	Sim
4	Não	Sim	Não	Não
5	Não	Sim	Não	Sim
6	Não	Sim	Sim	Não
7	Não	Sim	Sim	Sim
8	Sim	Não	Não	Não
9	Sim	Não	Não	Sim
10	Sim	Não	Sim	Não
11	Sim	Não	Sim	Sim
12	Sim	Sim	Não	Não
13	Sim	Sim	Não	Sim
14	Sim	Sim	Sim	Não
15	Sim	Sim	Sim	Sim

Tabela 5.8 - Combinações de Recursos Aplicáveis ao Contexto 2, Subcontextos 2.1 da falha de comunicação no CN 1.

Subcontexto 2.2) Não há tentativa de comunicação entre ferramenta de programação e canal 1 em curso.

Nesse subcontexto, não há necessidade de determinar-se combinações de recursos, pois o mesmo não constitui uma falha no canal de comunicação 1.

5.2.2.5 Seleção das Combinações de Recursos Aplicáveis de acordo com os Contextos e Subcontextos e Versões

A tabela 5.9 apresenta as combinações de recursos cuja disponibilidade torna possível a geração de ações de teste e correção de falhas de comunicação no canal 1. Cabe notar que, dependendo do subcontexto, mesmo a ausência total de recursos disponíveis (caso da linha 0 das tabelas de combinações de recursos aplicáveis) ainda é possível a aplicação de algumas ações corretivas e testes, como testes visuais e testes de conexão correta de cabos, por exemplo.

Contexto	Subcontexto	Nº da Combinação (Tabela)
1	1.1	Nenhum, depende das instalações locais (não definida)
	1.2	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 (5.7)
	1.3	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 (5.7)
2	2.1	4,8,9,10,11,12,13,14,15 (5.8)
	2.2	Nenhum, não é falha.

Tabela 5.9 - Seleção das Combinações de Recursos Aplicáveis de acordo com os Contextos e Subcontextos e Versões da falha de comunicação no CN 1.

No subcontexto 2.1, nota-se que as combinações 0,1,2,3,5,6,7 não são listadas como geradoras de ações e testes, isso dá - se pelo fato que os recursos presentes nessas combinações não atendem o mínimo necessário para que alguma ação seja proposta.

5.2.3 Estruturação do Conhecimento

5.2.3.1 Classificação e Seqüência de Aplicação das Ações Corretivas aplicáveis à Solução de Falhas

A classificação e seqüência de aplicação das ações corretivas é apresentada a seguir:

- 1) Ações referentes a detecção do elemento falho por inspeção visual ou teste eletro-mecânico (por teste eletro-mecânico entende-se testes de conexão elétrica).
- 2) Ações que exijam a substituição de elementos de conexão (como cabos) e elementos não conectados ao chassi do CLP.
- 3) Ações de conexão da ferramenta de programação e configuração do canal de comunicação e ações que exija a substituição de elementos conectados ao chassi do CLP.

Cabe ressaltar que a classificação apresentada acima parte de ações de menor complexidade (testes visuais) para ações de maior complexidade (conexão de ferramenta de programação ou desmonte de partes do sistema). No caso do item 3), pode ser necessário desenergizar o CLP, o que implica em parada do processo controlado pelo mesmo.

5.2.3.2 Determinação dos Conjuntos de Ações Corretivas e Causas

A seguir são apresentadas as ações corretivas aplicáveis para cada um dos contextos, subcontextos e versões, divididas em conjuntos de ações. Nesse ponto,

as restrições de recursos disponíveis ainda não são consideradas, sendo consideradas apenas a seqüência de aplicação apresentada no item 5.2.3.1.

Contexto 1) O CLP é nó de uma rede.

Subcontexto 1.1) Outros CLPs da rede perderam comunicação: consultar Tabela A1 do Anexo 3.

Subcontexto 1.2) Outros CLPs da rede não perderam comunicação e o módulo AIC do CLP que perdeu comunicação recebe energia de uma fonte externa. Versões 1 (LED POWER AIC = Apagado) e 2 (LED POWER AIC = Aceso): consultar Tabela A2 do Anexo 3.

Subcontexto 1.3) Outros CLPs da rede não perderam comunicação e o módulo AIC do CLP que perdeu comunicação recebe energia da UCP que perdeu comunicação: consultar Tabela A3 do Anexo 3.

Contexto 2) O CLP não é nó de uma rede.

Subcontexto 2.1) Tentativa de comunicação entre ferramenta de programação e canal 1 em curso: consultar Tabela A4 do Anexo 3.

Subcontexto 2.2) Não há tentativa de comunicação entre ferramenta de programação e canal 1 em curso.

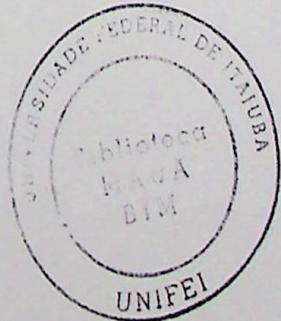
Observando as tabelas dos conjuntos de ações corretivas e causas referentes a cada contexto/subcontexto/versão, pode-se perceber que as mesmas baseiam-se nas informações do fabricante (tabela 5.4), porém são acrescidas de informações provenientes da experiência do especialista, recebendo um maior detalhamento, divisão em passos (num mesmo conjunto de ações), inclusão novas ações e utilização recursos não apresentados explicitamente pelo fabricante.

5.2.3.3 Associação dos Conjuntos de Ações Corretivas e Causas aos Contextos/Subcontextos/Versões e Combinações de Recursos Aplicáveis

Uma vez definidos os contextos/subcontextos/versões, os recursos disponíveis, suas combinações geradoras de ações corretivas e as ações corretivas aplicáveis à cada contexto/subcontexto/versão, podemos selecionar as ações aplicáveis à cada combinação de recursos. Nesse ponto da metodologia podemos também definir as BACs que conterão as ações adequadas à cada combinação de recursos. A tabela 5.10 apresenta a seleção das ações corretivas de acordo com as combinações de recursos e as associa as BACs adequadas.

Contexto	Subcontexto	Versão	Nº da Combinação de Recursos Aplicáveis (Nº da Tabela)	Conjuntos de Ações Aplicáveis (Nº do Anexo - Nº da Tabela)	Nº BAC (BCO)
1	1.1	-	- (-)	1,2,3,4 (A3-1)	01
			00 (5.7)	1,2,3 (A3-2)	02
			01 (5.7)	1,2,3,4 (A3-2)	03
			02 (5.7)	1,2,3,5 (A3-2)	04
			03 (5.7)	1,2,3,4,5 (A3-2)	05
			04 (5.7)	1,2,3,6 (A3-2)	06
			05 (5.7)	1,2,3,4,6 (A3-2)	07
			06 (5.7)	1,2,3,5,6 (A3-2)	08
			07 (5.7)	1,2,3,4,5,6 (A3-2)	09
			08 (5.7)	1,2,3,7 (A3-2)	10
			09 (5.7)	1,2,3,4,7 (A3-2)	11
			10 (5.7)	1,2,3,5,7 (A3-2)	12
			11 (5.7)	1,2,3,4,5,7 (A3-2)	13
			12 (5.7)	1,2,3,6,7 (A3-2)	14
			13 (5.7)	1,2,3,4,6,7 (A3-2)	15
	1.2	1	14 (5.7)	1,2,3,5,6,7 (A3-2)	16
			15 (5.7)	1,2,3,4,5,6,7 (A3-2)	17
			00 (5.7)	1,2 (A3-2)	18
			01 (5.7)	1,2,3 (A3-2)	19
			02 (5.7)	1,2,4 (A3-2)	20
			03 (5.7)	1,2,3,4 (A3-2)	21
			04 (5.7)	1,2,5 (A3-2)	22
			05 (5.7)	1,2,3,5 (A3-2)	23
			06 (5.7)	1,2,4,5 (A3-2)	24
			07 (5.7)	1,2,3,4,5 (A3-2)	25
			08 (5.7)	1,2,6 (A3-2)	26
			09 (5.7)	1,2,3,6 (A3-2)	27
			10 (5.7)	1,2,4,6 (A3-2)	28
			11 (5.7)	1,2,3,4,6 (A3-2)	29
			12 (5.7)	1,2,5,6 (A3-2)	30
			13 (5.7)	1,2,3,5,6 (A3-2)	31
			14 (5.7)	1,2,4,5,6 (A3-2)	32
			15 (5.7)	1,2,3,4,5,6 (A3-2)	33
	1.3	-	00 (5.7)	1,2 (A3-3)	34
			01 (5.7)	1,2,3 (A3-3)	35
			02 (5.7)	1,2,4 (A3-3)	36
			03 (5.7)	1,2,3,4 (A3-3)	37
			04 (5.7)	1,2,5 (A3-3)	38
			05 (5.7)	1,2,3,5 (A3-3)	39
			06 (5.7)	1,2,4,5 (A3-3)	40
			07 (5.7)	1,2,3,4,5 (A3-3)	41
			08 (5.7)	1,2,6 (A3-3)	42
			09 (5.7)	1,2,3,6 (A3-3)	43
			10 (5.7)	1,2,4,6 (A3-3)	44
			11 (5.7)	1,2,3,4,6 (A3-3)	45
			12 (5.7)	1,2,5,6 (A3-3)	46
			13 (5.7)	1,2,3,5,6 (A3-3)	47
			14 (5.7)	1,2,4,5,6 (A3-3)	48
			15 (5.7)	1,2,3,4,5,6 (A3-3)	49

Tabela 5.10 - Seleção das ações corretivas de acordo com as combinações de recursos e sua associação as BACs .



2	2.1	-	04 (5.8)	4 (A3-4)	50
			08 (5.8)	1,2,3 (A3-4)	51
			09 (5.8)	1,2,3,6 (A3-4)	52
			10 (5.8)	1,2,3,5 (A3-4)	53
			11 (5.8)	1,2,3,5,6 (A3-4)	54
			12 (5.8)	1,2,3,4 (A3-4)	55
			13 (5.8)	1,2,3,4,6 (A3-4)	56
			14 (5.8)	1,2,3,4,5 (A3-4)	57
			15 (5.8)	1,2,3,4,5,6 (A3-4)	58
			2.2	- (-)	Não é falha

Tabela 5.10 - Seleção das ações corretivas de acordo com as combinações de recursos e sua associação as BACs (continuação).

5.2.3.4 Criação da Árvore de Decisões

A árvore de decisões é uma representação gráfica (fluxograma) de como ocorre o processo de diagnóstico, determinação de contexto/subcontexto/versão, avaliação dos recursos disponíveis e determinação da base de ações e causas (BAC) ou base de casos ocorridos (BCO) adequada a solução da falha. O Anexo 4 apresenta a árvore de decisões aplicável a falha de comunicação tratada nesse exemplo.

É importante ressaltar que a árvore de decisões além de sintetizar todo o processo descrito até o momento, também será utilizada para a geração das regras de produção utilizadas pelo sistema especialista responsável pelo diagnóstico e solução da falha.

5.2.3.5 Conversão da Árvore de Decisões em Regras de Produção

As regras de produção são retiradas diretamente da árvore de decisões, sendo o último passo antes da implementação computacional da solução, seguindo os modelos de regras tipo 1,2,3 e 4 descritos no capítulo 4. O Anexo 2 apresenta as regras elaboradas.

5.2.3.6 Determinação dos Estados Indicadores de Falha Resolvida

Uma vez que as ações corretivas adequadas sejam propostas, é necessário determinar que estado os indicadores luminosos (LEDs) do painel assumem quando a falha foi resolvida, de forma que o executor das ações possa saber quando essas lograram sucesso.

Os indicadores luminosos relevantes a falha de comunicação e seus estados são apresentados na figura 5.6.

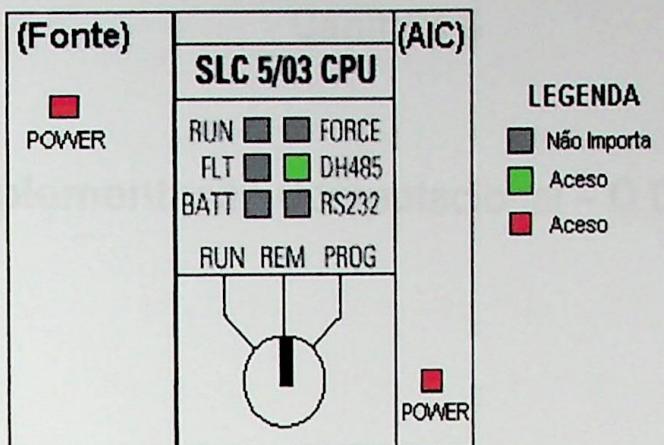


Figura 5.6 – Estado assumido pelos indicadores luminosos do painel do CLP indicando falha resolvida.

O Capítulo 6 descreve um sistema baseado em regras e casos que implementa de forma computacional a descrição proposta para o processo de diagnóstico, solução e aprendizado de falhas em CLP, utilizando o conhecimento referente a uma falha de comunicação tratado pela metodologia proposta.

Capítulo 6

A Implementação Computacional – O CLPSolve

Nesse capítulo é apresentada uma ferramenta computacional de apoio ao diagnóstico, solução e aprendizado de falhas em CLP, o CLPSolve, que implementa o exemplo de aplicação da metodologia proposta para aquisição, tratamento e estruturação do conhecimento descrito no capítulo 5.

Inicialmente, descreve-se os módulos componentes do CLPSolve e suas respectivas interfaces com o usuário. A seguir, um exemplo de utilização da ferramenta é apresentado.

6.1 A Implementação Computacional : O CLP Solve

A implementação computacional descrita a seguir utiliza os resultados obtidos a partir da aplicação da metodologia para aquisição, tratamento e estruturação do conhecimento relativo ao diagnóstico, solução e aprendizado de falhas em CLP, para a criação de uma ferramenta de apoio a manutenção utilizável por operadores de máquina. Nessa ferramenta, chamada CLPSolve, são implementadas as regras para diagnóstico da falha e definição de contexto/subcontexto/versão, os algoritmos de seleção de BAC, BCO e aprendizado de casos, descritos nos Capítulos 4 e 5.

Os detalhes da interface para entrada de dados e apresentação de resultados, assim como as telas de configuração da ferramenta e de edição de BAC e BCO são apresentadas, assim como o módulo de acompanhamento do desempenho da ferramenta.

6.1.1 Visão Geral do CLPSolve

O CLP Solve é a ferramenta computacional que implementa a descrição proposta para o processo de diagnóstico, solução e aprendizado de falhas através de um sistema baseado em regras e casos, utilizando informações manipuladas pela metodologia para aquisição, tratamento e estruturação do conhecimento proposta nessa dissertação. É composto pelos seguintes módulos:

- Módulo de Configuração do CLP.

- Módulo de Entrada dos Sintomas de Falha.
- Módulo de Apresentação do Diagnóstico e Ações Corretivas.
- Módulo de Edição de Bases (BAC e/ou BCO).
- Módulo de Acompanhamento de Desempenho.

A figura 6.1 apresenta o tela interface principal do CLPSolve.

