**UNIFACS UNIVERSIDADE SALVADOR**
**MESTRADO EM SISTEMAS E COMPUTAÇÃO**

**THIAGO DIAS FREIRE DE MELLO**

# UMA ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PARA RESOLUÇÕES DE INCIDENTES EM TI CENTRADA NA COMBINAÇÃO DE RACIOCÍNIO BASEADO EM CASOS E BASEADO EM REGRAS

Salvador
2015

**THIAGO DIAS FREIRE DE MELLO**

# UMA ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PARA RESOLUÇÕES DE INCIDENTES EM TI CENTRADA NA COMBINAÇÃO DE RACIOCÍNIO BASEADO EM CASOS E BASEADO EM REGRAS

Dissertação apresentada ao Mestrado em Sistemas e Computação da UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial para a obtenção do título de Mestre.

Orientador: Prof. Dr. Expedito Carlos Lopes.

Salvador
2015

FICHA CATALOGRÁFICA
Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador, Laureate International Universities)

Mello, Thiago Dias Freire de

Uma abordagem metodológica de apoio à decisão para resoluções de incidentes em TI centrada na combinação de raciocínio baseado em casos e baseado em regras.

137 f. : il.

Dissertação Programa de Pós-Graduação em Sistemas e Computação de UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Expedito Carlos Lopes.

1.Tecnologia da informação. 2. Sistemas de informação gerencial. 3. ITIL. I. Lopes, Expedito Carlos, orient. II. Título.

CDD: 658.403811

THIAGO DIAS FREIRE DE MELLO

UMA ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PARA RESOLUÇÕES DE INCIDENTES EM TI CENTRADA NA COMBINAÇÃO DE RACIOCÍNIO BASEADO EM CASOS E BASEADO EM REGRAS

Dissertação de Mestrado aprovada como requisito parcial para obtenção do grau de mestre em Sistemas e Computação, UNIFACS Universidade Salvador, Laureate International Universities pela seguinte banca examinadora:

Expedito Carlos Lopes – Orientador _____
Doutor em Ciência da Computação pela Universidade Federal de Campina Grande,
UNIFACS Universidade Salvador, Laureate International Universities

Éldman de Oliveira Nunes _____
Doutor em Computação pela Universidade Federal Fluminense
UNIFACS Universidade Salvador, Laureate International Universities

Leobino Nascimento Sampaio _____
Doutor em Ciência da Computação pela Universidade Federal de Pernambuco
Universidade Federal da Bahia - UFBA

Salvador, de de 2015.

## AGRADECIMENTOS

A minha noiva Uldi Leoni, pela sua compreensão e incentivo nos momentos mais complicados que passei para construção deste trabalho.

Aos meus pais, por estarem sempre me incentivando a buscar mais conhecimento e tornar-me cada dia melhor.

Aos meus colegas de trabalho, que compreenderam a importância do desenvolvimento deste trabalho e constantemente estavam me incentivando.

Ao meu orientador, por ter compartilhado a sua experiência e conhecimento.

**RESUMO**

Nos dias atuais, os serviços oferecidos pela Tecnologia da Informação (TI) tornou-se um fator primordial para que as organizações alcancem o sucesso. A interrupção inesperada destes serviços, conhecida como incidentes, causam prejuízos para as organizações que podem chegar a milhões de dólares. Decisões e ações são tomadas constantemente por equipes de suporte para reparar e normalizar os serviços de TI ocasionados por incidentes. Um processo de tomada de decisão eficiente exige sincronismo entre as ações a serem tomadas por todos os membros de uma equipe. Muitas organizações adotam a *Information Technology Infrastructure Library* (ITIL) para estruturar equipes e viabilizar as suas tomadas de decisões. Por sua vez, duas técnicas da inteligência artificial, Raciocínio Baseado em Casos e Raciocínio Baseado em Regras, têm auxiliado o processo de tomada de decisão, utilizando o raciocínio por regra e por analogia, respectivamente. Por outro lado, é preciso estruturar em qual momento e quando se utilizar cada um desses raciocínios no apoio à decisão, definindo e ordenando um conjunto de passos. Este trabalho apresenta uma abordagem metodológica para apoiar tomadas de decisão na resolução de incidentes de TI, descrita através de um processo de decisão geral, respeitando um modelo de decisão consolidado, o modelo de decisão de Simon, auxiliado pela combinação das técnicas do Raciocínio Baseado em Casos (RBC) e do Raciocínio Baseado em Regras (RBR), visto que o ITIL não propõe nas suas melhores práticas a resolução de incidentes por analogia e nem por regra. Portanto, o objetivo é apoiar a tomada de decisão das equipes de suporte através de uma abordagem metodológica, com o intuito de resolver os incidentes ocorridos, considerando o raciocínio por analogia e por regra. Um protótipo foi construído para apoiar à tomada de decisão com esses dois raciocínios citados utilizando-se os *frameworks JColibri* e *Expert Sinta*. Um estudo de caso foi realizado dentro de uma organização com o intuito de avaliar a proposta deste trabalho, sendo resolvidos, com êxito, quatro casos de incidentes com a aplicação da abordagem metodológica construída.

**Palavras-chave**: Raciocínio Baseado em Casos. Raciocínio Baseado em Regras. Tomada de decisão. Abordagem Metodológica de apoio à decisão. ITIL.

**ABSTRACT**

*In present day, the services offered by Information Technology (IT) has become a key factor for organizations to achieve success. The unexpected interruption of these services, known as incidents, causing damage to organizations that can reach millions of dollars. Decisions and actions are constantly taken for support teams to repair and to normalize the services caused by these incidents. The process of efficient decision-making requires timing and standardization among all members of a team. Many organizations adopt Information Technology Infrastructure Library (ITIL) to structure and facilitate their decision-making. On its turn, two important models of artificial intelligence, Case-Based Reasoning (CBR) and Rule Based Reasoning (RBR), has aided the decision-making process using reasoning by analogy and rule, respectively. On the other hand we must structure at what time and when to use each of these arguments in decision support, defining and ordering a set of steps. This paper presents a methodological approach to support decision making in resolving IT incidents, described through a general decision process respecting a consolidated decision model, the model Simon decision, aided by the combination of techniques Case Based Reasoning (CBR) and Rules Based Reasoning (RBR), since ITIL does not propose the best practices for incident resolution analogy nor by rule. Therefore, the aim is to support decision-making of the support team through a methodological approach, in order to solve the incidents, considering the reasoning by analogy and reasoning by rule. A prototype was built to support decision-making with these two cited arguments relying on frameworks JColibri and Expert Sinta. A case study was carried out within an organization in order to evaluate the proposal of this work, being resolved four cases of incidents using the prototype with the built methodological approach.*

**Keywords:** *Case-Based Reasoning, Rules-Based Reasoning, Decision-making, Decision support methodological approach, ITIL.*

## LISTA DE FIGURAS

Figura 1 - Fases do processo decisório de Simon ..... 24
Figura 2 - Níveis de suporte para resolução de incidentes de TI ..... 27
Figura 3 - Atividades para resolução de incidentes de TI baseado no ITIL ..... 28
Figura 4 - Ciclo básico do RBC ..... 31
Figura 5 - Regra para mascar chiclete ..... 39
Figura 6 – Arquitetura e componentes do RBR ..... 40
Figura 7 – Regras para sistema especialista de problemas automotivos ..... 41
Figura 8 - Regra1 na memória de trabalho ..... 42
Figura 9 - Regra3 acionada na memória de trabalho ..... 42
Figura 10 - Regra 2 na memória de trabalho ..... 43
Figura 11 - Regras 2 e 3 na memória de trabalho ..... 44
Figura 12 – Classificação dos modelos de combinação RBC/RBR ..... 47
Figura 13 – Elaborar especificação inicial ..... 50
Figura 14 – Iniciar implementação ..... 50
Figura 15 – Concluir especificação ..... 51
Figura 16 – Concluir implementação ..... 51
Figura 17 - Modelo de processo RBC estendido para geração de relatórios de segurança de rede ..... 53
Figura 18 – Fases do processo decisório com RBC ..... 55
Figura 19 – Ciclo RBC adaptado para o sistema de penas alternativas ..... 56
Figura 20 - Imagem explicativa do sistema desenvolvido ..... 58
Figura 21 – Categorização dos componentes relacionados ao modem ..... 61
Figura 22 – Novo fluxo de trabalho RBC ..... 62
Figura 23 - Representação do caso ..... 67
Figura 24 – Visão geral da abordagem metodológica proposta ..... 71
Figura 25 - Processo de tomada de decisão para o diagnóstico inicial utilizando RBR. 73
Figura 26 - Processo de tomada de decisão para a atividade investigação e diagnóstico utilizando RBC ..... 76
Figura 27 - Processo de tomada de decisão para a atividade resolução e recuperação utilizando RBC com RBR ..... 78
Figura 28 - Consulta base de regras do Problema do incidente ..... 81
Figura 29 - Atividades completas para Abordagem metodológica proposta ..... 84