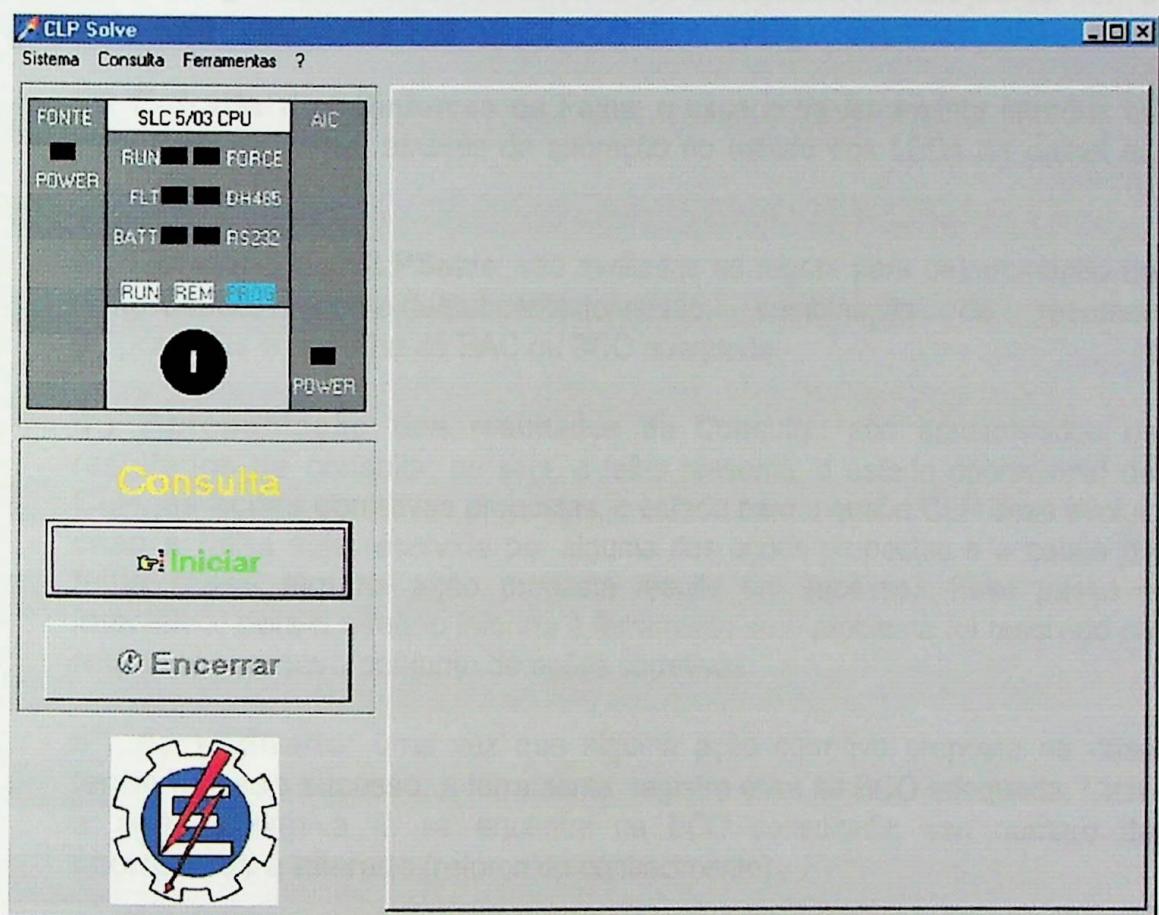


Figura 6.1 – Interface principal do CLPSolve.

Além dos módulos apresentados acima. O CLP Solve implementa:

- As regras para diagnóstico de falha (regras Tipo 1).
- As regras para determinação de contexto/subcontexto/versão (regras Tipo 2 e 3).
- As regras para avaliação das combinações de recursos disponíveis (regras Tipo 4).

- O Algoritmo de Seleção e Ordenação dos Casos Armazenados.
- O Algoritmo de Aprendizado e Reforço do Aprendizado dos conjuntos de ações - causa que lograram sucesso.
- O Algoritmo de Transição entre Conhecimento Geral (BAC) e Conhecimento Específico (BCO).

Os passos para a utilização do CLPSolve são:

1º) Configuração: consiste em definir as condições de instalação do CLP e recursos disponíveis para manutenção/substituição.

2º) Entrada dos Sintomas da Falha: o usuário da ferramenta introduz os sintomas da falha, através da alteração do estado dos LEDs do painel da UCP do CLP.

3º) Consulta ao CLPSolve: são avaliadas as regras para determinação do diagnóstico, contexto/subcontexto/versão, combinação de recursos disponíveis e escolha da BAC ou BCO adequada.

4º) Apresentação dos resultados da Consulta: são apresentados os resultados da consulta, ou seja, a falha presente, o estado operacional do CLP, as ações corretivas propostas, o estado para o qual o CLP deve evoluir caso a falha seja resolvida por alguma das ações propostas e a causa da falha (caso alguma ação proposta resulte em sucesso). Esse passo é interativo, pois o usuário informa à ferramenta se o problema foi resolvido ou requisita um novo conjunto de ações corretivas.

5º) Aprendizado: uma vez que alguma ação corretiva proposta no caso tenha logrado sucesso, a ferramenta registra essa na BCO adequada. Caso a ação corretiva já se encontre na BCO consultada, seu número de ocorrências é alterado (reforço do conhecimento).

7º) Encerramento da Consulta e Atualização do Desempenho da Ferramenta: a cada final de consulta, são atualizados os dados referentes ao número de vezes que o sistema foi consultado, número de ocorrências de determinada falha, número de sucessos e fracassos, gerais e por falha. Além disso todas as variáveis recebem valores padrão e a ferramenta fica em estado de espera pelo inicio de uma nova consulta.

6.1.2 Módulo de Configuração do CLP

No módulo de configuração são inseridas informações referentes ao contexto de operação do CLP e aos recursos (cabos, placas, etc...) disponíveis para testes e substituição. Esse consiste num banco de dados cujos campos estão associados a uma interface visual. As informações referentes as configurações locais do CLP e recursos disponíveis para manutenção são extraídas da análise das diversas falhas que a UCP em questão pode apresentar e das ações corretivas cabíveis, complementados com a experiência do especialista.

As regras responsáveis pela determinação dos contextos, subcontextos, versões, recursos disponíveis e escolha dos conjuntos de ações aplicáveis à solução da falha (regras Tipo 2,3,4) avaliam suas premissas através de consultas ao banco de dados de configuração (BDC).

As informações de configuração devem ser inseridas antes do começo do uso da ferramenta, numa etapa instalação, devendo ser mantidas atualizadas. Imprecisão nas informações fornecidas conduzem a decisões erradas. O CLPSolve implementa rotinas de teste das informações inseridas no BDC, de forma a impedir inconsistências nos dados inseridos. Um exemplo de inconsistência dessa natureza é informar ao sistema que a ferramenta de programação não está disponível e que existe um módulo PIC disponível, pois a disponibilidade do módulo PIC sem a ferramenta de programação é absolutamente inócuia.

A figura 6.2 apresenta a interface visual para entrada de dados no banco de dados de configuração (BDC) do CLPSolve. A figura 6.3 apresenta o detalhe das informações de configuração relevantes à falha de comunicação cuja solução serve de exemplo para aplicação da metodologia proposta nessa dissertação.

6.1.3 Módulo de Entrada dos Sintomas de Falha

A inserção dos sintomas da falha (estado assumido pelos LEDs do painel frontal da fonte, UCP e módulo AIC do CLP) é realizada através da alteração do estado de indicadores visuais (LEDs) colocados no módulo de entrada de sintomas, indicadores esses equivalentes aos presentes no painel da UCP.

Uma vez que o estado de um LED presente no módulo de entrada dos sintomas é alterado, tal alteração manifesta-se na interface visual, sendo que os estados assumido pelos LEDs serão avaliados pelas premissas das regras que realizam o diagnóstico de falha (regras Tipo 1) quando uma consulta ao sistema for iniciada.

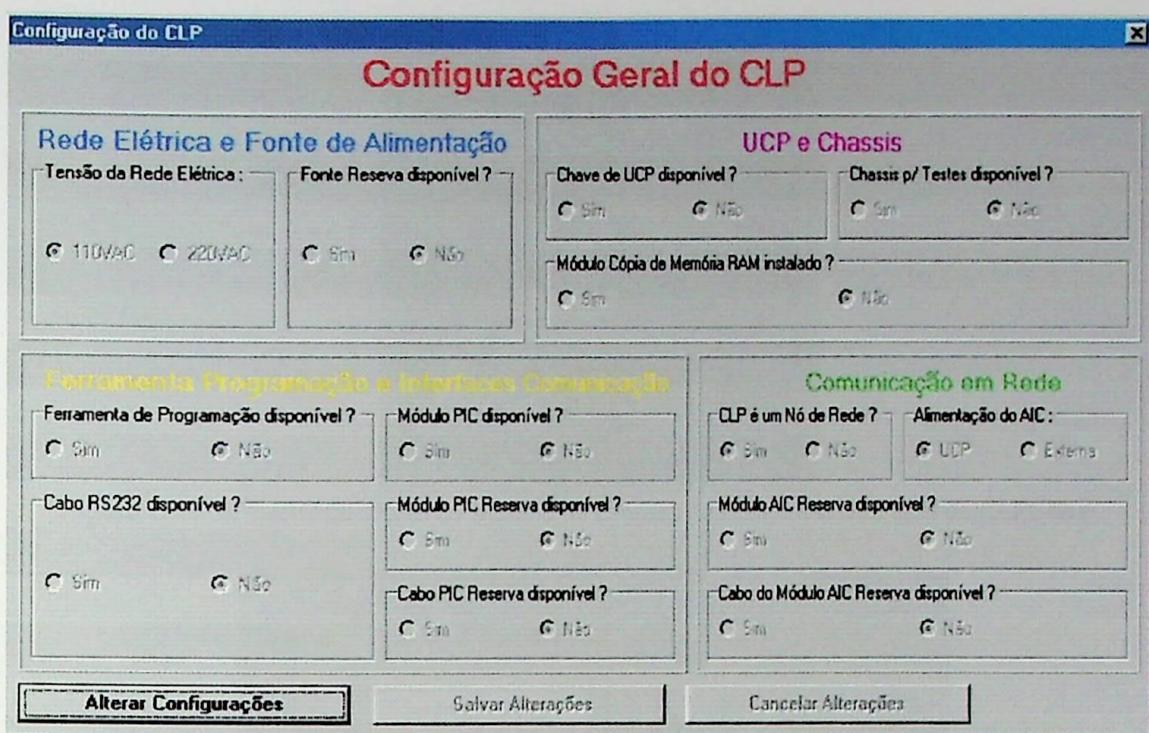


Figura 6.2 - A interface visual do Módulo de Configuração do CLPSolve.

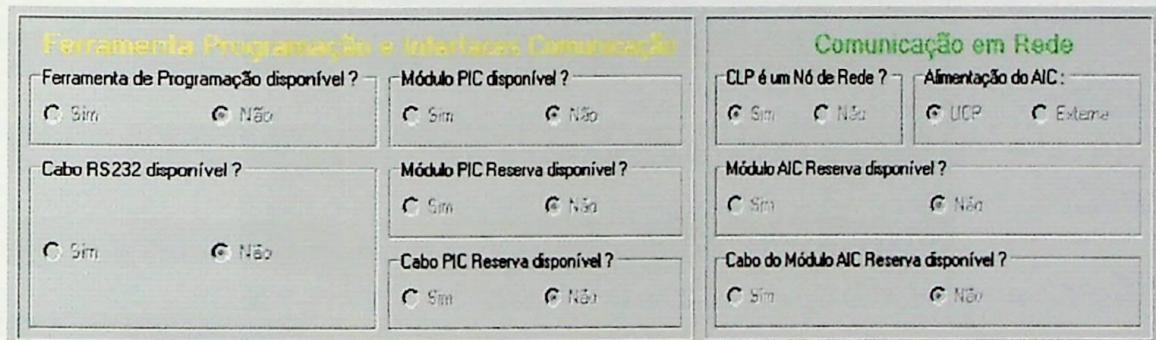


Figura 6.3 - Detalhe da interface visual do Módulo de Configuração com as informações relevantes à falha de comunicação no CN 1.

A figura 6.4 apresenta a interface visual para entrada dos sintomas da falha. A figura 6.5 apresenta um exemplo da interface para alteração de estados assumidos pela interface visual de entrada de Sintomas de Falha.

Novamente, o CLPSolve implementa rotinas que impedem a inconsistência na entrada de sintomas. Nesse caso, um exemplo de inconsistência seria informar ao módulo de entrada que o LED de "RUN" esta aceso (UCP executando programa), enquanto o LED de "POWER" da fonte de alimentação esta apagado (fonte de alimentação desenergizada ou em falha), estado que nunca ocorre em situações práticas.

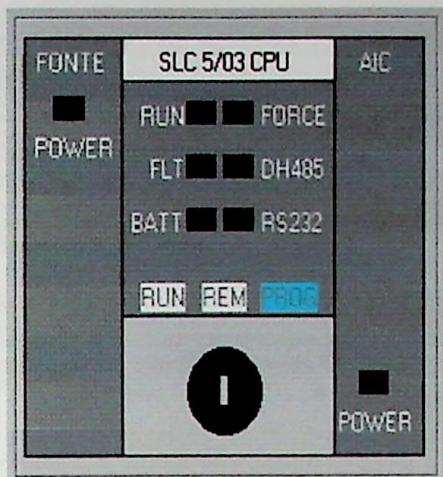


Figura 6.4 - A interface visual do Módulo de Entrada dos Sintomas de Falha do CLPSolve.

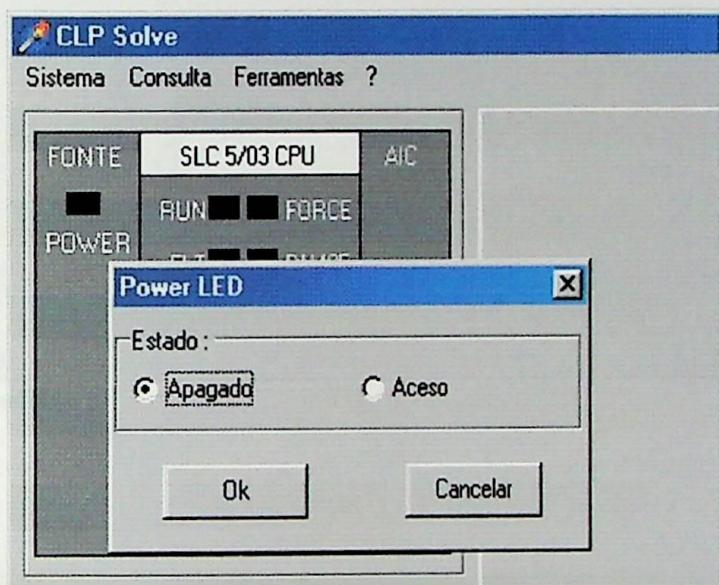


Figura 6.5 - Exemplo da interface para alteração de estados assumidos pela interface visual de entrada de Sintomas da Falha.

6.1.4 Módulo de Apresentação do Diagnóstico, Ações Corretivas e Causas da Falha

Uma vez iniciada uma consulta, diagnosticada uma falha, avaliados os recursos disponíveis para sua solução, o módulo de apresentação do diagnóstico, ações corretivas e causas é apresentado ao usuário. Esse contém os campos apresentados na figura 6.6.

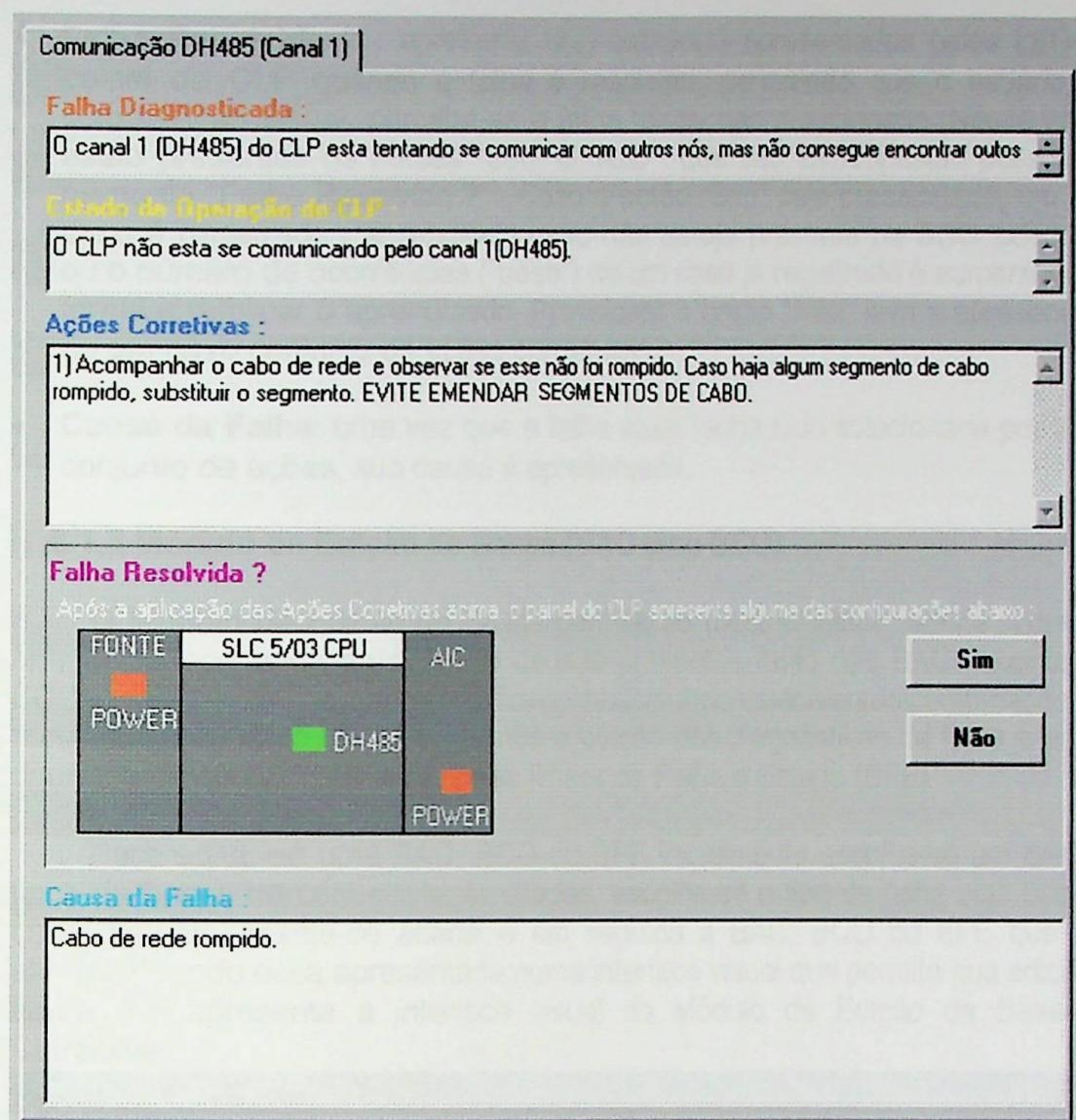


Figura 6.6 – A interface visual do Módulo de Apresentação do Diagnóstico, Ações Corretivas e Causas do CLPSolve.

- **Falha Diagnosticada:** apresenta a falha diagnosticada a partir da análise do estado dos LEDs da UCP, introduzidos pelo usuário.
- **Estado de Operação do CLP:** apresenta o estado de operação determinado pela falha diagnosticada.
- **Ações Corretivas:** apresenta os conjuntos de ações corretivas aplicável ao contexto/subcontexto/versão/ recursos disponíveis. Essas, aplicadas pelo usuário, podem gerar dois resultados: a solução da falha, o que gera um novo caso ou o aumento do número de ocorrências ("peso") de um caso já registrado, ou a permanência da falha, o que gera a apresentação de um novo conjunto de ações.

- **Falha Resolvida ?** : apresenta o(s) estado(s) apresentados pelos LEDs do painel do CLP quando a falha é resolvida, permitindo que o usuário, por comparação visual, perceba se a falha foi ou não solucionada. Nesse campo estão presentes os botões “Sim” e “Não”, que são as respostas possíveis a pergunta “Falha Resolvida ?”. Caso o botão “Sim” seja pressionado, um novo caso é registrado (“aprendido”) caso não esteja presente na BCO adequada, ou o número de ocorrências (“peso”) de um caso já registrado é aumentado, de forma a reforçar o aprendizado. Pressionar o botão “Não” leva a apresentação de um novo conjunto de ações corretivas.
- **Causa da Falha:** uma vez que a falha atual tenha sido solucionada por algum conjunto de ações, sua causa é apresentada.

6.1.5 Módulo de Edição de Bases (BAC e/ou BCO)

O Módulo de Edição de Bases permite ao usuário inserir, alterar, corrigir e apagar os conjuntos de ações e causas presentes tanto nas BACs quanto nas BCOs aplicáveis a cada falha/contexto/subcontexto/versão/combinação de recursos. O módulo também permite a edição dos diagnósticos de falha e estado operacional do CLP, presentes nas Bases de Falha e Estado (BFE).

Para editar-se uma BAC, BCO ou BFE inicialmente escolhe-se um dos três tipos de bases de conhecimento citadas, escolhe-se o tipo de falha cuja base de conhecimento deseja-se alterar e em seguida a BAC, BCO ou BFE que será alterada, sendo essa apresentada numa interface visual que permite sua edição. A figura 6.7 apresenta a interface visual do Módulo de Edição de Bases do CLPSolve.

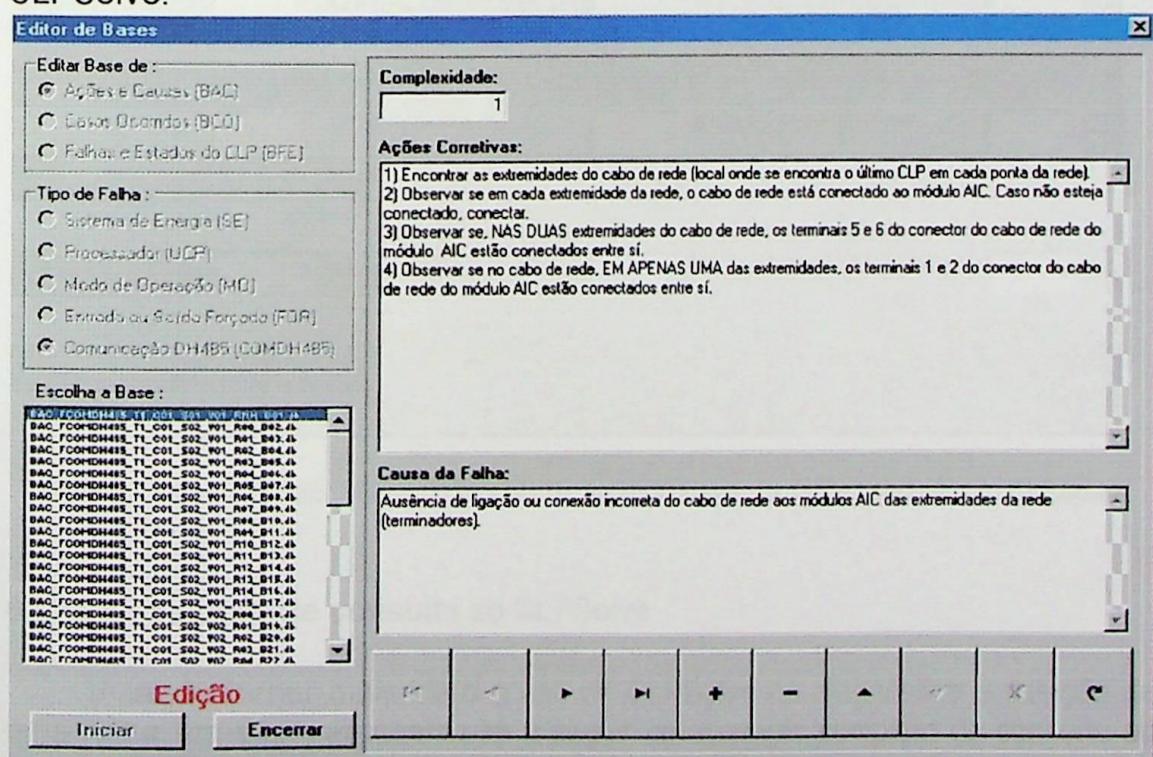


Figura 6.7 – Interface visual do Módulo de Edição de Bases do CLPSolve.

6.1.6 Módulo de Acompanhamento de Desempenho

O Módulo de Acompanhamento de Desempenho do permite a avaliação da eficácia dos conjuntos de ações presentes nas BACs e BCOs presentes no CLPSolve. Nesse módulo temos um indicador geral do número de vezes que o CLPSolve foi consultado, o número de vezes que a consulta encerrou-se com a falha solucionada e o número de vezes que a falha não foi resolvida.

Além do acompanhamento geral do desempenho do sistema, há também um acompanhamento por tipo de falha, no qual registra-se o número de ocorrências de um determinado tipo de falha, o número de vezes que a ocorrência foi resolvida e o número de vezes que essa não foi resolvida. A figura 6.8 apresenta a interface visual do Módulo de Acompanhamento de Desempenho.

A interface visual do Módulo de Acompanhamento de Desempenho do CLPSolve é uma janela com o seguinte layout:

- Título:** CLPSolve - Desempenho
- Seção Geral:** Exibe os seguintes valores:
 - Nº de Consultas :
 - Nº Falhas Resolvidas :
 - Nº Falhas Não Resolvidas :
- Seção Por Tipo de Falha:** Subseções para diferentes tipos de falhas:
 - Sistema de Energia:** Nº Ocorrências : Nº Ocorrências Resolvidas : Nº Ocorrências não Resolvidas :
 - UCP:** Nº Ocorrências : Nº Ocorrências Resolvidas : Nº Ocorrências não Resolvidas :
 - Estado Operacional:** Nº Ocorrências : Nº Ocorrências Resolvidas : Nº Ocorrências não Resolvidas :
 - Comunicação DH485 (Canal 1):** Nº Ocorrências : Nº Ocorrências Resolvidas : Nº Ocorrências não Resolvidas :
- Botão Zerar:** Botão para zerar os valores exibidos.

Figura 6.8 – A interface visual do Módulo de Acompanhamento de Desempenho do CLPSolve.

6.2 Um exemplo de consulta ao CLPSolve

Visando tornar mais claro o uso do CLPSolve no diagnóstico e solução de uma falha em CLP, apresenta-se a seguir um exemplo completo de consulta ao

mesmo, considerando a ocorrência da falha de comunicação no canal 1, na qual a metodologia para aquisição/tratamento/estruturação do conhecimento foi aplicada.

6.2.1 A Configuração

Os recursos aplicáveis à solução da falha são determinados nos agrupamentos “Ferramenta de Programação e Interfaces de Comunicação” e “Comunicação em Rede”, ambos presentes na tela de “Configuração do CLP”, do CLPSolve.

No agrupamento “**Ferramenta de Programação e Interfaces de Comunicação**” os itens configurados são apresentados na tabela 6.1.

Item	Valor
Ferramenta de Programação disponível ?	Sim
Módulo PIC disponível ?	Sim
Módulo PIC reserva disponível ?	Não
Cabo PIC reserva disponível ?	Não
Cabo RS232 disponível ?	Não

Tabela 6.1 – Configurações relativas ao agrupamento “Ferramenta de Programação e Interfaces de Comunicação” do módulo de Configuração do CLP.

No agrupamento “**Comunicação em Rede**” os itens configurados são apresentados na tabela 6.2.

Item	Valor
CLP é Nó de Rede ?	Sim
Alimentação do AIC ?	UCP
Módulo AIC Reserva disponível ?	Não
Cabo do Módulo AIC Reserva disponível ?	Não

Tabela 6.2 – Configurações relativas ao agrupamento “Comunicação em Rede” do módulo de Configuração do CLP.

6.2.2 Entrada dos Sintomas

Os sintomas relevantes ao diagnóstico da falha de comunicação no canal 1 são os estados assumidos pelos LEDs “POWER”, “DH485” e “POWER AIC”, sendo esses apresentados na tabela 6.3.

LED	Estado
POWER :	ACESO
DH485 :	PISCANDO
POWER AIC:	ACESO

Tabela 6.3 – Estado assumido pelos LEDs relevantes ao diagnóstico de falha de comunicação no canal 1.

6.2.3 Consulta ao CLPSolve

Uma vez que os estados dos LEDs indicam um estado de falha de comunicação no canal 1 (DH485) e o item de configuração “CLP é Nό de Rede ?” recebeu o valor “Sim”, o CLPSolve questiona o usuário se “Outros nós perderam comunicação em rede ?”, sendo que no caso desse exemplo, a resposta é “Não”. A figura 6.9 apresenta o estado final da interface visual após a alteração dos estados dos LEDs relevantes ao diagnóstico da falha e a pergunta feita ao usuário durante a consulta ao CLPSolve.

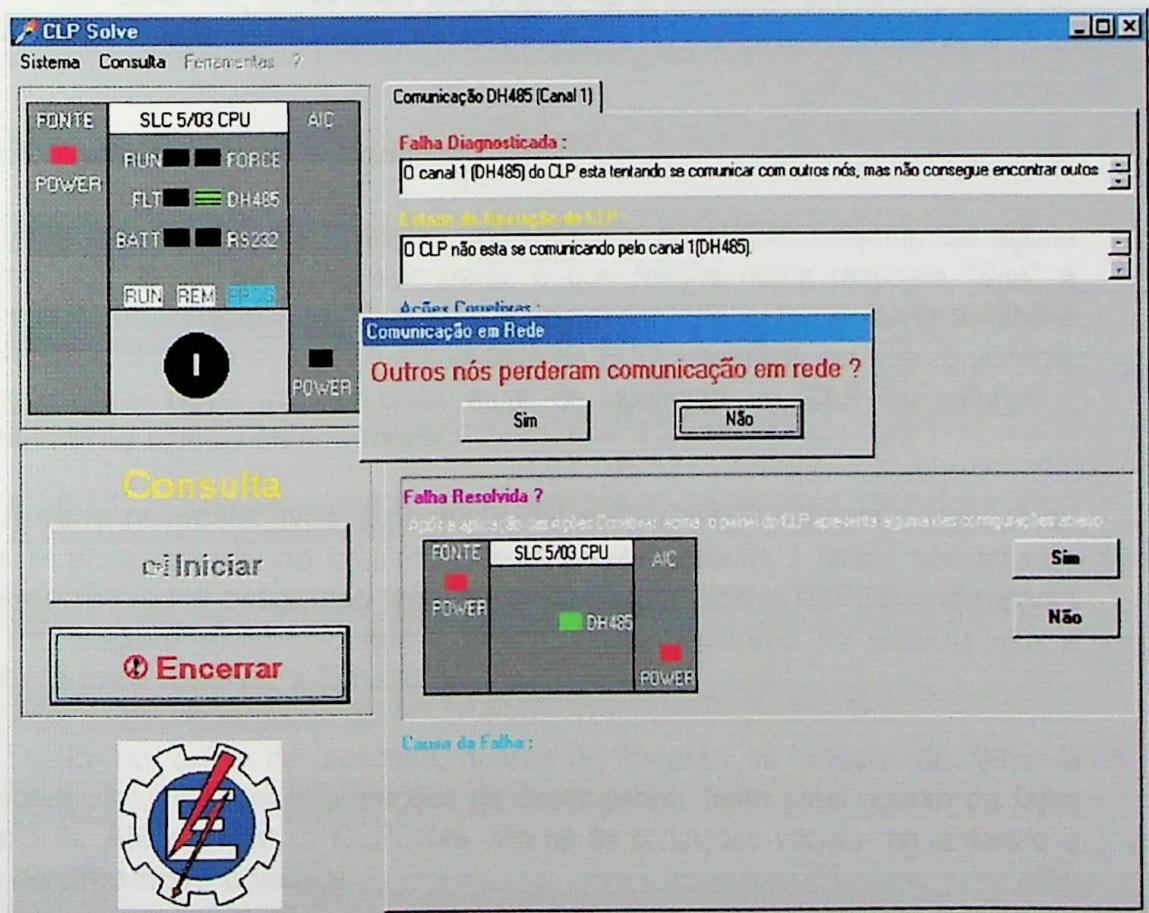


Figura 6.9 – Estado final da interface visual após a alteração dos estados dos LEDs relevantes ao diagnóstico da falha e a pergunta feita ao usuário durante a consulta ao CLPSolve.