Figura 30 – Diagrama de casos de uso do protótipo ..... 87
Figura 31 – Arquitetura do protótipo..... 89
Figura 32 - Ocorrência de incidente 1 - aberta por ferramenta web..... 93
Figura 33 – Tela inicial do protótipo ..... 93
Figura 34 - Tela de escolha da situação-problema para regra ..... 94
Figura 35 – Regra 1 apresentada ..... 95
Figura 36 - Regra 2 apresentada ..... 95
Figura 37 – Regra 3 apresentada ..... 96
Figura 38 – Regra 4 apresentada ..... 96
Figura 39 – Regra 5 apresentada ..... 97
Figura 40 – Conclusão com solução para o incidente ..... 97
Figura 41 – Registro do caso 1 na base de casos..... 98
Figura 42 – Ocorrência de incidente 2..... 99
Figura 43 – Pesquisa situação-problema ..... 100
Figura 44 – Resultado da recuperação da situação-problema ..... 100
Figura 45 – Consulta informação do IC (circuito de dados de Paulo Afonso)..... 101
Figura 46 – Consulta informações do IC (SERVER_BD) ..... 102
Figura 47 – Consulta problemas similares ..... 102
Figura 48 – Resultado da recuperação para descrição do problema..... 103
Figura 49 – Registro do novo caso ..... 104
Figura 50 - Ocorrência de incidente 3 - aberta por ferramenta web..... 105
Figura 51 – Regra 4 apresentada ..... 106
Figura 52 – Regra 5 apresentada ..... 106
Figura 53 – Pesquisa situação-problema – caso3 ..... 107
Figura 54 – Casos recuperados da situação-problema – caso 3 ..... 107
Figura 55 – Consulta serviço NTDS..... 108
Figura 56– Consulta problemas similares - caso 3 ..... 109
Figura 57 – Resultado da recuperação para descrição problema - caso 3 ..... 109
Figura 58 – Ocorrência de incidente 4 - aberta por ferramenta web ..... 111
Figura 59 - Consulta problemas similares - caso 4..... 112
Figura 60 - Resultado da recuperação para descrição problema - caso 4..... 112
Figura 61 - Tela de escolha do problema de incidente da regra ..... 113
Figura 62 – Questionamento de problema 1..... 114

Figura 63 - Questionamento de problema 2 ..... 114
Figura 64 – Conclusão das regras para problema de incidente ..... 114

## LISTA DE QUADROS

Quadro 1 – Vantagens e desvantagens das técnicas de raciocínio baseado em casos e baseado em regras ..... 46

Quadro 2 - Análise comparativa dos trabalhos correlatos ..... 64

Quadro 3 - Similaridade local para os atributos do caso ..... 68

Quadro 4 – Regras implementadas para situação-problema “sem acesso aos sistemas da organização” ..... 69

Quadro 5 - Regras implementadas para situação-problema “não ingressa as pastas de rede” ..... 70

Quadro 6 – Regras para desbloqueio de porta web ..... 70

## LISTA DE ABREVIATURAS

| | |
|---|---|
| AD | Active Directory |
| BD | Banco de Dados |
| BPMN | *Business Process Model and Notation* |
| DB | *Data Base* |
| DHCP | Dynamic Host Configuration Protocol |
| IC | Item de Configuração |
| ID | Identificação |
| IDS | *Intrusion Detection System* |
| IIS | *Internet Information Services* |
| IP | *Internet Protocol* |
| ITIL | *Information Technology  Infrastructure Library* |
| LIA | Laboratório de Inteligência Artificial |
| MBA | Meningite Bacteriana Aguda |
| MPLS | *Multiprotocol Label Switching* |
| NTDS | NT *Directory Service* |
| POP | Programação Orientada ao Problema |
| RBC | Raciocínio Baseado em Casos |
| RBR | Raciocínio Baseado em Regras |
| SQL | *Structure Query Language* |
| TCP | *Transmission Control Protocol* |
| TI | Tecnologia da Informação |
| UML | *Unified Modeling Language* |
| UTI | Unidade de Tratamento Intensivo |

**SUMÁRIO**

**1 INTRODUÇÃO ..... 16**
1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO ..... 16
1.2 OBJETIVO ..... 19
1.3 METODOLOGIA ..... 20
1.4 ESTRUTURA DA DISSERTAÇÃO ..... 20
**2 FUNDAMENTAÇÃO TEÓRICA ..... 22**
2.1 PROCESSO DE TOMADA DE DECISÃO ..... 22
**2.1.1 Conceitos de Tomada de Decisão ..... 22**
**2.1.2 Processo de Tomada de Decisão ..... 23**
2.2 A BIBLIOTECA ITIL (INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY) ..... 25
**2.2.1 Gerenciamento dos Serviços de TI e a ITIL ..... 25**
**2.2.2 Gerenciamento de Incidentes de TI e a Análise do Ambiente ..... 26**
**2.2.3 Processo, Métodos e Técnicas para Resolução de Incidentes de TI ..... 28**
**2.2.4 Gerenciamento de Configuração ..... 29**
2.3 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA APOIO À DECISÃO ..... 30
**2.3.1 Raciocínio Baseado em Casos ..... 30**
**2.3.1.1 Ciclo Básico de Processamento em RBC ..... 31**
**2.3.1.2 Representação de Caso ..... 32**
**2.3.1.3 Vantagens do RBC ..... 37**
**2.3.1.4 Raciocínio Baseado em Regras ..... 38**
**2.3.1.5 Componentes e arquitetura do Raciocínio Baseado em Regras ..... 40**
**2.3.1.6 Resolvendo Problema com Raciocínio Baseado em Regras ..... 41**
**2.3.1.7 Procedimentos para Construção de Base de Regras ..... 44**
**2.3.2 Combinando Raciocínio Baseado em Casos e Raciocínio Baseado em Regras ..... 45**
**3 TRABALHOS CORRELATOS ..... 48**
3.1 PROGRAMAÇÃO ORIENTADA AO PROBLEMA: UMA METODOLOGIA PARA O ENTENDIMENTO DE PROBLEMAS E ESPECIFICAÇÃO NO CONTEXTO DE ENSINO DE PROGRAMAÇÃO PARA INICIANTES ..... 49
3.2 ESTENDENDO O RACIOCÍNIO BASEADO EM CASOS PARA GERAR RELATÓRIOS DE ALERTA DE REDES ..... 52
3.3 RACIOCÍNIO BASEADO EM CASOS APLICANDO NO PROCESSO DECISÓRIO ..... 54

3.4 METODOLOGIA DE APOIO À DECISÃO PARA O CONTROLE DE PENAS ALTERNATIVAS: UMA ABORDAGEM UTILIZANDO RACIOCÍNIO BASEADO EM CASOS ..... 55

3.5 INTEGRAÇÃO DE SISTEMA ESPECIALISTA BASEADO EM REGRAS E RACIOCÍNIO BASEADO EM CASOS PARA UM SISTEMA DE APOIO À DECISÃO CLÍNICA DA MENINGITE BACTERIANA AGUDA. ..... 57

3.6 UMA ABORDAGEM BASEADA EM RBC PARA CENTRAL DE SERVIÇOS BASEADAS NO ITIL ..... 60

3.7 ANÁLISE COMPARATIVA DOS TRABALHOS RELACIONADOS ..... 63

**4 ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PROPOSTA ..... 66**

4.1 CONSIDERAÇÕES INICIAIS ..... 67

**4.1.1 Representação de caso para utilização do RBC ..... 67**

**4.1.2 Representação das regras para utilização do RBR ..... 68**

4.2 VISÃO GERAL DA ABORDAGEM METODOLÓCIA ..... 70

4.3 DETALHAMENTO DA ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PARA RESOLUÇÃO DE INCIDENTES DE TI ..... 72

**4.3.1 Diagnóstico Inicial Auxiliado por RBR ..... 72**

**4.3.2 Investigação e Diagnóstico Auxiliados por RBC ..... 75**

**4.3.3 Resolução e Recuperação Auxiliadas pela Combinação RBC com RBR ..... 78**

4.4 ABORDAGEM METODOLÓGICA - PROPOSTA FINAL ..... 82

**5 ESTUDO DE CASO APLICADO À RESOLUÇÃO DE INCIDENTES DE TI ... 85**

5.1 QUESTIONÁRIO APLICADO PARA AVALIAR A ABORDAGEM METODOLÓGICA PROPOSTA ..... 85

5.2 ARTEFATOS PARA O DESENVOLVIMENTO DO PROTÓTIPO ..... 86

**5.2.1 Casos de Uso ..... 86**

**5.2.2 Projeto Arquitetural ..... 88**

**5.2.3 Tecnologias utilizadas ..... 90**

**5.2.3.1 JCOLIBRI ..... 90**

**5.2.3.2 Expert Sinta ..... 91**

5.3 APLICANDO A ABORDAGEM METODOLÓGICA PROPOSTA PARA RESOLUÇÃO DE INCIDENTES DE TI ..... 92

**5.3.1 Caso 1 - Sem acesso aos sistemas financeiro e recibos ..... 92**

**5.3.2 Caso 2 - Sem acesso aos sistemas da organização ..... 98**

**5.3.3 Caso 3 – Não ingressa nas pastas de rede ..... 104**

**5.3.4 Caso 4 – Intranet da organização não atualiza conteúdo ..... 111**

5.4 CONSIDERAÇÕES FINAIS ..... 115

**6 CONCLUSÃO ..... 117**

6.1 REVISÃO DO TRABALHO ..... 117
6.2 PRINCIPAIS CONTRIBUIÇÕES ..... 118
6.3 TRABALHOS FUTUROS ..... 118
**REFERÊNCIAS ..... 120**
**Apêndice A – Questionário Aplicado ..... 124**
**Apêndice B – Código Jcolibri ..... 127**
**Apêndice C – Regras do Expertsinta ..... 134**

# 1 INTRODUÇÃO

Neste capítulo introdutório são abordadas a contextualização e as motivações que levaram à realização deste trabalho, juntamente com os objetivos propostos. A organização aqui ocorre da seguinte forma: na Seção 1.1, são contextualizados os problemas que serão discutidos e os aspectos que motivaram a realização desta dissertação; na Seção 1.2, são detalhados os objetivos; na Seção 1.3, é descrita a metodologia utilizada para o desenvolvimento do referido trabalho e, finalmente, na Seção 1.4, é apresentada a estrutura organizacional desta dissertação.