É importante notar que no caso da falha de comunicação, não há como determinar a priori se outros nós da rede perderam ou não comunicação, sendo necessário, durante a consulta, inserir essa informação, pois a mesma é fundamental na determinação do subcontexto correto no qual ocorre a falha de comunicação.

Uma vez respondida a pergunta, o CLPSolve apresenta os itens “Falha Diagnosticada:”, “Estado de Operação do CLP:”, presentes na BFE adequada a falha atual.

A seguir, o CLPSolve realiza a escolha da BAC ou BCO, seguindo os algoritmos estabelecidos no Cap. 4. No caso dessa consulta, a BAC ou BCO número 42 é a escolhida, pois é a adequada aos recursos disponíveis. As “Ações Corretivas:” e a “Causa da Falha:” são apresentadas apenas na ocasião oportuna, dependendo da consulta a BAC ou BCO e dos resultados obtidos com a aplicação das ações corretivas propostas.

O Anexo 5 apresenta a busca realizada na árvore de decisões até a escolha da BAC ou BCO 42, lembrando que a BAC 42 é escolhida apenas no caso da BCO 42 estar vazia (sem casos armazenados).

6.2.4 Encerramento da Consulta

Uma vez que a falha tenha sido resolvida por algum conjunto de ações presente na BAC ou BCO adequada, o que resulta numa resposta “Sim” à pergunta “Falha Resolvida ?”, o conjunto de ações que logrou resolver a falha é armazenado ou tem seu peso aumentado na BCO adequada, ou seja, o sistema aprende algo novo no contexto atual de operação do CLP ou reforça o conhecimento obtido anteriormente.

A cada resposta “Não” à pergunta “Falha Resolvida ?”, um novo conjunto de ações é apresentado ao usuário. Caso tenha-se aplicado o último conjunto de ações corretivas e essas não tenham solucionado a falha, o CLPSolve apresenta uma mensagem indicando que o conhecimento disponível no sistema não é suficiente para resolver a falha atual.

Tanto no caso de sucesso, quanto de fracasso na solução da falha, o CLPSolve atualiza as informações de desempenho, tanto geral quanto da falha específica. Além disso, o CLPSolve retorna às condições iniciais, aguardando o inicio de uma nova consulta.

O Capítulo 7 apresenta as conclusões e propostas de desenvolvimentos futuros e complementares para o assunto abordado nessa dissertação.

Capítulo 7

Conclusão

Ferramentas de apoio ao diagnóstico, solução e aprendizado de falhas em equipamentos industriais podem ser extremamente úteis, principalmente em contextos de escassez e/ou baixa qualificação dos recursos humanos disponíveis para tal função. Quando o problema é particularizado para equipamentos de alta complexidade e que desempenham papel fundamental no controle e segurança de processos industriais, como o CLP, a disponibilidade de tais ferramentas torna-se fator decisivo na seleção do fornecedor do CLP, pois preenchem um lacuna normalmente ignorada: a manutenção eficiente e a retenção do conhecimento acumulado.

A metodologia proposta para aquisição, tratamento e estruturação do conhecimento referente ao diagnóstico e solução de falhas em CLP mostrou-se eficiente, principalmente estabelecer um procedimento sistemático, com etapas encadeadas bem definidas, de tal forma que, ao final de sua aplicação, tem-se o conhecimento adequado à uma implementação computacional. Mesmo nos casos de falhas de grande complexidade, como a de comunicação utilizada como exemplo, a aplicação da metodologia mostrou-se perfeitamente viável.

O uso combinado de sistemas baseados em regras e casos permite a superar limitações inerentes a cada um deles. No caso dos sistema baseados em regras, supera-se a impossibilidade do aprendizado automático de novos conhecimentos ao longo do uso do sistema. No caso dos sistemas baseados em casos, permite que mesmo na ausência de casos registrados, regras gerais sejam aplicadas e, pelo sucesso da aplicação das ações sugeridas pelas mesmas, casos de sucesso sejam armazenados de forma incremental. A síntese de um formato padrão para as regras e casos introduziu um formalismo útil para implementações computacionais.

A implementação computacional da metodologia proposta nesse trabalho mostrou a necessidade do desenvolvimento de algoritmos eficientes para implementação de aspectos relevantes na implementação da descrição proposta para o processo de diagnóstico, solução e aprendizado de falhas, principalmente no tocante ao aprendizado de novos conhecimentos, reforço do acervo disponível e ordenação dos casos apresentados ao usuário considerando os critérios definidos.

A implementação do CLPSolve apresentou sugestões tanto para a interface com usuário quanto para módulos que possam compor ferramentas computacionais com a mesma finalidade.

É importante salientar que o CLP foi testado em condições de laboratório, nas quais apresentou bom desempenho, porém tal desempenho necessita de comprovação em condições reais de operação em ambiente industrial.

Outro ponto importante é que o conhecimento armazenado na formas de casos para uma determinada falha num local de operação, pode não ser válido ou não apresentar a mesma incidência em outro. Tal fato pode ocorrer devido a diferenças relacionadas a aspectos ambientais e/ou qualidade das instalações elétricas dos locais de operação dos sistemas automatizados.

Desenvolvimentos futuros, complementares ou mais aprofundados poderiam investigar:

- formas de síntese da similaridade entre UCPs de diversas famílias de CLP de um mesmo fabricante, buscando utilizar o conhecimento relativo ao diagnóstico e solução de falha de um determinado modelo de UCP em outro modelo;
- formas de síntese de similaridade entre ações corretivas aplicadas a partir de recursos parcialmente disponíveis, ou seja, o reaproveitamento parcial de soluções;
- a possibilidade da inclusão de um planejador de ações, de forma que uma vez determinada a falha, o contexto, subcontexto e a versão, o próprio sistema seria capaz de montar ações corretivas e sugerir-las, sem que as mesmas fossem previamente definidas;
- formas de integração, via porta de comunicação, entre o CLPSolve e a memória da UCP do CLP, de forma a possibilitar a detecção de falhas em tempo real, principalmente das falhas relativas ao funcionamento da UCP e, num segundo momento, falhas no próprio processo controlado pelo CLP.

Referências Bibliográficas

- [01] Alves S.A., Gebrael A.G.F., Penteado A.C.P.O – "Seleção de Métodos e Técnicas de Treinamento". São Paulo, Universidade Mackenzie – Faculdade de Tecnologia, 1998.
- [02] Ballakrishnan K., Honavar V. – "Intelligent Diagnostic Systems". Ames, Iowa State University, Departament of Computer Science, 1998.
- [03] Wilson J.N. – "Using a Knowledge – Based System to Guide Shop Floor Assembly". Glasgow, University of Strathclyde – Department of Computer Science, 1988.
- [04] Kamel M.N., McCaffrey M.J., Metzler P.G. – "Analisis, Desing, Implmentation and Deployment of a Prototype Maintenance Advisor Expert System for MK92 Fire Control System". Elsevier Science Ltd, Expert Systems with Applications, Vol.10 No 2, pp. 193-207, 1996. .
- [05] Aamodt A., Plaza E. – "Case-Based Reasoning: Foundational, Issues, Methodological Variations, and System Approaches". IOS Press, AI Communications, Vol. 7: 1, pp. 39-59, 1994.
- [06] Ross B.H. – "Some psychological results on case-based reasoning". Morgan Kaufmann, Case-Based Reasoning Workshop. Pensacola Beach, pp.144-147, 1989.
- [07] Kolodner J. – "Maintaining organization in a dynamic long-term memory". Cognitive Science, Vol.7, pp.243-280, 1983.
- [08] Kolodner J. – "Reconstructive memory, a computer model". Cognitive Science, Vol.7, pp.281-328, 1983.
- [09] Nascimento C., Yoneyama T. – "Inteligência Artificial em Controle e Automação" – Edgard Blücher, São Paulo - 2000.
- [10] Barreto J. M. – "Inteligência Artificial no Limiar do Século XXI". ppp edições, Florianópolis, 1999.
- [11] Price C. – "Computer - Based Diagnostic System" – Springer Verlag, Londres, 1999.
- [12] Medsker L. – "Hybrid Intelligent Systems" – Kluwer Academic Plubishers, Boston, 1995.

- [13] "SLC 500 Modular Hardware Style, Installation and Operation Manual" – Rockwell Automation -Allen Bradley, Milwaukee, 1998.
- [14] Villanueva H., Lamba H. – "Operator Guidance System for Industrial Plant Supervision". Elsevier Science Ltd, Expert Systems with Applications, Vol.12 No 4, pp. 441-454, 1997.
- [15] Pereira S.L., Andrade A.A. – "Linking Supervisory Systems to Expert Systems for Improvement in Industrial Automation Process". São Paulo, Universidade de São Paulo – Escola Politécnica - Departamento de Energia e Automação Elétricas, 2002.
- [16] Teng S.H., Ho S.Y. – "Failure mode and effects analysis, an integrated approach for product design and process control". MCB University Press, International Journal of Quality & Reliability Management, Vol. 13 No. 5, pp. 8-26, 1996.
- [17] Golding A.R., Rosenbloom P.S. – "Improving accuracy by combining rule-based and case-based reasoning". Cambridge, Mitsubishi Electric Research Labs, 1995.
- [18] Laboratório de Inteligência Artificial – "Expert SINTA – Manual de Operação". Fortaleza, Universidade Federal do Ceará, 1996.
- [19] Russel S., Norvig P. – "Artificial Intelligence, A Modern Approach" – Prentice Hall, Englewood Cliffs , 1995.
- [20] Hopgood A. – "Knowledge based Systems for Engineers and Scientists". CRC Press, Boca Raton, 2000.
- [21] Martin J., Oxman S. – "Building Expert Systems, a Tutorial" – Prentice Hall, Englewood Cliffs, 1988.
- [22] Holtz F. – "Sistemas Especialistas, Programando em Turbo C" – Campus, Rio de Janeiro - 1991.
- [23] Cunha H., Ribeiro S. – "Introdução aos Sistemas Especialistas" – Livros Técnicos e Científicos, Rio de Janeiro, 1987.
- [24] Bryan L.A., Bryan E. A. – "Programmable Controllers, Theory and Implementation" – Industrial Text, Atlanta, 1997.
- [25] Cantú M. – "Dominando o Delphi 3" – Makron Books, São Paulo, 1998.
- [26] Osier D. – "Aprenda em 21 Dias Delphi 2" – Campus, Rio de Janeiro, 1997.

Anexo 1

Algoritmos do Capítulo 4 representados em Pseudocódigo

1. Algoritmo de Seleção e Ordenação dos Casos Armazenados (Referente ao Item 4.4.1)
2. Algoritmo de Aprendizado e Reforço do Aprendizado (Referente ao Item 4.4.2)
3. Algoritmo de Transição entre Conhecimento Geral (BAC) e Conhecimento Específico (BCO) (Referente ao Item 4.4.3)

1. Algoritmo de Seleção e Ordenação dos Casos Armazenados (Referente ao Item 4.4.1)

```
// Início do algoritmo para BCO não vazia. //  
SE (BCO = não vazia) ENTÃO  
INICIO  
  
BCO = ativa; // Contém os casos armazenados em consultas anteriores. //  
BCOO = vazia; // Vazia. //  
  
// Rotina de preenchimento e ordenação da BCOO com os dados da BCO. //  
  
caso_com_maior_número_de_ocorrências_na_BCO = 1; //  
Inicialização de variável que armazena o caso com maior número de ocorrências.//  
número_de_ocorrências_do_caso_atual_na_BCO = 1; //  
Inicialização da variável que armazena o número de ocorrências do caso atual em análise na BCO. //  
número_de_casos_armazenados_na_BCO = CONTA_NÚMERO_CASOS(BCO); // Inicialização da variável que contém o número de casos armazenados na BCO sob consulta, contados pela função CONTA_NUMERO_DE_CASOS. //  
Ir para primeiro caso da BCO; // Vai para primeiro caso da BCO.  
//  
Ir para primeiro caso da BCOO; // Vai para primeiro caso da BCOO, que ainda não contém informações.//  
  
//A rotina a seguir realiza uma busca em toda BCO, analisando o número de ocorrências de cada caso armazenado. Caso o número de ocorrências do caso atual seja maior que o do caso previamente analisado, esse torna-se o maior número de ocorrências de um caso armazenado, caso contrário, mantém-se o previamente armazenado e passa-se à análise do próximo caso. Tal procedimento repete-se até que todos os casos armazenados tenham sido analisados, ou seja, até o fim da BCO. Ao término da execução do laço de repetição, o caso com maior número de ocorrências, contido na BCO, será conhecido. //
```


REPETE

número_de_ocorrências_do_caso_atual_na_BCO:= Nº de Ocorrências;
// Atribui à variável o número de ocorrências do caso atual em análise na BCO //

SE(número_de_ocorrências_do_caso_atual_na_BCO>caso_com_maior_número_de_ocorrências_na_BCO) **ENTÃO**
INICIO

caso_com_maior_número_de_ocorrências_na_BCO = Nº de Ocorrências; // Atribui à variável o número de ocorrências do caso atual em análise na BCO //

FIM

Ir para próximo caso da BCO; // Vai para próximo caso da BCO //

ATÉ (caso_atual_da_BCO = último caso da BCO); // Encerra repetição ao atingir o final da BCO. //

Ir para primeiro caso da BCO; // Vai para primeiro caso da BCO //

// A rotina a seguir realiza a cópia dos casos com maior número de ocorrências da BCO para a BCOO. Para tal, compara-se o número de ocorrências do caso atual da BCO com o maior número de ocorrências de um caso presente na BCO (levantado na rotina anterior). Se o caso atual tiver seu número de ocorrências igual ao maior número de ocorrências, esse é copiado para o BCOO, caso contrário, parte-se para o próximo caso da BCO. Quando o último caso da BCO é analisado, o valor da variável que contém o maior número de ocorrências de um caso é decrementado em uma unidade, repetindo-se o processo até que o número de casos contido na BCOO seja igual ao número de casos contidos na BCO, porém ordenados pelo número de ocorrência. Cabe notar que como a BCO armazena os casos ordenados por complexidade (da menor para a maior), ao realizar a cópia de um caso da BCO para a BCOO, o caso copiado já é o de menor complexidade, pois aparece primeiro que um de maior complexidade, mesmo que ambos tenham o mesmo número de ocorrências. //

REPETE

// Se o número de ocorrências do caso atual da BCO é igual ao maior número de ocorrências de um caso, os dados desse caso são copiados da BCO para a BCOO.//

SE (Nº de Ocorrências = caso_com_maior_número_de_ocorrências_na_BCO) **ENTÃO**
INICIO

Copia dados do caso atual para BCOO; // Realiza cópia do caso da BCO para a BCOO. //

FIM

Ir para próximo caso da BCO; // Vai para próximo caso da BCO. //

// Se a busca na BCO chegou ao último caso e o maior número de ocorrências de um caso é maior que uma (01) ocorrência, decrementa-se o valor do caso com maior número de ocorrências. O objetivo desse procedimento é encontrar-se o caso com o segundo, terceiro, quarto... maior número de ocorrências.//

SE (caso_atual = último caso da BCO) ENTÃO
INICIO

SE (caso_com_maior_número_de_ocorrências_na_BCO > 1) ENTÃO
INICIO

caso_com_maior_número_de_ocorrências_na_BCO=caso_com_maior_número_de_ocorrências_na_BCO - 1; //Decrementa o maior número de ocorrências de um caso, de forma a permitir a busca por casos de ocorrência inferior a maior. //

Ir para primeiro caso da BCO; // Vai para o inicio da BCO. //

FIM

FIM

número_de_casos_armazenados_na_BCOO=CONTA_NÚMERO_CASOS(BCOO); // Conta o número de casos presente na BCOO. //

ATÉ (número_de_casos_armazenados_na_BCOO = número_de_casos_armazenados_na_BCO); // Abandona o laço quando o número de casos presentes na BCOO é igual ao presente na BCO, nesse ponto ambas as bases de casos tem os mesmo casos, porém a BCOO contém os casos seguindo a ordenação por maior número de ocorrências e menor complexidade. //

FIM

// Fim do algoritmo para BCO não vazia. //

// Inicio do algoritmo para BCO vazia. Consulta a BAC. //

SE (BCO = vazia) ENTÃO

INICIO

```
BCO = inativa; // Desativa BCO determinada. //
BAC = ativa; // Ativa BAC determinada, a consulta será
realizada utilizando ações e causas contidas nessa.
Quando a BAC é ativada, o sistema apresenta ao usuário o
primeiro conjunto de ações presente nessa, sendo que
esses são ordenados por complexidade. //
```

FIM

```
// Fim do algoritmo para BCO vazia. Consulta a BAC. //
```

2. O Algoritmo de Aprendizado e Reforço do Aprendizado (Item 4.4.2)

```
// Inicio do Algoritmo de aprendizado. //
```

```
SE (falha_resolvida = sim) ENTÃO
```

INICIO

```
// Inicio da rotina de aprendizado quando a consulta é
realizada a BCOO (BCO). //
```

```
SE (BCOO = ativo) ENTÃO
```

INICIO

- *Busca na BCO adequada um caso cuja Complexidade seja igual a complexidade do caso atual da BCOO;*
- *Incrementa o N° de Ocorrências do caso presente na BCO adequada;*

```
FIM
```

```
// Fim da rotina de aprendizado quando a consulta é
realizada a BCOO (BCO). //
```

```
// Inicio da rotina de aprendizado quando a consulta é
realizada a BAC. //
```

```
SE (BAC = ativo) ENTÃO
```

INICIO

- *Abre novo caso na BCO adequada;*
- *Executa a cópia da Complexidade, Conjunto de Ações e Causa para a BCO adequada;*
- *Inicializa o N° de Ocorrências do caso copiado com a unidade (01);*

```
FIM
```

```
// Fim da rotina de aprendizado quando a consulta é
realizada a BAC. //
```

FIM

```
// Fim do Algoritmo de aprendizado. //
```

3. Algoritmo de Transição entre Conhecimento Geral (BAC) e Conhecimento Específico (BCO) (Item 4.4.3)

// Inicio do algoritmo de mudança de consulta ao conhecimento específico (BCO,BCOO) para consulta ao conhecimento geral (BAC). //

SE (*falha_resolvida* = *não*) **ENTÃO**
INICIO

//Inicio da rotina para consulta sendo realizada a BCOO.//

SE (*BCOO* = *ativo*) **ENTÃO**
INICIO

//Se o caso consultado na BCOO não é o último caso presente na BCOO, então executa a cópia da Complexidade do caso atual para BCC e avança para próximo caso da BCOO. //

SE (*caso_consultado* ≠ *último caso armazenado*) **ENTÃO**
INICIO

- *Copiar complexidade do caso atual para BCC;*
- *Apresentar próximo caso da BCOO;*

FIM

//Se o caso consultado na BCOO é o último caso presente na BCOO, então executa a cópia da Complexidade do caso atual para BCC e inicia consulta a BAC adequada. //

SE (*caso_consultado_BCOO* = *último caso armazenado na BCOO*)
ENTÃO
INICIO

- *Copiar complexidade do caso atual para BCC;*
- *Contar o número de casos armazenados na BCOO;*
- *BCOO = inativa;*
- *Abrir BAC adequada;*
- *Contar o número de conjuntos de ações - causa na BAC;*

// Caso o conteúdo da BCOO não seja igual ao da BAC adequada, fato detectado pela comparação entre o número de casos presentes dentro de cada uma, o sistema começa a consultar a BAC adequada. //

SE(*número_de_casos_armazenados_na_BCOO*≠*número_de_conjutos_de_ações-causa_da_BAC*) **ENTÃO**
INICIO

// Laço de comparação entre a Complexidade do conjunto ações - causa atual da BAC com as Complexidades dos casos presentes na BCC, provenientes da consulta prévia realizada a BCOO (BCO). O objetivo da busca é evitar que um conjunto de ações - causa apresentado na consulta a BCOO (e que não solucionou a falha) seja reapresentado durante a consulta a BAC. O laço encerra-se apenas quando a primeira complexidade de um conjunto ações - causa presente na BAC não é encontrada na BCC. //

REPETE

- *Buscar complexidade do caso atual da BAC na BCC;*

// Caso a complexidade do conjunto ações - causa atual esteja presente na BCC, avança para próximo conjunto. //

SE (*complexidade_caso_atual_da_BAC = presente na BCC*)

ENTÃO

INICIO

- *Avançar para próximo conjunto ações - causa da BAC;*

FIM

ATÉ (*complexidade_caso_atual_da_BAC = não presente na BCC*);

// Fim do Laço. //

FIM

// Caso o número de casos presentes na BCOO seja igual ao número de casos presentes na BAC adequada, todas as ações corretivas e causas possíveis para o contexto atual já foram aprendidas pelo sistema, ou seja , todo conhecimento geral já foi utilizado para resolução de falhas no local, assim, não existem mais conjuntos de ações - causa a serem apresentados, logo os sistema informa ao usuário que o conhecimento disponível não é suficiente para resolver a falha presente. //

SE(*número_de_casos_armazenados_na_BCOO=número_de_conjutos_de_ações-causa_da_BAC*) **ENTÃO**

INICIO

- Informar ao usuário que não é possível resolver a falha com o conhecimento disponível no sistema;
- BAC = inativo;

FIM

FIM

//Fim da rotina para consulta sendo realizada a BAC.//

//Inicio da rotina para consulta sendo realizada a BAC, sem antes ter sido realizada na BCO (BCOO), ou seja, ainda não existem casos armazenados.//

SE ((BAC = ativo) E (BCOO= inativo) E (BCC = inativo)) ENTÃO
INICIO

// Caso o conjunto atual de ações - causa não seja o último, ou seja, todos os conjuntos de ações - causa presentes na BAC ainda não foram apresentados, apresenta o próximo conjunto de ações - causa da BAC.
//

SE(conjunto_ações-causa_consultado ≠ último conjunto ações-causa armazenado) ENTÃO
INICIO

- Apresentar o próximo conjunto ações-causa da BAC;

FIM

// Caso o conjunto atual de ações - causa seja o último, ou seja, todos os conjuntos de ações - causa presentes na BAC foram apresentados, o sistema informa ao usuário que o conhecimento disponível não é suficiente para resolver a falha presente o . //

SE(conjunto_ações-causa_consultado = último conjunto ações-causa armazenado) ENTÃO
INICIO

- Informar ao usuário que não é possível resolver a falha com o conhecimento disponível no sistema;

FIM

FIM

//Fim da rotina para consulta sendo realizada a BAC.//

FIM//Fim do algoritmo de mudança de consulta ao conhecimento específico (BCO,BCOO) para consulta ao conhecimento geral (BAC). //

Anexo 2

Regras para Diagnóstico e Solução da Falha de Comunicação no Canal 1

SE NÃO Falha = Sistema ok
E: NÃO Falha = Erro 1
E: LED POWER = ACENDO
D: DM105 = PISCANDO
ENTAO
{ Falha = Falha de comunicação
SOLUÇÃO
{ Falha = Falha de comunicação
E: LED POWER = DESLIGADO
D: DM105 = ACENDO
ENTAO
{ Falha = Não há link de rede
SOLUÇÃO
{ Falha = Falha de comunicação
E: LED POWER = DESLIGADO
D: DM105 = PISCANDO
ENTAO
{ Falha = Falha de comunicação
E: CLP é nó da rede = SIM
ENTAO
{ Contador = 1
SOLUÇÃO
{ Falha = Falha de comunicação
E: CLP é nó da rede = NAO
ENTAO
{ Contador = 0

Regras para Diagnóstico e Solução da Falha de Comunicação no Canal 1

Regras Tipo 1 - Diagnóstico

Regra 1:

SE NÃO (Falha = Sistema de Energia)
E NÃO (Falha = Erro fatal de UCP)
E (LED POWER = ACESO)
E (DH485 = PISCANDO)
ENTÃO
(**Falha = Falha de comunicação no canal 1 da UCP**)

Regra 2:

SE NÃO (Falha = Sistema de Energia)
E NÃO (Falha = Erro fatal de UCP)
E (LED POWER = ACESO)
E (DH485 = ACESO)
ENTÃO
(**Falha = Não há falha no canal de comunicação 1 da UCP**)

Regras Tipo 2 – Determinação de Contexto

Regra 3:

SE (**Falha = Falha de comunicação no canal 1 da UCP**)
E (**CLP é nó de rede = Sim**)
ENTÃO
(Contexto = 1)

Regra 4:

SE (**Falha = Falha de comunicação no canal 1 da UCP**)
E (**CLP é nó de rede = Não**)
ENTÃO
(Contexto = 2)

Regras Tipo 2 para Determinação Subcontexto

Regra 5:

SE (Contexto = 1)
E (Outros nós perderam comunicação = Sim)
ENTÃO
(Subcontexto = 1.1)

Regra 6:

SE (Contexto = 1)
E (Outros nós perderam comunicação = Não)
E (Alimentação do Módulo AIC = Externa)
ENTÃO
(Subcontexto = 1.2)

Regra 7:

SE (Contexto = 1)
E (Outros nós perderam comunicação = Não)
E (Alimentação do Módulo AIC = Interna (UCP))
ENTÃO
(Subcontexto = 1.3)

Regra 8:

SE (Contexto = 2)
E (Tentativa de Estabelecer comunicação com outro dispositivo = Sim)
ENTÃO
(Subcontexto = 2.1)

Regra 9:

SE (Contexto = 2)
E (Tentativa de Estabelecer comunicação com outro dispositivo = Não)
ENTÃO
(Subcontexto = 2.2)

Regras Tipo 3 para Determinação da Versão

Regra 10:

SE (Subcontexto = 1.2)
E (LED POWER AIC = APAGADO)
ENTÃO
(Versão = 1)

Regra 11:

SE (Subcontexto = 1.2)
E (LED POWER AIC = ACESO)
ENTÃO
(Versão = 2)

Regras Tipo 4 para Determinação dos Conjuntos de Ações Corretivas e Causas

Contexto 1, Subcontexto 1.1

Regra 12:

SE (Contexto = 1)
E (Subcontexto = 1.1)
ENTÃO
(BAC = 01) [ou BCO = 01]

Contexto 1, Subcontexto 1.2, Versão 1

Regra 13:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)

E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 02) [ou BCO = 02]

Regra 14:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 03) [ou BCO = 03]

Regra 15:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 04) [ou BCO = 04]

Regra 16:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 05) [ou BCO = 05]

Regra 17:

SE (Contexto = 1)
E (Subcontexto = 1.2)

E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 06) [ou BCO = 06]

Regra 18:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 07) [ou BCO = 07]

Regra 19:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 08) [ou BCO = 08]

Regra 20:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 09) [ou BCO = 09]

Regra 21:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 10) [ou BCO = 10]

Regra 22:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 11) [ou BCO = 11]

Regra 23:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 12) [ou BCO = 12]

Regra 24:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 13) [ou BCO = 13]

Regra 25:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 14) [ou BCO = 14]

Regra 26:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 15) [ou BCO = 15]

Regra 27:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 16) [ou BCO = 16]

Regra 28:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 17) [ou BCO = 17]



Contexto 1, Subcontexto 1.2, Versão 2

Regra 29:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 18) [ou BCO = 18]

Regra 30:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 19) [ou BCO = 19]

Regra 31:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 20) [ou BCO = 20]

Regra 32:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)

E (Cabo AIC – CLP reserva = Sim)

ENTÃO

(BAC = 21) [ou BCO = 21]

Regra 33:

SE (Contexto = 1)

E (Subcontexto = 1.2)

E (Versão = 2)

E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)

E (Ferram. Prog. + Cabo RS-232 = Sim)

E (Mód. AIC reserva = Não)

E (Cabo AIC – CLP reserva = Não)

ENTÃO

(BAC = 22) [ou BCO = 22]

Regra 34:

SE (Contexto = 1)

E (Subcontexto = 1.2)

E (Versão = 2)

E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)

E (Ferram. Prog. + Cabo RS-232 = Sim)

E (Mód. AIC reserva = Não)

E (Cabo AIC – CLP reserva = Sim)

ENTÃO

(BAC = 23) [ou BCO = 23]

Regra 35:

SE (Contexto = 1)

E (Subcontexto = 1.2)

E (Versão = 2)

E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)

E (Ferram. Prog. + Cabo RS-232 = Sim)

E (Mód. AIC reserva = Sim)

E (Cabo AIC – CLP reserva = Não)

ENTÃO

(BAC = 24) [ou BCO = 24]

Regra 36:

SE (Contexto = 1)

E (Subcontexto = 1.2)

E (Versão = 2)

E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)

E (Ferram. Prog. + Cabo RS-232 = Sim)

E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 25) [ou BCO = 25]

Regra 37:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 26) [ou BCO = 26]

Regra 38:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 27) [ou BCO = 27]

Regra 39:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 28) [ou BCO = 28]

Regra 40:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)

E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 29) [ou BCO = 29]

Regra 41:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 30) [ou BCO = 30]

Regra 42:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 31) [ou BCO = 31]

Regra 43:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 32) [ou BCO = 32]

Regra 44:

SE (Contexto = 1)
E (Subcontexto = 1.2)
E (Versão = 2)

E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 33) [ou BCO = 33]

Contexto 1, Subcontexto 1.3

Regra 45:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 34) [ou BCO = 34]

Regra 46:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 35) [ou BCO = 35]

Regra 47:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 36) [ou BCO = 36]

Regra 48:

SE (Contexto = 1)

E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 37) [ou BCO = 37]