## 1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A Tecnologia da Informação (TI) vem se transformando em uma ferramenta estratégica para o desenvolvimento dos negócios nas empresas. Por muitos anos, algumas organizações conseguiam continuar os próprios negócios sem o apoio da TI. Porém, nos dias atuais, a TI tornou-se um fator primordial para se alcançar o sucesso (PINHEIRO, 2014). Uma das áreas da TI que demanda grandes investimentos e exige uma equipe cada vez mais especializada é a área de suporte, principalmente na atuação de controle de resoluções de problemas ocasionados por incidentes.

Problemas que geram a indisponibilidade dos serviços de TI causam prejuízos para as organizações. A interrupção inesperada que degrada o funcionamento padrão destes serviços é conhecida como incidentes (OGC, 2011a). Para se ter uma ideia do valor financeiro decorrente desses problemas, basta verificar a receita média perdida por hora de inatividade de um serviço de TI. Por exemplo, o serviço de vendas por cartão de crédito em uma organização pode chegar a milhões de dólares em perdas (MAGALHÃES; PINHEIRO, 2007).

Por outro lado, para uma maior eficiência no suporte aos serviços de TI para os clientes, muitas empresas passam a adotar ITIL (IT *Infrastructure Library*). A ITIL é uma biblioteca que apresenta um guia com as melhores práticas para o gerenciamento de serviços de TI mais utilizado no mundo. O objetivo do ITIL é fornecer orientações de como proporcionar os serviços de TI com qualidade para as organizações (PINHEIRO, 2014). Dentro da abordagem da ITIL, destaca-se o gerenciamento de incidentes, que tem como objetivo apresentar um processo de tomada de decisão geral, envolvendo a realização de atividades utilizando

procedimentos e técnicas para restaurar a operação normal do serviço o mais rápido possível (OGC, 2011a).

A equipe de suporte, responsável por reparar os serviços que sofreram incidentes, toma decisões e realiza ações frequentemente dentro de uma organização. O profissional do suporte, ao ser confrontado com um problema, necessita de um processo decisório para orientar e tornar a resolução mais eficiente. De acordo com Simon (1960), o processo de tomada de decisão geral contempla fases e atividades que abrangem desde a identificação, investigação e diagnóstico do problema até a implementação da solução e avaliação das decisões tomadas, passando pela procura e análise de alternativas.

Por sua vez, uma importante técnica da inteligência artificial, Raciocínio Baseado em Casos (RBC), também tem auxiliado o processo de tomada de decisão. RBC utiliza raciocínio por analogia, cujo objetivo é resolver um novo problema, relembrando uma situação anterior similar, chamada de caso, que consiste basicamente em problema e solução (WANGENHEIM; WANGENHEIM, 2003). Os casos representam o conhecimento acumulado do especialista, a partir de situações vivenciadas (PRENTZAS; HATZILYGEROUDIS, 2007).

A utilização de RBC para resolver problemas apresenta algumas vantagens. Por exemplo: (i) a aquisição do conhecimento acontece armazenando casos (experiências vivenciadas), diferenciando-se de outras técnicas que necessitam modelar todo o conhecimento do domínio inicialmente; e (ii) ao pesquisar por um problema na base de casos, são recuperados casos similares, apresentando-se soluções similares, que podem ser adaptadas gerando soluções aceitáveis (SHIU; PAL, 2004).

De acordo com Shiu e Pal (2004), o raciocínio por analogia é amplamente utilizado pelos seres humanos no seu cotidiano para a resolução de problemas. Na área profissional, funcionários experientes recorrem com frequência a soluções passadas e se necessário, adaptam-nas para resolverem novos problemas, tornando-os mais fáceis de solucioná-los (GENTNER; SMITH, 2012; OLIVEIRA; GREGHI; PIMENTEL 2008).

Pode-se utilizar o raciocínio por analogia com o uso do RBC para apoiar a resolução de problemas de incidentes, como por exemplo, “Serviço de e-mail da organização inoperante”. Uma busca em uma base de casos de incidentes de TI, com o intuito de resolver este novo problema, pode resultar na exibição de um ranking de vários casos, tendo cada caso uma série de informações, tais como: a descrição do problema do incidente, a solução a ser aplicada e os passos para implementá-la. As soluções do novo caso, bem como os procedimentos

estratégicos para alcançá-las, devem constituir uma base para apoiar soluções futuras e servir de aprendizagem para diversos usuários.

Outro tipo de raciocínio empregado para apoiar tomadas de decisões é o Raciocínio Baseado em Regras (RBR). Ele é uma técnica da inteligência artificial que atua como colaboradora na tomada de decisão em áreas específicas dominadas por especialistas humanos. O objetivo é condensar o conhecimento geral de um ou mais especialistas e utilizar este conhecimento armazenado para apoiar na resolução de problemas do usuário (SILVA, 2005).

A forma básica de se representar uma regra é: SE x ENTÃO y, em que x é a descrição de determinada situação, e y é a ação desencadeada como consequência. De acordo com Luger (2013), esta forma de representar o raciocínio é indicada principalmente para a tomada de decisões repetitivas e para o controle de ações e diagnósticos, o que tende a produzir uma conclusão em um tempo curto. Por exemplo, SE *cartucho de impressora vazio* ENTÃO *trocar cartucho*.

O raciocínio por regras pode auxiliar na resolução de problemas de incidentes. Ao se tomar o mesmo exemplo de problema de incidente descrito anteriormente, “Serviço de e-mail da organização inoperante”, pode-se executar um conjunto de regras pré-definidas para este problema, com o objetivo de se solucionar o incidente. As conclusões das regras podem prover uma solução contendo informações do que deve ser realizado para sanar o problema de incidente.

Por sua vez, a combinação destas duas técnicas, RBC e RBR, vêm sendo utilizada dentro da inteligência artificial para aprimorar soluções de problemas. De acordo com Eshete (2009), essa combinação melhora significativamente o processo de tomada de decisão, visto que o processamento das informações passa pelos dois raciocínios, fornecendo conhecimento e soluções mais precisas.

Uma vez que o RBC e RBR são importantes para apoiar tomadas de decisão e a utilização desses raciocínios favorece a solução de problemas de incidentes, a definição de quando e em qual momento deve-se aplicar etapas de cada um desses raciocínios para solucionar incidentes, guiando os decisores em um processo decisório sistemático, trata-se de um desafio.

Desta forma, acredita-se que a combinação de RBC e RBR pode contribuir para melhorar os procedimentos das atividades definidas pelo ITIL para resolução de incidentes em TI e, aditivamente, aprimorar a tomada de decisão do profissional de suporte.

## 1.2 OBJETIVO

O objetivo desta dissertação é desenvolver uma abordagem metodológica de apoio à decisão para resolução de incidentes de TI, de acordo com a ITIL, respeitando o processo de tomada de decisão de Simon, apoiado pela utilização e combinação das técnicas RBC e RBR. O propósito é auxiliar a tomada de decisão dos integrantes da equipe de suporte na resolução de incidentes de TI. Com a finalidade de atingir esse propósito, são almejados os seguintes objetivos específicos:

a) Aquisição de conhecimento sobre o domínio do processo de resolução de incidentes no ambiente de TI, através do ITIL, e sobre o processo de tomada de decisão, em especial o modelo de tomada de decisão de Simon, e das técnicas de apoio à decisão, tais como RBC e RBR;
b) Descrever a sequência de passos para guiar a tomada de decisão na resolução de incidentes de TI, respeitando o processo decisório de Simon apoiado pela utilização do raciocínio por analogia, com o uso do RBC;
c) Descrever a sequência de passos para guiar a tomada de decisão para a resolução de incidentes de TI no seu estágio inicial, respeitando o processo decisório de Simon e utilizando o raciocínio por regras com uso do RBR;
d) Apresentar a abordagem metodológica proposta de apoio à decisão para resolução de incidentes no ambiente de TI, integrando (ii) e (iii) acima;
e) Desenvolver um protótipo utilizando RBC e RBR para resolução de incidentes em TI em um ambiente organizacional, visando avaliar a abordagem metodológica proposta;
f) Avaliar a abordagem metodológica proposta através de estudo de caso.

O desenvolvimento da abordagem proposta envolve multidisciplinaridade nas áreas de: a) Sistemas de Informação, pois é necessária à adoção de um processo de tomada de decisão para guiar o profissional de suporte; b) Gestão de Serviços, quando da utilização do ITIL para entender o domínio de resolução de incidentes; e c) Inteligência Artificial, quando da aplicação dos raciocínios baseado em casos e baseado em regras para apoiar decisões.