Regra 49:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 38) [ou BCO = 38]

Regra 50:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 39) [ou BCO = 39]

Regra 51:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 40) [ou BCO = 40]

Regra 52:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)

E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 41) [ou BCO = 41]

Regra 53:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 42) [ou BCO = 42]

Regra 54:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 43) [ou BCO = 43]

Regra 55:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 44) [ou BCO = 44]

Regra 56:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO

(BAC = 45) [ou BCO = 45]

Regra 57:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 46) [ou BCO = 46]

Regra 58:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Não)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 47) [ou BCO = 47]

Regra 59:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Não)
ENTÃO
(BAC = 48) [ou BCO = 48]

Regra 60:

SE (Contexto = 1)
E (Subcontexto = 1.3)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. AIC reserva = Sim)
E (Cabo AIC – CLP reserva = Sim)
ENTÃO
(BAC = 49) [ou BCO = 49]

Contexto 2, Subcontexto 2.1

Regra 61:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = não aplicável)

Regra 62:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = não aplicável)

Regra 63:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = não aplicável)

Regra 64:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = não aplicável)

Regra 65:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = 50) [ou BCO = 50]

Regra 66:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = não aplicável)

Regra 67:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = não aplicável)

Regra 68:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Não)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = não aplicável)

Regra 69:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = 51) [ou BCO = 51]

Regra 70:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = 52) [ou BCO = 52]

Regra 71:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = 53) [ou BCO = 53]

Regra 72:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Não)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = 54) [ou BCO = 54]

Regra 73:

SE (Contexto = 2)

E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = 55) [ou BCO = 55]

Regra 74:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Não)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = 56) [ou BCO = 56]

Regra 75:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Não)
ENTÃO
(BAC = 57) [ou BCO = 57]

Regra 76:

SE (Contexto = 2)
E (Subcontexto = 2.1)
E (Ferram. Prog. + Mód. PIC + Cabo PIC – CLP = Sim)
E (Ferram. Prog. + Cabo RS-232 = Sim)
E (Mód. PIC reserva = Sim)
E (Cabo PIC – CLP reserva = Sim)
ENTÃO
(BAC = 58) [ou BCO = 58]

Anexo 3

Conjuntos de Ações e Causas aplicáveis a Falha de Comunicação no Canal 1

Conjuntos de Ações e Causas aplicáveis a Falha de Comunicação no Canal 1

Tabela A1 – Contexto 1, Subcontexto 1.1

Nº Conj.	Classif.	Ações	Causa
1	1	1) Encontrar as extremidades do cabo de rede (local onde se encontra o último CLP em cada ponta da rede). 2) Observar se em cada extremidade da rede, o cabo de rede está conectado ao módulo AIC. Caso não esteja conectado, conectar. 3) Observar se, NAS DUAS extremidades do cabo de rede, os terminais 5 e 6 do conector do cabo de rede do módulo AIC estão conectados entre si. 4) Observar se no cabo de rede, EM APENAS UMA das extremidades, os terminais 1 e 2 do conector do cabo de rede do módulo AIC estão conectados entre si.	Ausência de ligação ou conexão incorreta do cabo de rede aos módulos AIC das extremidades da rede (terminadores).
2	1	1) Observar se os fios do cabo de rede conectados aos terminais 2 (blindagem do cabo) e 3 (comum) e 4 e 5 (dados) dos conectores dos módulos AIC de todos os CLPs da rede estão bem conectados e se não houve inversão em suas ligações em algum dos conectores dos módulos AIC que compõe a rede. 2) Caso haja algum fio do cabo de rede solto ou invertido, fixar o fio e/ou desinverter.	Mau contato ou inversão da ligação de algum fio do cabo de rede.
3	1	1) Acompanhar o cabo de rede e observar se esse não foi rompido. Caso haja algum segmento de cabo rompido, substituir o segmento. EVITE EMENDAR SEGMENTOS DE CABO.	Cabo de rede rompido.
4	1	1) Observar se a soma de todos os comprimentos dos segmentos de cabos de rede não ultrapassa 1219 metros. Caso a soma dos segmentos de cabo ultrapassar 1219 metros, rever a disposição física dos CLPs de forma a reduzir o comprimento total dos segmentos de cabo ou providenciar a instalação de um repetidor de sinal (Contatar a assistência técnica do fornecedor do CLP). 2) Observar se os cabos de rede não estão dispostos no campo em paralelo com os cabos da rede de energia elétrica. Caso estejam, separar os cabos de rede dos cabos de energia elétrica (passar os cabos de rede e energia por caminhos físicos diferentes).	Comprimento máximo dos segmentos dos cabos de rede desrespeitado ou perda de comunicação devido a interferência eletromagnética causada pela rede elétrica.

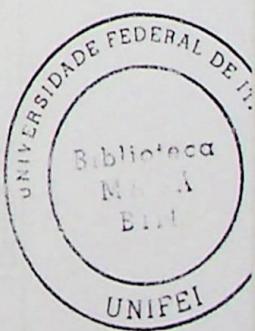


Tabela A2 – Contexto 1, Subcontexto 1.2, Versões 1 e 2

Nº Conj.	Classif.	Ações	Causa
1	1	1) Medir se o nível de tensão entre os terminais 24VDC e DCNEUT do módulo AIC (localizados na parte inferior do módulo) do CLP que perdeu comunicação está entre 18V e 30V. Caso seja maior que 30V ou menor que 18V, buscar a causa desse nível de tensão no sistema de fornecimento de energia ao módulo AIC. 1) Observar se o cabo de rede esta conectado ao módulo AIC do CLP que perdeu comunicação. Caso não esteja, conectar o cabo de rede ao módulo AIC do CLP que perdeu comunicação.	Tensão de alimentação do módulo AIC fora dos limites de operação tolerados para o bom funcionamento do módulo. Cabo de rede desconectado do módulo AIC do CLP que perdeu comunicação.
2	1	1) Observar se o cabo que liga o Canal 1 do CLP ao seu módulo AIC esta bem conectado. Caso não esteja bem conectado, conectar.	Cabo de ligação CLP - AIC solto ou mal encaixado.
3	1	1) Substituir o cabo CLP - AIC por um cabo reserva.	Cabo CLP - AIC com defeito.
4	2	1) Desconectar cabo CLP - AIC e cabo de rede do CLP que perdeu comunicação.	Módulo AIC com defeito. Enviar módulo AIC para assistência técnica do fabricante do CLP.
5	3	1) Desconectar cabos CLP - AIC e cabos de energia do módulo AIC. 2) Desconectar fios de energia do módulo AIC. 3) Retirar módulo AIC e substituir por outro módulo AIC reserva. 4) Reconectar cabos CLP - AIC e cabos de rede ao CLP que perdeu comunicação. 5) Reconectar fios de energia ao módulo AIC.	
6	3	1) Desconectar qualquer cabo conectado ao canal 0 (porta RS232) do CLP que perdeu comunicação. 2) Conectar a ferramenta de programação com cabo RS232 ao canal de comunicação 0 (porta RS232) do CLP que perdeu comunicação. 3) Abrir o programa de comunicação entre a ferramenta de programação e o CLP. 4) Configurar no programa de comunicação os seguintes parâmetros: a) Desativar qualquer comunicação ativa. b) Configurar o dispositivo de comunicação para AB-DF1 e ative o "driver" de comunicação. c) Configurar a porta de comunicação para porta serial COM1 ou COM2 (a que estiver disponível). d) Configurar o Protocolo de comunicação para DF1, "full duplex", "No Handshaking". e) Ajustar a taxa de transmissão de dados ("baund rate") para 1200 baud. f) Ativar os seguintes parâmetros de comunicação: "CRC Error Check", "Duplicate Packet Detect: On", "No Parity", "1 Stop Bit".	CLP que perdeu comunicação com endereço de nó de rede igual ao de outro CLP da rede (endereços de nó de rede duplicados) ou com endereço de nó maior que máximo endereço possível.

		<p>5) Estabelecer Comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED RS232 no painel da UCP do CLP começa a piscar. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate").</p> <p>6) Monitorar o canal de comunicação 1 da UCP.</p> <p>7) Configurar o maior número de nó de rede para o canal 1 para 31 ("Maximum node address = 31").</p> <p>8) Observar qual o número do nó correspondente ao CLP que perdeu comunicação. Observar se nenhum outro nó da rede tem o mesmo número. Caso algum outro nó da rede tenha o mesmo número do nó que perdeu comunicação, trocar o número do nó que perdeu comunicação para algum número não utilizado.</p>
7	3	<p>1) Desconectar o cabo CLP - AIC do canal de comunicação 1 (DH485) do CLP que perdeu comunicação.</p> <p>2) Conectar a ferramenta de programação com módulo PIC ao canal de comunicação 1 (DH485) do CLP que perdeu comunicação.</p> <p>3) Abrir programa de comunicação entre ferramenta de programação e CLP.</p> <p>4) Configurar os seguintes parâmetros no programa de comunicação entre ferramenta de programação e CLP:</p> <ul style="list-style-type: none"> a) Desabilitar qualquer comunicação ativa. b) Configurar dispositivo de comunicação para DH485PIC (AB-PIC). c) Configura a porta de comunicação serial da ferramenta de programação para COM1 ou COM2 (a que estiver disponível). d) Conferir se o número de rede atribuído a ferramenta de programação é o 0 ("Station Number = 0"). e) Ajustar a taxa de transmissão de dados ("baund rate") para 9600 baud (ou 19200 baud). <p>5) Estabelecer Comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED DH485 no painel da UCP do CLP fica acesso interrompido. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate").</p> <p>6) Monitorar o canal de comunicação 1 da UCP.</p> <p>7) Configurar o maior número de nó de rede para o canal 1 para 31 ("Maximum node address = 31").</p> <p>8) Observar qual o número do nó correspondente ao CLP que perdeu comunicação. Observar se nenhum outro nó da rede tem o mesmo número. Caso algum outro nó da rede tenha o mesmo número do nó que perdeu comunicação, trocar o número do nó que perdeu comunicação para algum número não utilizado.</p>

Tabela A3 – Contexto 1, Subcontexto 1.3

Nº Conj.	Classif.	Ações	Causa
1	1	1) Observar se o cabo de rede esta conectado ao módulo AIC do CLP que perdeu comunicação. Caso não esteja, conectar o cabo de rede ao módulo AIC do CLP que perdeu comunicação.	Cabo de rede desconectado do módulo AIC do CLP que perdeu comunicação.
2	1	1) Observar se o cabo que liga o Canal 1 do CLP ao seu módulo AIC esta bem conectado. Caso não esteja bem conectado, conectar.	Cabo de ligação CLP - AIC solto ou mal encaixado.
3	2	1) Substituir o cabo CLP - AIC por um cabo reserva.	Cabo CLP - AIC com defeito.
4	3	1) Desconectar o cabo CLP - AIC e o cabo de rede do módulo AIC. 2) Retirar o módulo AIC e substituir o mesmo por um em bom estado de funcionamento. 3) Conectar o cabo CLP - AIC e o cabo de rede ao módulo AIC.	Módulo AIC com defeito. Enviar o módulo AIC defeituoso para a assistência técnica do fabricante.
5	3	1) Desconectar qualquer cabo conectado ao canal 0 (porta RS232) do CLP que perdeu comunicação com endereço de nó de rede igual ao de outro CLP da rede (endereços de nó de rede duplicados) ou com endereço de nó maior que máximo endereço possível. 2) Conectar a ferramenta de programação com cabo RS232 ao canal de comunicação 0 (porta RS232) do CLP que perdeu comunicação. 3) Abrir o programa de comunicação entre a ferramenta de programação e o CLP. 4) Configurar no programa de comunicação os seguintes parâmetros: a) Desativar qualquer comunicação ativa. b) Configurar o dispositivo de comunicação para AB-DF1 e ative o "driver" de comunicação. c) Configurar a porta de comunicação para porta serial COM1 ou COM2 (a que estiver disponível). d) Configurar o Protocolo de comunicação para DF1, "full duplex", "No Handshaking". e) Ajustar a taxa de transmissão de dados ("baund rate") para 1200 baud. f) Ativar os seguintes parâmetros de comunicação: "CRC Error Check", "Duplicate Packet Detect: On", "No Parity", "1 Stop Bit". 5) Estabelecer Comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED RS232 no painel da UCP do CLP começa a piscar. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate"). 6) Monitorar o canal de comunicação 1 da UCP. 7) Configurar o maior número de rede para o canal 1 para 31 ("Maximum node address = 31").	Cabo de rede desconectado do módulo AIC do CLP que perdeu comunicação.

		<p>8) Observar qual o número do nó correspondente ao CLP que perdeu comunicação. Observar se nenhum outro nó da rede tem o mesmo número. Caso algum outro nó da rede tenha o mesmo número do nó que perdeu comunicação, trocar o número do nó que perdeu comunicação para algum número não utilizado.</p>
6	3	<p>1) Desconectar o cabo CLP - AIC do canal de comunicação 1 (DH485) do CLP que perdeu comunicação.</p> <p>2) Conectar a ferramenta de programação com módulo PIC ao canal de comunicação 1 (DH485) do CLP que perdeu comunicação.</p> <p>3) Abrir programa de comunicação entre ferramenta de programação e CLP.</p> <p>4) Configurar os seguintes parâmetros no programa de comunicação entre ferramenta de programação e CLP.</p> <ul style="list-style-type: none"> a) Desabilitar qualquer comunicação ativa. b) Configurar dispositivo de comunicação para DH485PIC (AB-PIC). c) Configura a porta de comunicação serial da ferramenta de programação para COM1 ou COM2 (a que estiver disponível). d) Conferir se o número de nó de rede atribuído a ferramenta de programação é o 0 ("Station Number = 0"). e) Ajustar a taxa de transmissão de dados ("baund rate") para 9600 baund (ou 19200 baund). <p>5) Estabelecer Comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED DH485 no painel da UCP do CLP fica acesso interruptamente. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate").</p> <p>6) Monitorar o canal de canal de comunicação 1 da UCP.</p> <p>7) Configurar o maior número de nó de rede para o canal 1 para 31 ("Maximum node address = 31").</p> <p>8) Observar qual o número do nó correspondente ao CLP que perdeu comunicação. Observar se nenhum outro nó da rede tem o mesmo número. Caso algum outro nó da rede tenha o mesmo número do nó que perdeu comunicação, trocar o número do nó que perdeu comunicação para algum número não utilizado.</p>

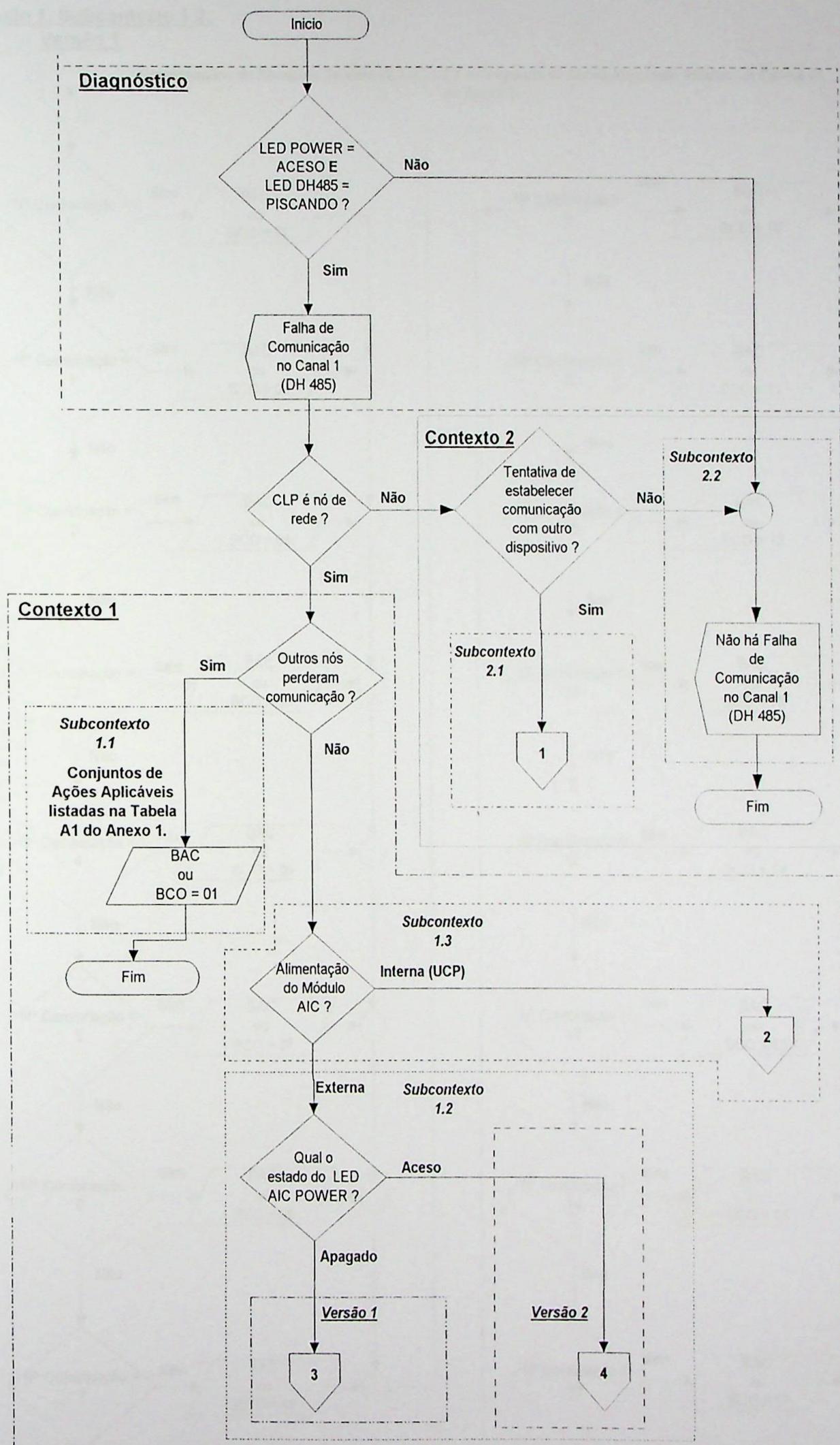
Tabela A4 – Contexto 2, Subcontexto 2.1

Nº Conj.	Classif.	Ações	Causa
1	1	1) Observar se o módulo PIC está conectado a porta serial (COM1 ou COM2) da ferramenta de programação. Caso não esteja, conectar. 1) Observar se o cabo que conecta o módulo PIC ao CLP está encaixado de forma correta. Caso não esteja, encaixar.	Módulo PIC não conectado a ferramenta de programação. Cabo PIC - CLP solto ou encaixado de forma incorreta.
2	1		
3	3	1) Abrir programa de comunicação entre ferramenta de programação e CLP. 2) Configurar os seguintes parâmetros no programa de comunicação entre ferramenta de programação e CLP: a) Desabilitar qualquer comunicação ativa. b) Configurar dispositivo de comunicação para DH485PIC (AB-PIC). c) Configura a porta de comunicação serial da ferramenta de programação para COM1 ou COM2 (a que estiver disponível). d) Conferir se o número de nó de rede atribuído a ferramenta de programação é o 0 ("Station Number = 0"). e) Ajustar a taxa de transmissão de dados ("baund rate") para 9600 baund (ou 19200 baund).	Algum parâmetro de comunicação do canal 1 (DH485) configurado de forma incorreta ou não configurado.
4	3	3) Estabelecer comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED DH485 no painel da UCP do CLP fica acesso interrompimento. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate"). 4) Desconectar qualquer cabo conectado ao canal 0 (porta RS232) do CLP. 2) Conectar a ferramenta de programação com cabo RS232 ao canal de comunicação 0 (porta RS232) do CLP que perdeu comunicação. 3) Abrir o programa de comunicação entre a ferramenta de programação e o CLP. 4) Configurar no programa de comunicação os seguintes parâmetros: a) Desativar qualquer comunicação ativa. b) Configurar o dispositivo de comunicação para AB-DF1 e ative o drive de comunicação. c) Configurar a porta de comunicação para porta serial COM1 ou COM2 (a que estiver disponível). d) Configurar o Protocolo de comunicação para DF1, " full duplex" , "No Handshaking". e) Ajustar a taxa de transmissão de dados ("baund rate") para 1200 baund. ñ Ativar os seguintes parâmetros de comunicação: "CRC Error Check", "Duplicate Packet"	Algum parâmetro de comunicação do canal 1 (DH485) configurado de forma incorreta ou não configurado.

		"Detect: On", "No Parity", "1 Stop Bit".	
		5) Estabelecer Comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED RS232 no painel da UCP do CLP começa a piscar. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate").	
		6) Monitorar o canal de comunicação 1 da UCP.	
		7) Configurar os seguintes parâmetros no programa de comunicação entre ferramenta de programação e CLP.	
		a) Desabilitar qualquer comunicação ativa.	
		b) Configurar dispositivo de comunicação para DH485PIC (AB-PIC).	
		c) Configura a porta de comunicação serial da ferramenta de programação para COM1 ou COM2 (a que estiver disponível).	
		d) Conferir se o número de rede atribuído a ferramenta de programação é o 0 ("Station Number = 0").	
		e) Ajustar a taxa de transmissão de dados ("baund rate") para 9600 baud (ou 19200 baud).	
		8) Encerrar a comunicação com o canal 0 (RS232) do CLP.	
		9) Desconectar cabo RS232 da ferramenta de programação.	
		10) Conectar o módulo PIC a porta serial (COM1 ou COM2) da ferramenta de programação. Conectar o cabo do módulo PIC ao canal 1 (DH485) de comunicação do CLP.	
		11) Estabelecer Comunicação entre a ferramenta de programação e o CLP. Caso se obtenha sucesso nessa operação, o LED DH485 no painel da UCP do CLP fica acesso intermitentemente. Caso não obtenha sucesso, testar outras opções de taxa de transmissão ("baund rate").	
5	3	1) Substituir o módulo PIC por um módulo reserva em bom estado de funcionamento.	Módulo PIC com defeito. Encaminhar para a assistência técnica do fabricante.
6	3	1) Substituir o cabo PIC - CLP por um reserva em bom estado.	Cabo PIC - CLP rompido ou com algum mau contato.

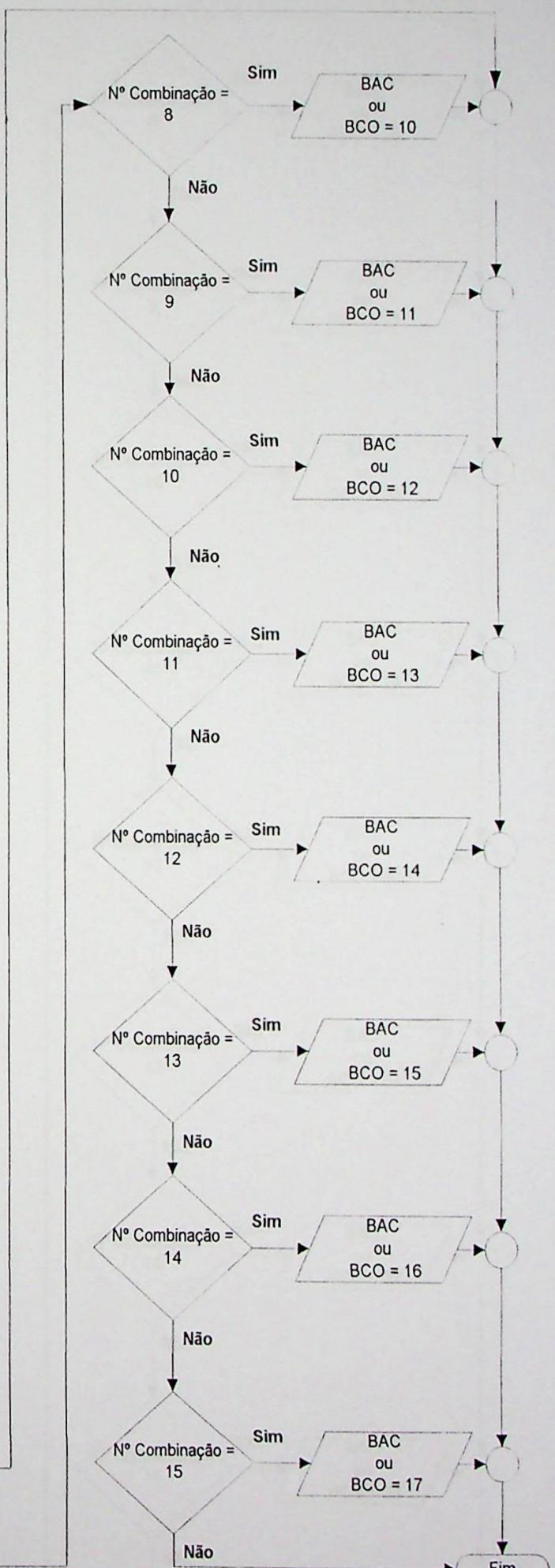
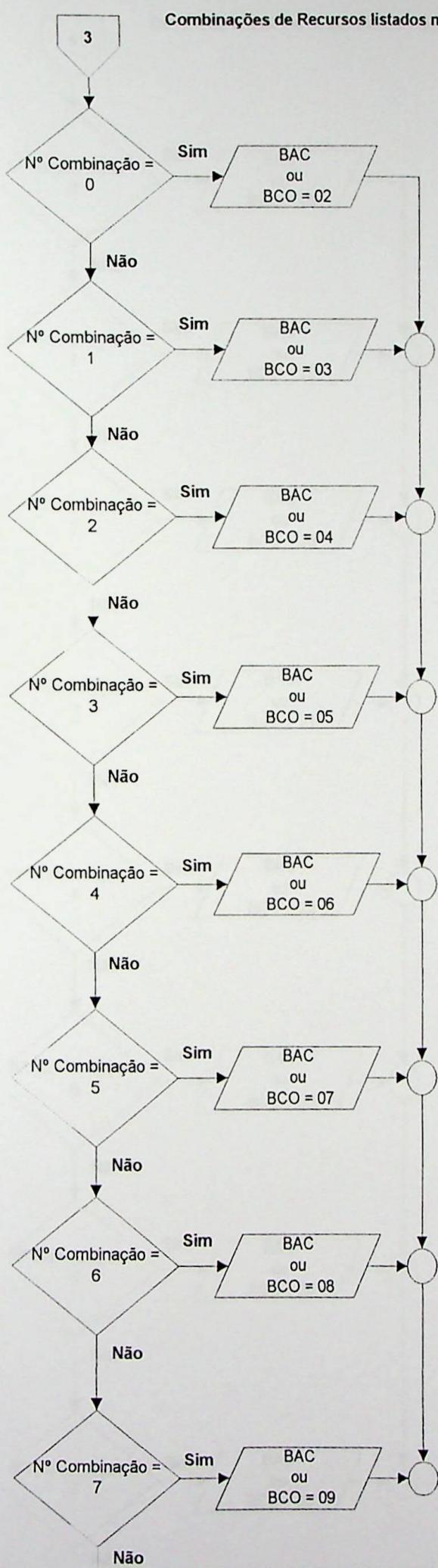
Anexo 4

Árvore de decisões para resolução da falha de comunicação no canal 1



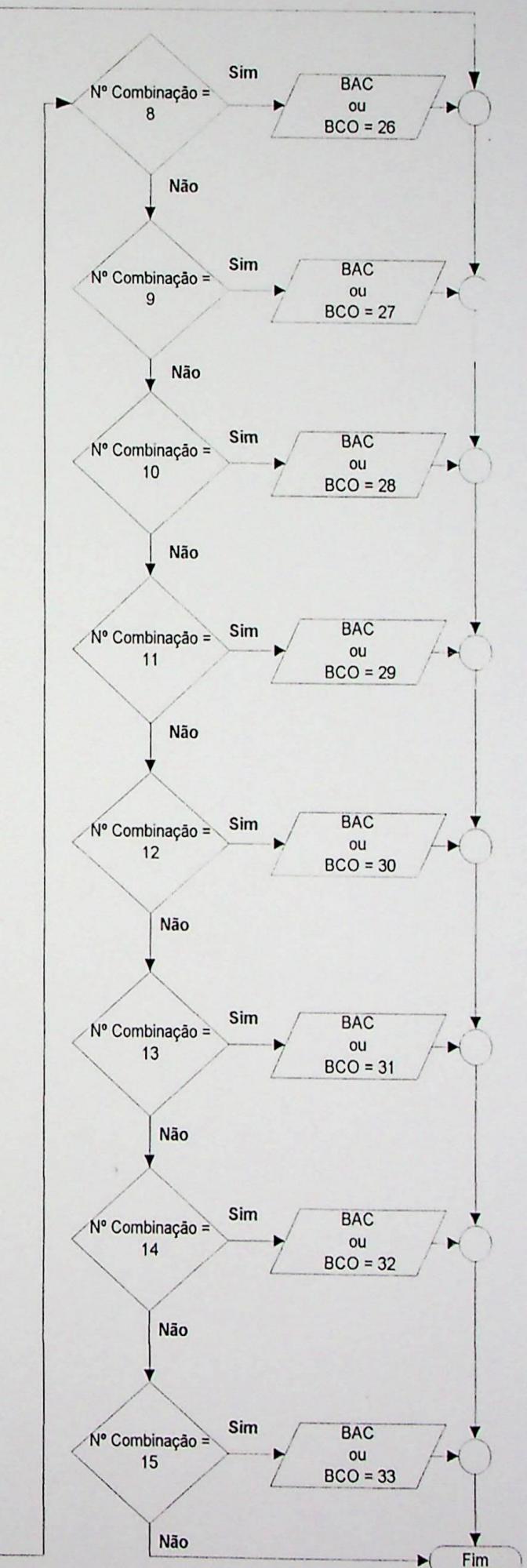
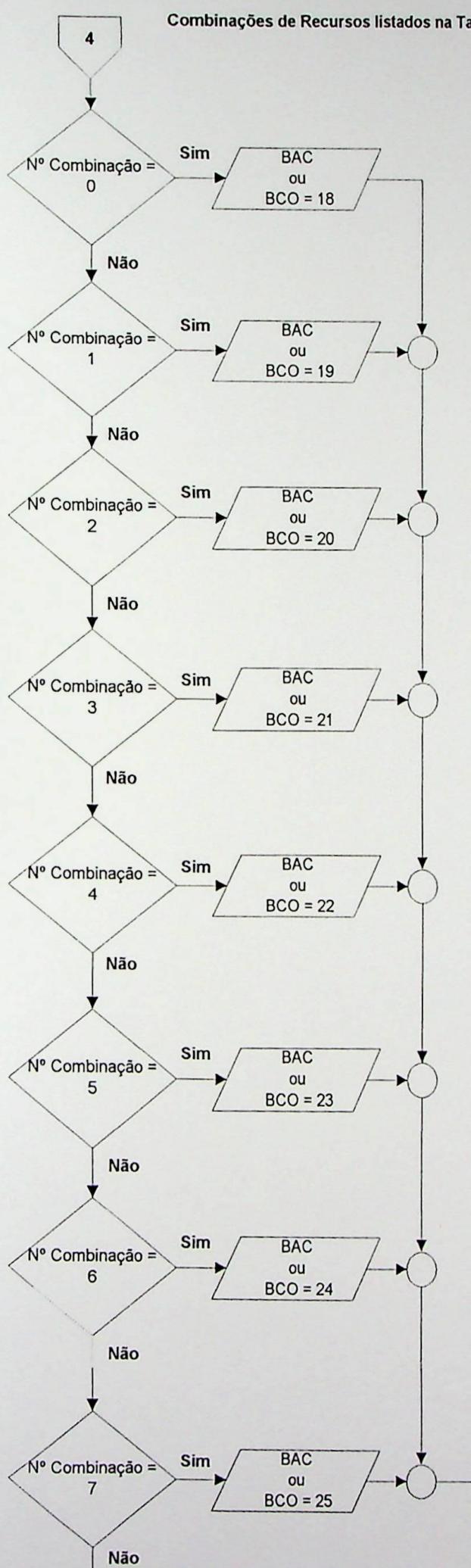
Contexto 1, Subcontexto 1.2,
Versão 1

Combinações de Recursos listados na Tabela 5.7 e Conjuntos de Ações Aplicáveis listadas na Tabela A2 do Anexo 3.