## 1.3 METODOLOGIA

De uma forma geral, a metodologia aplicada para a construção deste trabalho foi a seguinte:

a) Revisão bibliográfica acerca do processo de tomada de decisão, gerenciamento de serviços de TI, principalmente relacionado aos incidentes de TI, e técnicas de inteligência artificial RBC e RBR, identificando os trabalhos relacionados;
b) Entrevista com profissionais da área de suporte de TI;
c) Identificação de cenários para a resolução de incidentes dentro de ambiente organizacionais;
d) Definição, de forma sistemática, dos passos para solucionar incidentes que, por sua vez, foi fundamentada na revisão bibliográfica, na análise do ambiente de suporte de organizações e nas informações obtidas com especialistas da área de TI;
e) Confecção da solução proposta através de uma abordagem metodológica para resolução de incidentes de TI;
f) Projeto para construção de um protótipo, utilizando raciocínio baseado em casos e raciocínio baseado em regras;
g) Implementação do protótipo para ratificar o uso da abordagem metodológica proposta;
h) Aplicação de estudo de caso para avaliar a abordagem metodológica proposta utilizando o protótipo construído.

Maiores detalhes sobre a metodologia utilizada para o desenvolvimento de partes específicas deste trabalho estão descritos ao longo desta dissertação.

## 1.4 ESTRUTURA DA DISSERTAÇÃO

A seguir, é apresentada a relação sumária dos capítulos estruturados nesta dissertação.

a) **Segundo Capítulo:** são discutidos conceitos e definições sobre resoluções de incidentes no ambiente de TI, definidos pela ITIL, bem como o modelo de processo de tomada de decisão de Simon e as técnicas da inteligência artificial RBR e RBC;

b) **Terceiro Capítulo:** são discutidos os trabalhos correlatos relacionados com o processo de tomada de decisão RBR e RBC, além de incidentes de TI de acordo com a ITIL;

c) **Quarto Capítulo:** é proposta a abordagem metodologia de apoio à decisão para a resolução de incidentes no ambiente de TI, de acordo com o ITIL, contemplando o processo de tomada de decisão de Simon, apoiado pela utilização e combinação do RBR e RBC;

d) **Quinto Capítulo:** é apresentado um estudo de caso com o objetivo de avaliar a abordagem metodológica proposta utilizando um protótipo implementado;

e) **Sexto Capítulo:** são discutidas as considerações finais sobre a pesquisa realizada e feita uma descrição das principais contribuições, além das sugestões para pesquisas futuras.

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados temas que embasam e respaldam teoricamente este trabalho. Entre os principais tópicos abordados estão: (i) processo de tomada de decisão; (ii) *Information Technology Infrastructure Library* (ITIL); e (iii) técnicas da inteligência artificial para apoio à decisão.

## 2.1 PROCESSO DE TOMADA DE DECISÃO

### 2.1.1 Conceitos de Tomada de Decisão

As decisões estão presentes no cotidiano de qualquer ser humano. Durante o percurso do dia estamos constantemente tomando decisões: qual roupa vestir, qual prato comer, qual bebida tomar, etc. Decisão sempre ocorre quando nos deparamos com mais de uma alternativa possível para um curso de ação, ou seja, quando podemos fazer algo de duas ou mais maneiras diferentes (CHIAVENATO, 2010).

De acordo com Chiavenato (2010), as pessoas dentro de uma organização tomam decisões a respeito de suas tarefas e de metas a serem alcançadas. A organização contempla um sistema de decisões de vários membros envolvidos, em que cada pessoa participa escolhendo e decidindo entre as alternativas apresentadas para cumprir os próprios objetivos.

Para Lachtermacher (2009), a tomada de decisão envolve a identificação de um problema e a seleção de uma linha de ação para resolvê-lo. Segundo Chiavenato (2010), tomar decisões é identificar e selecionar um curso de ação para lidar com um problema específico.

Um problema ocorre quando o estado atual de uma situação é diferente do desejado (LACHTERMACHER, 2009). De acordo com Lorrigio (2006), problema é algo que precisa de solução em uma situação inesperada, causada por desvio em relação aos objetivos organizacionais. Os problemas podem acontecer devido à existência de determinadas ocorrências que, a princípio, não são claramente percebidas.

Conforme Pianezzola (2006), problemas, em geral, devem ser modelados levando em consideração, principalmente, as situações nos ambientes em que se inserem.

Por sua vez, situação-problema é aquilo que se torna aparente, tornando-se evidente (manifesto). Em geral, as situações-problema devem ser descritas em termos do processo onde as percepções vivenciadas são realmente relevantes, provocadas pelo “mau” ou “não”

funcionamento de um sistema. Por exemplo: um paciente apresentando febre alta é uma situação-problema manifestada pelo mau funcionamento do sistema linfático.

O tomador de decisão enfrenta dois tipos principais de problemas, sendo que e a natureza desses problemas irá influenciar nos artifícios aplicados para se alcançarem as soluções. Esses dois tipos de decisões são chamados de programadas e não programadas (MONTANA, 2010).

(i) As decisões programadas caracterizam os problemas rotineiros e repetitivos, que são bem estruturados e compreendidos e se prestam a procedimentos e regras. Essas decisões, cada vez que são tomadas, são semelhantes com as outras vezes. Por exemplo: retirar um livro de uma biblioteca ou sacar dinheiro em caixa eletrônico. Na primeira vez em que esses problemas são resolvidos, será necessário avaliar a melhor forma de resolver o problema, mas no final teremos uma solução algorítmica que poderá ser aplicada sempre que o mesmo problema se repetir (MONTANA, 2010);

(ii) As decisões não programadas acontecem esporadicamente, muito raramente, e existem pouco precedentes para a tomada de decisão. Destinam-se aos problemas pouco compreendidos e estruturados e não se encaixam aos procedimentos rotineiros e repetitivos (MONTANA, 2010). Por exemplo: decidir sobre a redução do quadro de funcionários, mudança na estrutura física da sede da empresa.

A abordagem metodológica proposta nesta dissertação é baseada nas decisões programadas. Sendo que a primeira vez que resolvemos um determinado incidente é preciso avaliar a melhor forma de resolvê-lo, mas na próxima vez que o mesmo incidente ocorrer é possível aplicar a solução aprendida.

### 2.1.2 Processo de Tomada de Decisão

O processo de tomada de decisão é o caminho mental utilizado para chegar a uma decisão que envolve uma sequência de passos em diferentes fases que orientam as ações a serem tomadas para encontrar uma solução que resolva ou não um problema (TURBAN, 2013).

De acordo com Chiavenato (2010), alguns dos principais elementos presentes no processo decisório são:

a) **Tomador de decisão**: indivíduo que faz uma opção entre várias alternativas;

b) **Objetivos**: são os resultados que o tomador de decisão deseja alcançar com suas ações;

c) **Preferências**: são os critérios para fazer a escolha;

d) **Estratégia**: curso de ação escolhida para melhor atingir os objetivos;

e) **Resultado**: é a consequência resultante da estratégia implementada.

Quando o tomador de decisão se vê diante de uma situação em que uma decisão deve ser tomada, duas opções básicas se apresentam: a) usar apenas a intuição ou b) realizar um processo de modelagem e estudar mais profundamente o problema (LACHTERMACHER, 2009).

De acordo com Lachtermacher (2009), diversas vantagens podem ser citadas no momento em que o decisor utiliza um processo de modelagem para tomada de decisão:

a) Impõe os decisores a tornarem transparentes seus objetivos;

b) Força a identificação e o armazenamento de diferentes decisões e os seus relacionamentos;

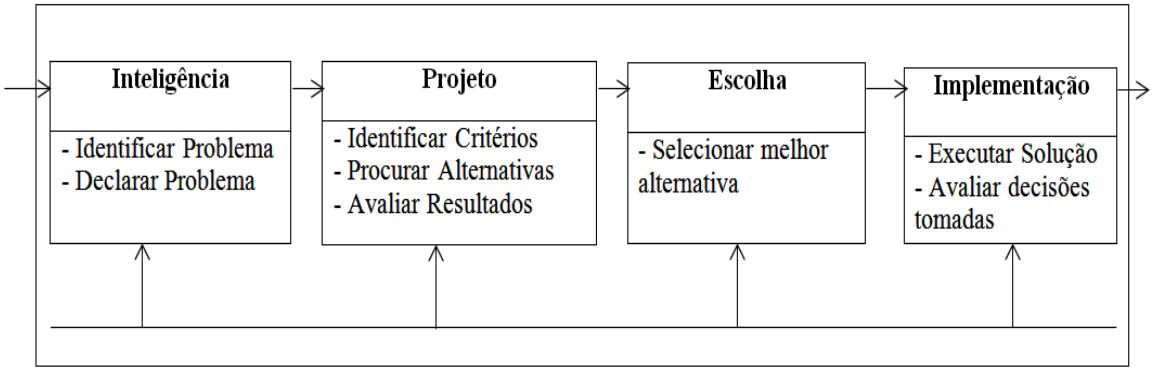
c) Forçam a identificação das variáveis a serem incluídas;

d) Permitem a comunicação de suas ideias e seu entendimento para facilitar o trabalho em grupo.

Segundo Pereira, Lobler e Simonetto (2010), entre os modelos de processo decisório, existentes na literatura da área, a maioria guarda em comum características semelhantes ao modelo descrito por Simon. Este modelo decisório, segundo Turban (2013), ainda é o mais conciso e com completa caracterização de processo de decisão racional.

O modelo clássico de Simon (SIMON, 1960), descreve o processo de tomada de decisão composto por três fases principais: inteligência, projeto e escolha. Podendo ser incluída a fase de implementação (Figura 1).

Figura 1 - Fases do processo decisório de Simon




Fonte: Adaptação de Simon (1960).

O processo de tomada de decisão começa com a fase denominada Inteligência, que tem o objetivo de identificar e entender os problemas que venham a acontecer na organização. Nesta fase, o tomador de decisão examina uma situação, identifica e define o problema (TURBAN, 2013).

Na segunda fase, chamada de projeto, os tomadores de decisão utilizam um modelo de representação que simplifica o problema com o objetivo de procurar e avaliar possíveis soluções. De acordo com Turban (2013), o modelo é então validado, e os tomadores de decisão especificam os critérios para a avaliação de soluções alternativas identificadas.

A fase escolha, a terceira, envolve escolher uma das alternativas com a solução que a fase de projeto ofertou, não sendo descartado que a solução pode ser composta da combinação de mais de uma alternativa. Essa fase envolve selecionar uma alternativa com a consequente solução que pareça viável para o problema (TURBAN, 2013).

A aplicação da solução escolhida é tratada na última fase, denominada Implementação. Segundo Turban (2013), uma aplicação bem sucedida resulta em solucionar o problema. Ao final, uma avaliação das tomadas de decisão é realizada.

Deve ser observado que ocorre um contínuo fluxo de informações da fase inteligência até a fase implantação, mas, em qualquer fase, pode haver um retorno para uma fase anterior (TURBAN, 2013). Assim, o processo de tomada de decisão é visto em um ciclo.

## 2.2 A BIBLIOTECA ITIL (INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY)

### 2.2.1 Gerenciamento dos Serviços de TI e a ITIL

Por muitos anos as organizações detinham seus negócios com pouco apoio da Tecnologia da Informação (TI). A realidade hoje é diferente: TI é um fator fundamental para o sucesso das organizações. Existe alguns negócios que seriam impossíveis tornar-se realidade sem o apoio dos serviços de TI. Como exemplo, o sistema de controle de contas bancárias sem o apoio do serviço de banco de dados (PINHEIRO, 2014).

O Gerenciamento dos Serviços de TI é um instrumento pelo qual a área de TI pode iniciar uma postura proativa para atender às necessidades da organização, contribuindo para diminuir ou evitar a indisponibilidade destes serviços. O objetivo é alocar adequadamente os recursos disponíveis e gerenciá-los, preservando a entrega e a operação dos serviços de TI (MAGALHÃES; PINHEIRO, 2007).

Para isso, vem sendo adotada a biblioteca ITIL, que representa um conjunto de melhores práticas para o gerenciamento de serviços de TI mais utilizada no mundo (PINHEIRO, 2014). A ITIL foi criada pelo esforço do governo Britânico durante a década de 1980 para documentar como as organizações de sucesso abordam o gerenciamento de serviços de TI, publicando documentações com as melhores práticas, em forma de livros, desde o início dos anos 1990 (MENKEN, 2015).

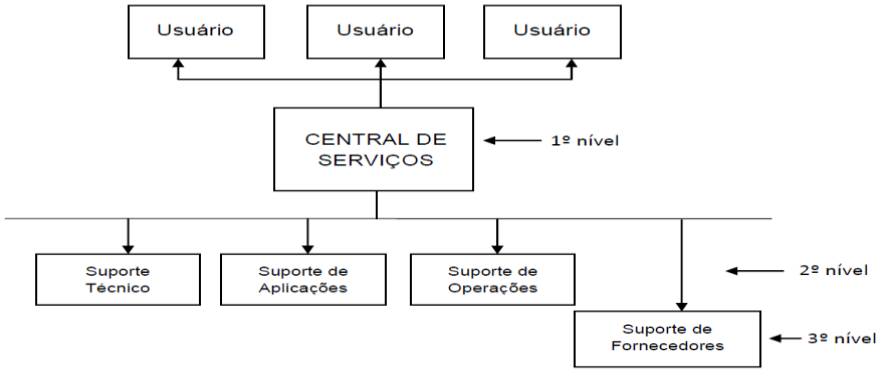
### 2.2.2 Gerenciamento de Incidentes de TI e a Análise do Ambiente

Dentro do que é documentado na ITIL, destaca-se o gerenciamento de incidentes, do livro Operação de Serviço. Gerenciamento de incidentes é altamente visível para o negócio, e por isso, mais fácil de demonstrar valor no ambiente de TI (OGC, 2011a).

De acordo com OGC (2011a), incidente é qualquer ocorrência detectada por um usuário ou ferramenta de monitoramento que não faz parte da operação padrão de um serviço, quando acontece algo inesperado que degrada ou interrompe os serviços. O Gerenciamento de Incidentes tem como principal objetivo a resolução do incidente, ou seja, restaurar a operação normal do serviço o mais rápido possível e minimizar os efeitos adversos causados nas operações da organização.

A resolução do incidente é realizada por equipes de suporte, separados em níveis. O 1º nível é representado pela Central de Serviços. Ela é frequentemente o primeiro ponto de contato dos usuários que, ao utilizarem um serviço de TI, percebem o funcionamento diferente do previsto (MAGALHÃES; PINHEIRO, 2007). O 2º nível faz-se representar por equipes de conhecimentos especializados, tais como: suporte técnico, suporte de aplicações e suporte de operações, e o 3º nível pode ser integrado por suporte de fornecedores terceirizados (Figura 2). A quantidade de níveis pode variar dependendo do tamanho do departamento de TI (OGC, 2011a).

Figura 2 - Níveis de suporte para resolução de incidentes de TI




Fonte: Autor deste trabalho (2015).

A Central de Serviços, 1º nível do suporte (Figura 2), é a responsável pelo tratamento das ocorrências de incidentes, podendo ser acionada através de chamadas telefônicas, sistemas de abertura de ocorrências web ou através de notificações automáticas de softwares que monitoram a infraestrutura de TI (OGC, 2011a).

O Suporte Técnico, que pertence ao 2º nível do suporte (Figura 2), refere-se à equipe responsável por resolver incidentes no domínio relacionado à infraestrutura de TI. Um exemplo de incidente neste nível: um servidor com porta de rede sem comunicação. Entre os objetivos da equipe está o uso de habilidades técnicas para diagnosticar e resolver rapidamente qualquer falha técnica ocorrida com a infraestrutura de TI (OGC, 2011a).

O Suporte de Aplicações (Figura 2), pertencente ao 2º nível do suporte, corresponde à equipe responsável por resolver os incidentes no domínio dos softwares (aplicativos) existentes na organização, por exemplo, a aplicação de gestão empresarial não realizou uma operação corretamente.

A equipe de Suporte de Operações de TI (Figura 2), que pertence ao 2º nível de suporte está relacionada ao domínio das atividades do dia a dia no ambiente de TI, gerenciando e mantendo a infraestrutura de TI em bom funcionamento (OGC, 2011a). Esta equipe fica responsável por resolver incidentes no domínio especificado. Por exemplo, a operação de backup dos arquivos apresentou falha ao ser executado.

O 3º nível do suporte (Figura 2) pode ser terceirizado pela organização, sendo representado por fornecedores, fabricantes ou mantedores de softwares e hardwares (OGC, 2011a). Como exemplo, as empresas fornecedoras de circuito de dados, computadores, sistemas, que podem ser acionadas para a resolução de incidentes.

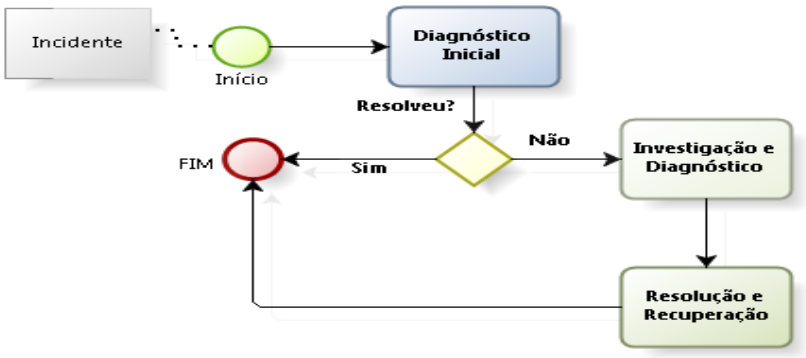
### 2.2.3 Processo, Métodos e Técnicas para Resolução de Incidentes de TI

O processo para a resolução de incidentes é extremamente reativo, entrará em atuação a partir dos incidentes levantados por usuários, com a possibilidade de ferramentas de monitoramento da infraestrutura também acusarem incidentes (PINHEIRO, 2014).

O primeiro passo para o processo de resolução de incidente é identificar o incidente que foi levantado por usuário, seja ele um (funcionário, cliente) ou uma ferramenta de monitoramento.

Identificado o incidente, as principais atividades para o processo de resolução, de acordo com o ITIL são: a) diagnóstico inicial; b) investigação e diagnóstico; e c) resolução e recuperação (Figura 3):

Figura 3 - Atividades para resolução de incidentes de TI baseado no ITIL




Fonte: Adaptado de (OGC, 2011a).