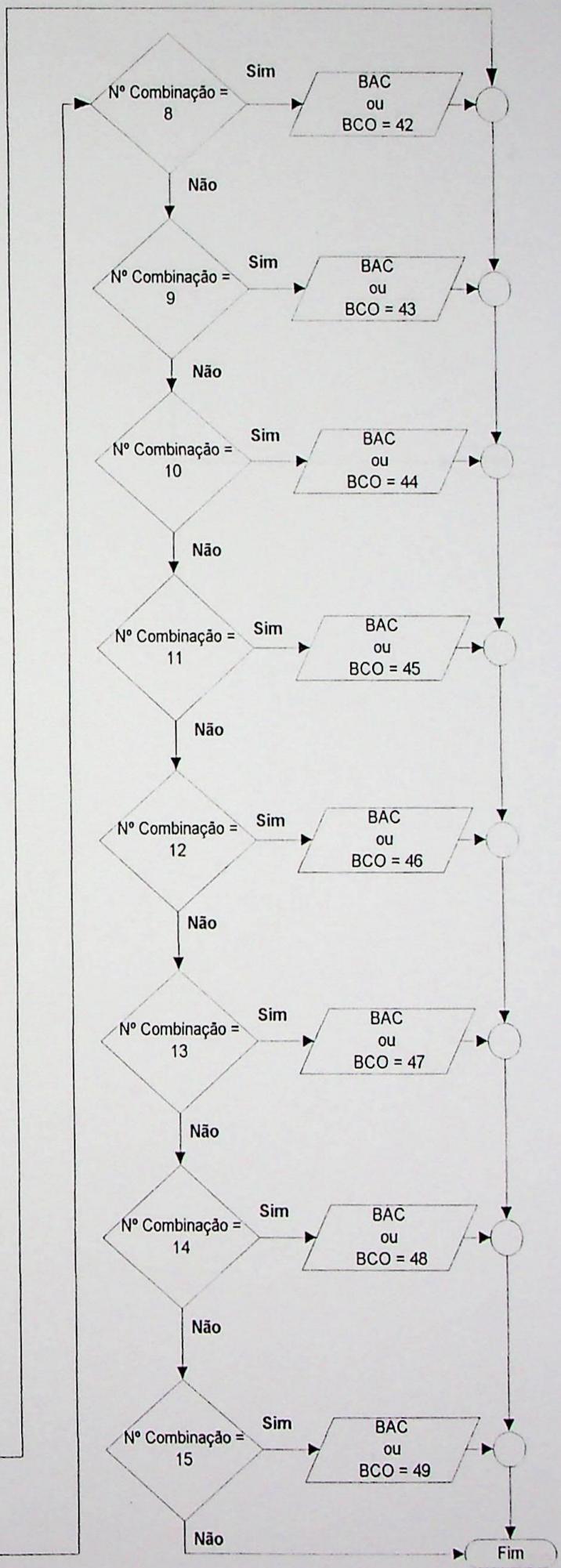
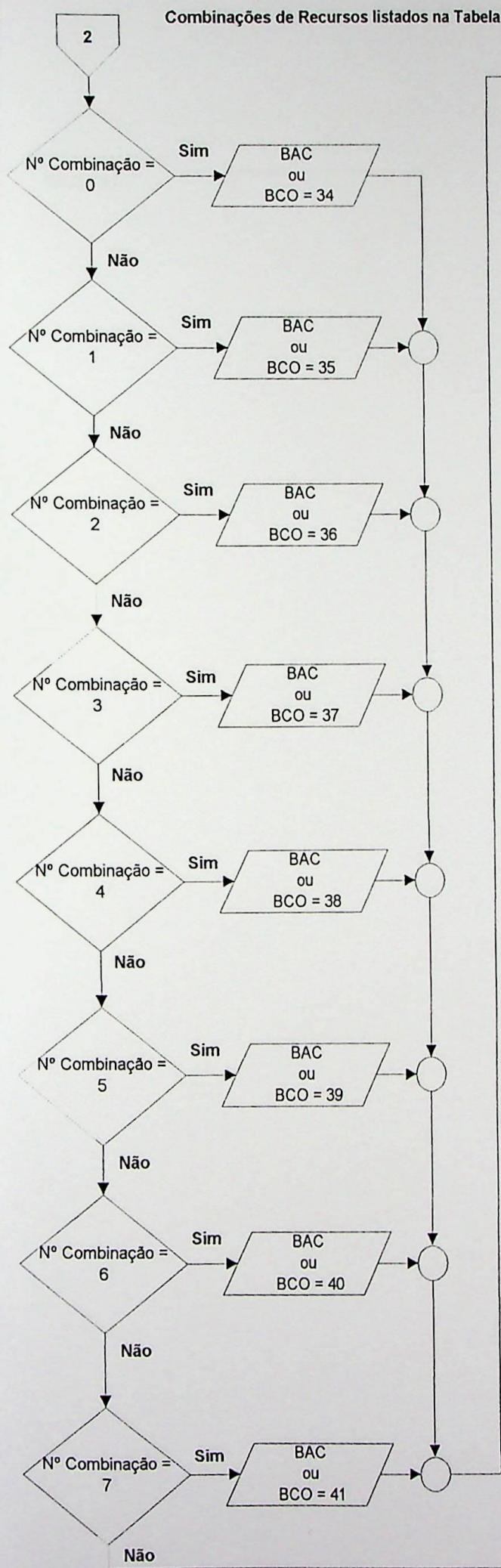
Contexto 1, Subcontexto 1.2,
Versão 2

Combinações de Recursos listados na Tabela 5.7 e Conjuntos de Ações Aplicáveis listadas na Tabela A2 do Anexo 3.



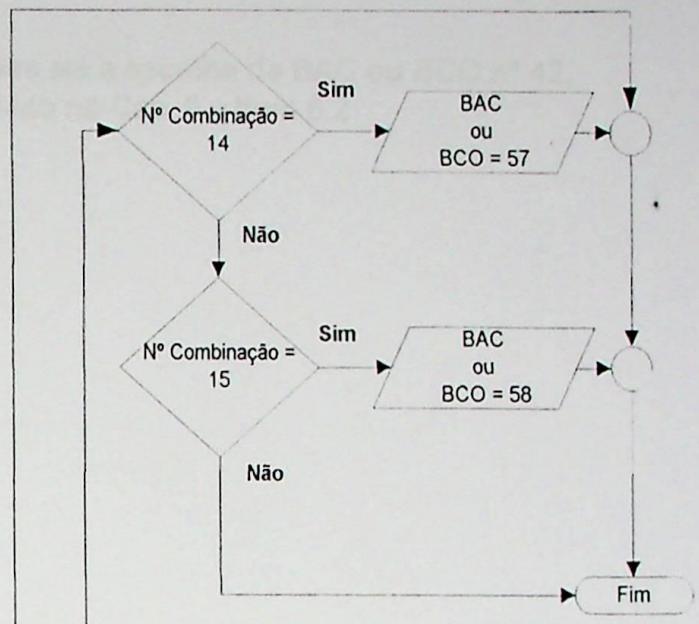
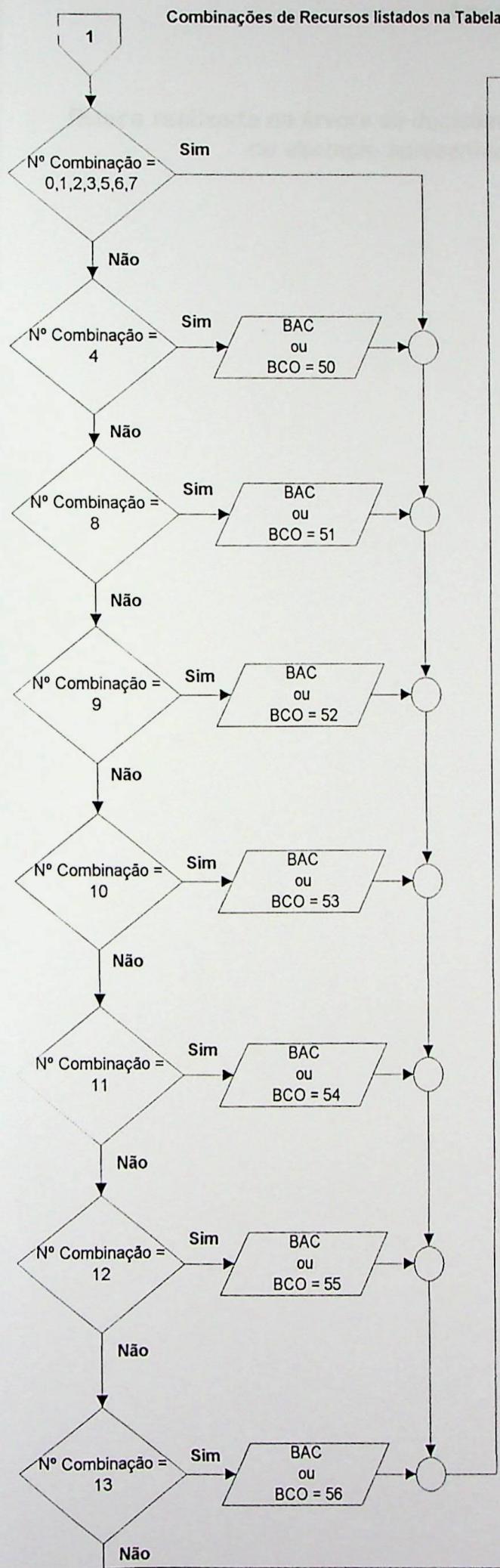
Contexto 1, Subcontexto 1.3

Combinações de Recursos listados na Tabela 5.7 e Conjuntos de Ações Aplicáveis listados na Tabela A3 do Anexo 3.



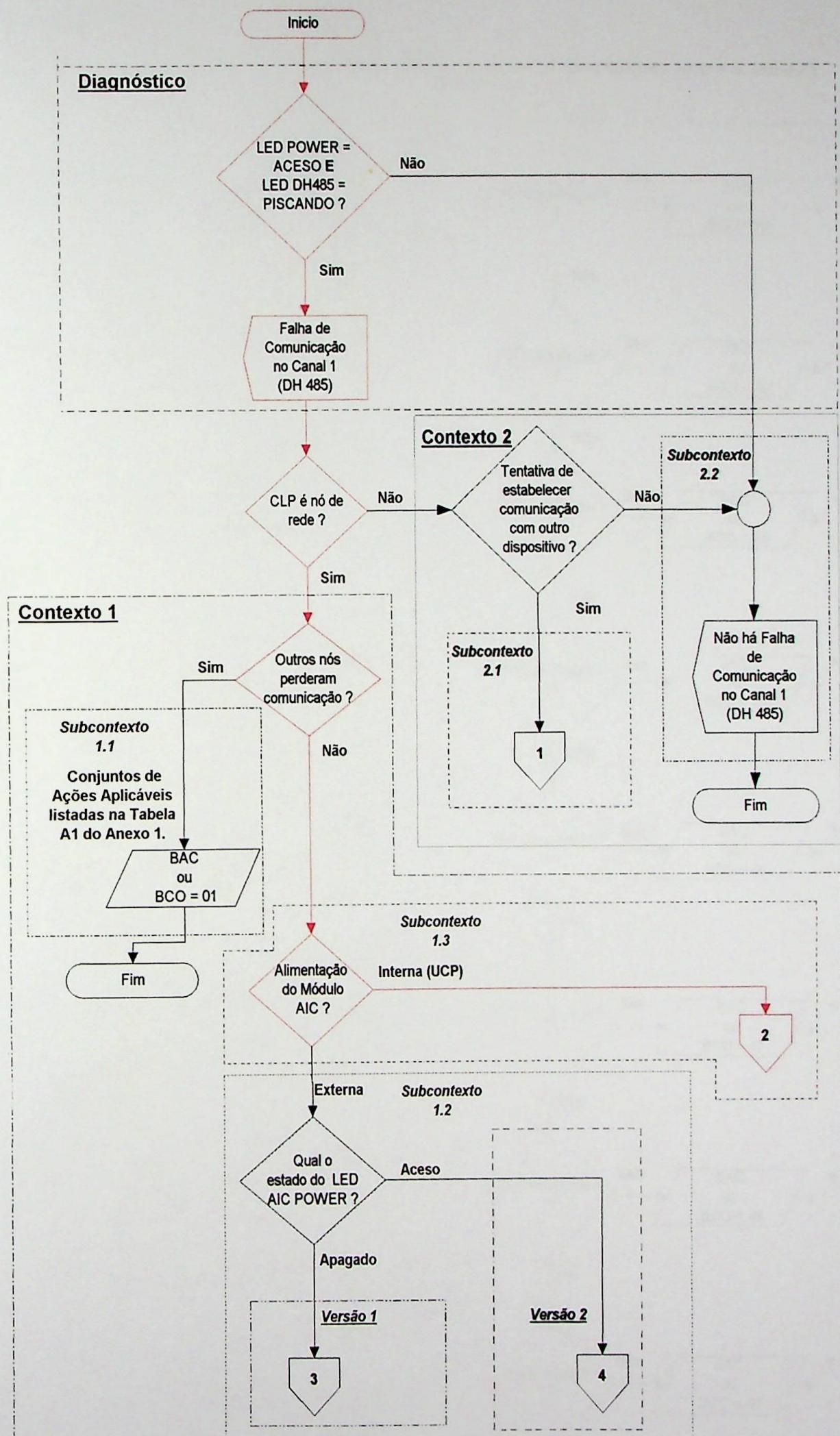
Contexto 2, Subcontexto 2.1

Combinações de Recursos listados na Tabela 5.8 e Conjuntos de Ações Aplicáveis listadas na Tabela A4 do Anexo 3.



Anexo 5

**Busca realizada na árvore de decisões até a escolha da BAC ou BCO nº 42,
no exemplo apresentado no Cap. 6 – Item 6.2**



Contexto 1. Subcontexto 1.3

Combinações de Recursos listados na Tabela 5.7 e Conjuntos de Ações Aplicáveis listadas na Tabela A3 do Anexo 1.