A Central de Serviços faz o suporte de 1º nível e efetua, de forma exclusiva, a seguinte atividade:

- **Diagnóstico Inicial**: A Central de Serviços é quem deve realizar o diagnóstico inicial, com o intuito de encontrar a falha (problema) e solucioná-la; tentará descobrir todos os sintomas do incidente, além de entender exatamente o que deu errado e como corrigir esse erro (OGC, 2011a). É nesta atividade que o script de diagnósticos contendo o conhecimento do especialista - o conhecimento geral sobre um assunto - e informações de erros conhecidos, será de extrema importância para permitir o diagnóstico precoce e preciso. Se com esta atividade, a Central de Serviços tiver êxito, o incidente é resolvido imediatamente e ocorre o fechamento do incidente (Figura 3). Caso não consiga resolver prontamente, mas tenha a perspectiva de fazê-lo sem a necessidade de escalar para outro nível, com conhecimento técnico mais profundo, a Central de Serviços deve

informar ao usuário sobre as intenções assumidas e dar ao usuário o número de referência do incidente, além de tentar encontrar uma solução realizando as atividades de (i) investigação e diagnóstico e (ii) resolução e recuperação (OGC, 2011a) (Figura 3).

As atividades a seguir podem ser realizadas por qualquer nível do suporte:

- **Investigação e Diagnóstico**: Esta atividade inclui investigar e diagnosticar o que está errado e devem-se documentar todas as atividades envolvidas na resolução, gerando-se um histórico. Além disso, deve-se identificar qualquer evento que possa ter mudado o bom funcionamento do serviço e consultar a base de conhecimento para reconhecer incidentes anteriores e documentações que venham ajudar a diagnosticar o incidente (OGC, 2011a). Segundo o autor referenciado, esta investigação é provável que inclua ações como: (i) estabelecer exatamente o que deu errado; (ii) identificar qualquer evento que possa ter provocado o incidente, como, por exemplo, alguma ação do usuário, alguma falha em um serviço; e (iii) buscar conhecimento em um banco de dados de conhecimento, procurando por ocorrências de incidentes anteriores.
- **Resolução e Recuperação**: Quando uma solução for identificada, esta deve ser aplicada e testada para garantir a recuperação do serviço. Os integrantes das equipes de suporte podem implementar uma resolução de forma centralizada, por exemplo, reiniciar um servidor ou remotamente usando um software, assumir o controle do computador do usuário para diagnosticar e executar uma solução (OGC, 2011a).

### 2.2.4 Gerenciamento de Configuração

Para auxiliar todos os níveis de suporte, em especial, na atividade investigação e diagnóstico, um importante gerenciamento descrito no ITIL, deve estar relacionado, que é o de configuração.

O gerenciamento de configuração é o responsável por controlar os itens de configurações e manter informações de configurações precisas sobre a infraestrutura e os serviços de TI (OGC, 2011b).

Todo serviço ou sistema, incluindo qualquer *hardware, software* e documentação é um item de configuração (IC) (OGC, 2011b). Alguns exemplos de IC são: servidores, roteadores, switches, serviços, impressoras, estação de trabalho (*desktop*). Cada incidente está conectado com um IC (PINHEIRO, 2014).

## 2.3 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA APOIO À DECISÃO

### 2.3.1 Raciocínio Baseado em Casos

O Raciocínio Baseado em Casos (RBC), do inglês *Case-based Reasoning*, apresenta-se como uma técnica da Inteligência Artificial para a solução de problemas e para o aprendizado com base em experiências passadas, chamado de casos, que são constituídos, principalmente, por problema e solução. Estas experiências (casos) irão ajudar a tomar decisões e resolver problemas futuros (RICHTER E WEBER, 2013). Em RBC, o tomador de decisão lembra-se de situações anteriores semelhantes à atual e as utiliza para ajudar a resolver um novo problema (KOLODNER, 1993). Ou seja, RBC pode auxiliar nas tomadas de decisões programadas.

No início dos estudos sobre RBC, são encontrados nos trabalhos de Schank e Abelson, em 1977, aspectos sobre a importância da memória dinâmica e o papel central que a lembrança de situações anteriores (casos, episódios) e o registro de situações padrão por meio de scripts têm no aprendizado e na resolução de problemas. Influenciado por estas pesquisas, Janet Kolodner, da Universidade de Yale, lançou, em 1983, o primeiro sistema utilizando raciocínio baseado em casos, o CYRUS, que era um sistema de perguntas e respostas construído com o conhecimento das várias viagens e reuniões de um ex-secretário do Estado Americano (AAMODT, 1994).

Conforme Richter e Weber (2013), RBC é uma metodologia para solucionar problemas. A tecnologia de RBC pode ser vista em duas formas: a) uma metodologia para modelar o pensamento humano, e b) uma metodologia para construir sistemas inteligentes (WANGENHEIM; WANGENHEIM, 2003).

Para solucionar problemas utilizando RBC, é aplicado o raciocínio por analogia, que consiste em comparar os sintomas apresentados pelo novo problema com um problema anterior conhecido ou uma experiência similar vivenciada. A solução do problema anterior é então utilizada, e se necessária, é adaptada para resolver o novo problema. Há situações em que um problema anterior é idêntico ao novo problema, então a solução do problema anterior pode ser aplicada ao problema novo. Se os problemas não forem idênticos, uma fase de adaptação ocorre (SHIU; PAL, 2004).

De acordo com Wangenheim e Wangenheim (2003), a metodologia do RBC é muito atraente porque é semelhante ao comportamento humano ao resolver um problema. Essa abordagem é conveniente e eficaz porque muitas vezes não é necessário se ter um

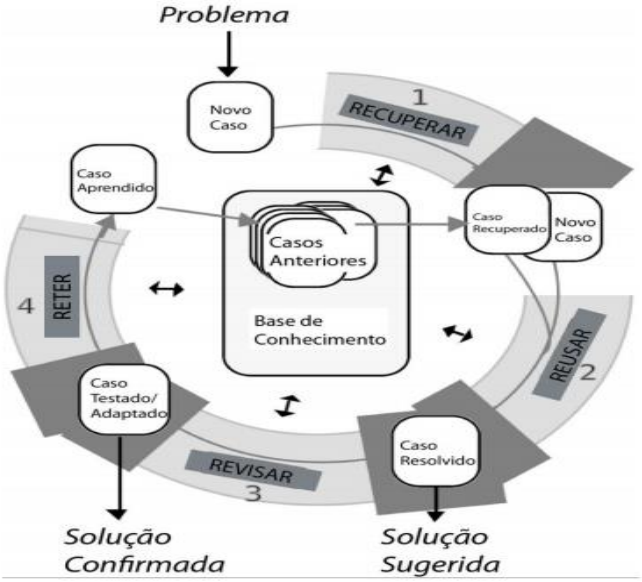
conhecimento amplo do domínio do problema. Por exemplo, um juiz reforça os seus argumentos com jurisprudências semelhantes: “Esse caso deve ser decidido como no caso empresa A versus Silva”.

O conhecimento em RBC é representado através de casos, que é uma experiência anterior que foi aprendida e armazenada na base de casos, e pode ser utilizado para solucionar um problema futuro (AAMODT, 1994). Ao ser dado um problema novo, problemas com descrições similares ao novo problema, são recuperados na base de casos e a partir da solução dos problemas similares recuperados, é apresentada uma solução derivada para o problema atual (SHIU; PAL, 2004). Conforme Wangenheim e Wangenheim (2003), problemas similares possuem soluções similares. Detalhes sobre o funcionamento de RBC serão apresentados na próxima seção.

#### 2.3.1.1 Ciclo Básico de Processamento em RBC

No início dos estudos sobre RBC, uma variedade de arquiteturas foi utilizada para desenvolver os sistemas de raciocínio baseado em casos (RICHTER; AAMODT, 2005). No ano de 1994, Aamodt e Plaza estabeleceram um modelo que passou a ser o modelo aceito para o desenvolvimento destes sistemas (Figura 4).

Figura 4 - Ciclo básico do RBC




Fonte: Adaptado de Shiu e Pal (2004).

O modelo é constituído de 4 (quatro) fases, conhecidas como “os quatro R’s”, sendo estes (AAMODT; PLAZA, 1994):

a) **Recuperar:** os casos mais semelhantes de uma base de casos são recuperados;
b) **Reusar:** o conhecimento do caso recuperado é reusado para resolver o problema;
c) **Revisar:** a solução proposta é revisada;
d) **Reter:** a experiência/caso (ou partes desta experiência) para solucionar futuros problemas é retida em uma base de casos.

O modelo estabelecido para o RBC proposto por Aamodt e Plaza, engloba um ciclo de raciocínio contínuo chamado Ciclo básico RBC (Figura 4). O funcionamento do ciclo é iniciado quando se descreve um problema, sendo definido assim um novo caso. Ao receber este novo caso, a base de conhecimento contendo casos anteriores é pesquisada para recuperar casos anteriores resolvidos cuja descrição do problema seja similar ao do novo caso informado. A expectativa é que a solução de um caso resolvido recuperado poderá ser reusada para o novo caso (AAMODT, 1994; WANGENHEIM E WANGENHEIM, 2003).

Seguindo o ciclo (Figura 4), a solução sugerida do caso resolvido mais similar é reusada para o novo caso, e se necessário, a solução sugerida será adaptada para satisfazer completamente o problema atual. Em seguida a solução gerada é revisada e testada no ambiente real, efetuando-se os ajustes necessários, tornando-se assim uma solução confirmada (AAMODT, 1994; WANGENHEIM; WANGENHEIM, 2003).

Toda vez que um problema é resolvido, o caso aprendido pode ser retido na base de casos, tornando-o disponível para solucionar problemas futuros (AAMODT, 1994; WANGENHEIM; WANGENHEIM, 2003).

Este ciclo raramente ocorre sem a intervenção humana (WATSON; MARIR, 1994). Segundo Zimmermann (2002), isto acontece porque a maioria dos sistemas de RBC aplica somente a fase de recuperar casos. As fases reusar, revisar e reter geralmente serão realizadas pelo tomador de decisão.

#### 2.3.1.2 Representação de Caso

Os casos para os sistemas de RBC são fundamentais para seu funcionamento. É necessário estruturar o seu conteúdo para serem armazenados e posteriormente recuperados. Conforme Shiu e Pal (2004), os casos em uma base de casos podem representar diferentes tipos de conhecimento e ser armazenados em diferentes formatos de representação. A forma como um caso é estruturado facilita a sua recuperação quando consultado.

Um caso representa o conhecimento primário que é estruturado ao se combinar a caracterização do problema com as ações associadas à resolução desse caso (LOPES; SANTOS, 2009).

De acordo com Richter e Weber (2013), RBC usa os casos para resolver problemas. Portanto, é necessário que se tenha solução no caso que informe como um problema é resolvido. O caso deve ser composto, tipicamente, de dois elementos principais: problema e solução, existindo a possibilidade de um terceiro componente adicional, o resultado. Conforme Wangenheim e Wangenheim (2003), estes três elementos têm as seguintes características:

a) **Problema** – Descreve o problema (e/ou a situação-problema) e deverá possuir informações suficientes para que seja possível julgar a aplicabilidade de um caso à nova situação;
b) **Solução** – Postula a solução derivada para o problema mencionado. Descreve uma ação ou um conjunto de passos de raciocínio a serem seguidos para tomada de decisão e resolução do problema;
c) **Resultado** – Especifica o que ocorreu como consequência de se executar a solução proposta. Inclui uma avaliação de sucesso ou insucesso da solução e/ou se as expectativas foram atingidas. Uma solução avaliada com insucesso é também uma informação valiosa para que outros tomadores de decisão evitem implementá-la (RICHTER; WEBER, 2013).

Existem diversas opções de formato para se representar um caso. Alguns exemplos destes formatos são: modelos de banco de dados, frames, objetos, redes semânticas, etc. (SHIU; PAL, 2004).

**Índices**

Os casos no RBC podem ser indexados, e para isso são atribuídos índices para esses casos, com o objetivo de fazer a recuperação de forma fácil e rápida. De acordo com Shiu e Pal (2004), a escolha dos índices é importante para permitir a recuperação do caso adequado ao momento certo. A indexação tem a função de orientar a similaridade entre casos, e a medida de similaridade indica o grau de semelhança entre dois casos, já os índices indicam os casos na base de casos que são mais similares a um novo caso apresentado (LOPES, 2009).

Para um sistema de RBC, um índice considerado bom é aquele que consegue ser, ao mesmo tempo, distintivo e não unívoco (SHIU; PAL, 2004). Por exemplo: o atributo que define a quantidade de quartos de uma casa é um bom índice. Isto porque a quantidade de

quartos diferencia as casas (distintivo), e esta informação também não representa apenas uma ocorrência, ou seja, existe um conjunto de casas com a determinada quantidade de quartos (não unívoco).

De acordo com Kolodner (1993), um índice de qualidade deve:

- Prever a futura utilização da informação contida no caso para solucionar diferentes situações-problemas;
- Endereçar as similaridades que são úteis entre os casos;
- Ser abstrato o suficiente para tornar o caso útil em situações diferentes;
- Ser efetivo o suficiente para facilitar o seu reconhecimento em futuras situações.

Estes índices serão utilizados no cálculo da similaridade entre os casos na fase de recuperação do ciclo do RBC, como veremos na seção seguinte.

**Recuperação dos casos**

A recuperação de casos tem o objetivo de encontrar casos similares ao consultado e com solução útil para o problema ou situação-problema. Para realizar a recuperação, é confrontada a descrição do problema atual com os dos problemas anteriores, aplicando-se uma medida de similaridade.

Um conjunto de subtarefas deve ser realizado para o processo de recuperação de casos. São estas (WANGENHEIM, 2003; AAMODT, 1994):

a) **Assessoramento na consulta**: objetiva a formulação de uma consulta representada por um conjunto de descritores relevantes da situação-problema ou problema atual;
b) **Casamento:** identifica casos suficientemente similares à consulta para solucionar o problema atual;
c) **Seleção:** identifica o melhor caso entre os selecionados no casamento. Nesta subtarefa, os casos recuperados com o casamento são ordenados de acordo com um critério de seleção. A finalização ocorre através da seleção do mais similar.

Existem diferentes métodos para se realizar a recuperação dos casos. Assim, entre os existentes, um dos mais estudados e utilizados é o algoritmo do vizinho mais próximo (*Nearest Neighbour*), que contempla cálculos de similaridade local e global entre o caso com o problema atual e os casos da base de casos (SHIU; PAL, 2004). O algoritmo *Nearest Neighbour* será detalhado nesta dissertação, pelo fato de o utilizarmos em nossa implementação.

Os cálculos da similaridade local e global são realizados da seguinte forma: inicialmente é calculada a similaridade com a função que informa o quão semelhantes são os valores para determinado atributo, devendo-se associar um peso para cada atributo, de acordo com sua relevância. Em seguida é feito o cálculo da similaridade global com a média ponderada das similaridades locais.

A similaridade é dada pela equação (Equação 2.1):

$$Sim(C,R) = \frac{\sum_{i=1}^{n} w_i * f(C_i R_i)}{\sum_{i=1}^{n} w_i}$$

**Equação 2.1**

Em que:

*C* = caso apresentado;

*R* = caso recuperado da base;

*n* = número de atributos no caso;

*i* = i-ésimo atributo;

*w* = peso atribuído ao i-ésimo atributo;

*f* = função que calcula a similaridade local entre os casos C e R para o i-ésimo atributo;

*Sim* = função que calcula a similaridade global.

Existem diversas funções para o cálculo da similaridade local. Dentre estas, as principais são (SHIU; PAL, 2004):

- **Distância Euclidiana**: é determinada pela medida da distância entre dois objetos no espaço euclidiano. Esta distância é calculada a partir da raiz quadrada do quadrado da diferença aritmética entre duas coordenadas no espaço euclidiano refinado pelo peso. O peso nessa equação indica a importância do atributo no caso. Formalmente, o cálculo se dá como exibido na Equação 2.2. Sendo $d^w_{pq}$ a distância euclidiana entre as coordenadas p e q, ajustada pelo peso w.

$$d^w_{pq} = \sqrt{w^2(x_p - x_q)^2}$$

**Equação 2.2**

- **Similaridade do cosseno:** É a medida de similaridade entre 2 (dois) vetores, e essa medida é dada pelo cosseno do ângulo entre eles. É normalmente usada em recuperação de informação para detectar documentos textuais a partir de palavras chaves de consulta, a similaridade é representada pela Equação 2.3 a seguir.

$$Sim(A,B) = \frac{A * B}{\sqrt{|A|} * \sqrt{|B|}}$$

**Equação 2.3**

- **Coeficiente de *Jaccard*:** calcula a similaridade entre conjuntos, dividindo a quantidade de elementos da interseção pela quantidade de elementos união. A equação 2.4 descreve essa função.

$$Sim(A,B) = \frac{A \cap B}{A \cup B}$$

**Equação 2.4**

A recuperação dos casos funciona de forma diferente dos bancos de dados que recuperam registros com informações idênticas ao consultado. A recuperação dos casos em RBC retorna casos que podem possuir informações que não casem perfeitamente ou faltantes. Conforme Wangenheim e Wangenheim (2003), a recuperação dos casos em RBC não será realizada em uma filosofia tudo ou nada, como ocorre na busca em um banco de dados.

Após a recuperação dos casos mais similares a próxima fase do RBC é o reuso.

**Reuso**

A fase de reuso do ciclo RBC sugere uma solução para um novo problema a partir das soluções dos casos recuperados. Consiste em reutilizar a solução do caso recuperado para solucionar o novo problema, realizando adaptações, se necessário (MÁNTARAS, 2005).

Esta adaptação pode ser realizada com a utilização de um dos métodos a seguir (SHIU; PAL, 2004):

a) **Reinstanciação**: é copiada na íntegra a solução de um dos casos recuperados para o novo problema;
b) **Substituição**: através dos casos recuperados, é aplicada a substituição de valores dos atributos, de modo que se tratem as diferenças significativas e atenda-se às necessidades do novo caso;
c) **Transformação**: a solução é derivada de regras e particularidades dos requisitos do novo caso. Esse método é utilizado comumente, quando a substituição não pode ser implementada.

Ao finalizar a adaptação, a solução será testada no ambiente real, objetivando verificar se a solução desenvolvida vai solucionar o problema atual. Esta etapa é efetuada na atividade a seguir.

**Revisão**

Logo quando a solução é gerada, na fase de reuso, ela é testada no ambiente real para verificar se a reutilização dessa solução foi executada com sucesso. Caso apresente falha, serão efetuados os reparos necessários (MÁNTARAS, 2005). Esta fase de revisão consiste de duas tarefas (WANGENHEIM; WANGENHEIM, 2003):

a) avaliar a solução: a solução gerada pelo reuso é avaliada criteriosamente. Se implementada com sucesso, é dado seguimento para a próxima fase do ciclo;
b) reparar as falhas: a solução gerada, apresentando falha na própria implementação, será reparada, utilizando-se conhecimento específico sobre o domínio de aplicação.

Ao se efetuar a avaliação da solução e, se necessário, a reparação da solução, passa-se à última fase do Ciclo RBC, a fase reter.

**Reter**

A retenção de casos é o processo de acrescentar o conhecimento de um novo caso à base de casos, tornando o sistema de RBC um solucionador poderoso, com o passar do tempo (WANGENHEIM ; WANGENHEIM, 2003). Esta fase é caracterizada por três etapas. Ei-las:

a) **extração de conhecimento**: extrair do RBC as informações retidas de como um caso foi solucionado. Estas informações contêm o problema e a solução do caso, indicando como solucionar novos casos similares.
b) **indexação**: nesta etapa são decididos quais índices serão utilizados para o armazenamento do caso, o objetivo é facilitar a recuperação dos casos e determinar como o novo será estruturado para consulta.
c) **integração**: etapa final da retenção de novos casos. Nela ocorre o armazenamento do caso.

### 2.3.1.3 Vantagens do RBC

Conforme Shiu e Pal (2004), algumas das principais vantagens do uso do RBC são:

a) **Diminuição de esforço na tarefa de adquirir de conhecimento**: A aquisição de conhecimento no RBC acontece através do armazenamento de experiências/casos, de modo que se elimina a necessidade de extrair o conhecimento de um especialista para montar o sistema.
b) **Evitar erros cometidos no passado**: como existe a possibilidade de armazenar informações de falha dos casos, é possível evitar falhas que já foram cometidas.
c) **Flexibilidade na modelagem do conhecimento**: diferente de outras técnicas que necessitam modelar todo o conhecimento do domínio para o uso do RBC, este se utiliza de experiências passadas para modelagem.
d) **Aprendendo com o tempo**: à medida que novos problemas vão surgindo, soluções são geradas, testadas e armazenadas. Novos casos serão adicionados na base de casos, aumentado à racionalidade do sistema e, consequentemente, o grau de refinamento e sucesso das soluções.
e) **Raciocínio com dados incompletos ou imprecisos**: os casos recuperados no RBC não são exatamente idênticos ao que foi pesquisado, porém, se existir alguma medida de similaridade entre eles, é possível criar uma solução aceitável.
f) **Reflete o raciocínio humano**: em muitas situações, o ser humano utiliza o raciocínio por analogia, e o RBC reflete este tipo de raciocínio.

#### 2.3.1.4 Raciocínio Baseado em Regras

Raciocínio Baseado em Regras (RBR) é uma técnica da Inteligência Artificial (IA) usada para representar o conhecimento geral de um especialista de áreas específicas no âmbito de um sistema computacional. De acordo com Luger (2013), os especialistas humanos conhecem muito suas áreas de especialidade e conseguem obter desempenho de alto nível. RBR é uma das técnicas mais naturais e continua sendo amplamente utilizada para o desenvolvimento de sistemas especialistas (LUGER, 2013).

Conforme Guessoum (2012), os sistemas RBR têm mostrado um ótimo desempenho na modelagem de raciocínio que pode ser explicado e formalizado por regras. E as regras representam o conhecimento geral sobre um domínio.

De acordo com Friedman-Hill (2003), uma regra é um tipo de instrução ou comando aplicado a uma determinada situação. Por exemplo: (i) não mascar chiclete na escola, (ii)

onde há fumaça, há fogo, (iii) equipamento não funciona sem energia elétrica; entre outras que aprendemos ao logo da vida.

A forma básica de uma regra é representada por: *IF* <condições> *THEN* <conclusão>. Conforme Prentzas e Hatzilygeroudis (2007), a parte que contém as condições, também conhecida como premissas de uma regra, é conectada entre si com elementos lógicos, tais como, *AND*, *OR*, *NOT* etc. Formando, assim, uma função lógica. No momento em que as condições da regra forem satisfeitas, a conclusão é então acionada (carregada).

Por exemplo: uma regra para mascar chiclete, ver Figura 5. O *if* (se) contém as seguintes condições ou premissas: "*IF* eu estou na escola *AND* Estou mascando chiclete". A conclusão desta regra é: "*THEN* Jogar chiclete fora". O *end* (fim) finaliza o processo.

Figura 5 - Regra para mascar chiclete

**IF**
Eu estou na escola
**AND** Estou mascando chiclete
**THEN**
Jogar chiclete fora
**END**

Fonte: Adaptado de (Friedman-Hill, 2003).

A experiência vivida mostra que qualquer informação pensada de forma lógica pode ser codificada em regras (FRIEDMAN-HILL, 2003).

Segundo Prentzas e Hatzilygeroudis (2007), este tipo de raciocínio consegue resolver uma variedade de problemas e obtêm um desempenho de alta qualidade em diversos domínios, como, por exemplo, na Medicina, na Matemática e na Ciência da Computação. RBR é indicado para auxiliar em tomadas de decisões programadas e pode ser aplicada em:

a) Tomadas de ações repetitivas em que erros humanos devem ser evitados. Por exemplo, sempre verificar o prontuário médico antes de aplicar o medicamento. Se prontuário verificado, então aplicar medicamento;
b) Situações em que a base de conhecimento pode ser bem caracterizada por um especialista humano, mas não pode ser facilmente transmitida a outros seres humanos. Por exemplo, o conhecimento de um enfermeiro sobre os procedimentos necessários dentro de uma UTI pode ser caracterizado em uma base de conhecimento, mas não é facilmente transmitida para outros profissionais;

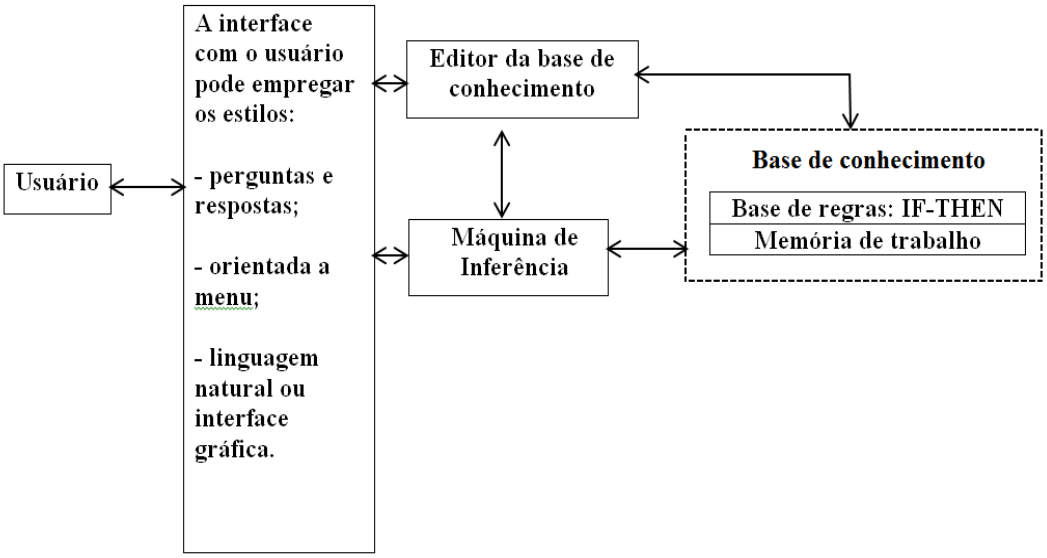
c) Controle de processos e diagnósticos. Por exemplo, se paciente apresenta temperatura acima de 38º graus então é diagnosticado paciente com febre.

#### 2.3.1.5 Componentes e arquitetura do Raciocínio Baseado em Regras

O funcionamento do RBR é estruturado em módulos e nos respectivos componentes (ver Figura 6). Os componentes formam o sistema de produção, o qual foi proposto por Newell e Simon, da Universidade de Carnegie-Mellon, no início dos anos de 1970 (ESHACH, 2006). Neste modelo, o conhecimento é separado do respectivo processamento.

O usuário (ver Figura 6) tem acesso ao sistema através da *interface com o usuário*, que tem o objetivo de facilitar a comunicação com o usuário e ocultar boa parte da complexidade, como a estrutura interna da base de regras. Estas interfaces podem empregar vários estilos, como exemplo, exibir uma interface gráfica com perguntas e alternativas para respostas. A interface que será empregada é uma relação entre as necessidades do usuário e os requisitos da base de conhecimento e do sistema de inferência que serão utilizados (LUGER, 2013).

Figura 6 – Arquitetura e componentes do RBR




Fonte: Adaptado de Luger (2013).

A *Base de conhecimento* contém o conhecimento de um domínio particular de aplicação. É composta pela base de regras, que retém o conhecimento pré-definido do especialista e os *dados específicos do caso*, que contêm: os fatos (dados de entrada fornecidos pelo usuário) e as conclusões das regras que foram acionadas. Estes dados podem ser mantidos em uma memória de trabalho (FRIEDMAN-HILL, 2003).

O *editor da base de conhecimento* tem como objetivos principais: (i) ajudar o programador a corrigir possíveis erros na execução do sistema, (ii) manter a sintaxe das regras funcionando de forma correta, e (iii) adicionar novo conhecimento na base (LUGER, 2013).

O principal componente do sistema baseado em regras é a máquina de inferência. Sua função é aplicar as regras, contidas na base de regras, com os dados fornecidos pelo usuário, que estão na memória de trabalho (FRIEDMAN-HILL, 2003). Mais detalhes sobre o funcionamento da máquina de inferência na seção seguinte.

#### 2.3.1.6 Resolvendo Problema com Raciocínio Baseado em Regras

A máquina de inferência é o módulo do Raciocínio Baseado em Regras responsável por realizar o "raciocínio lógico" que permite a resolução do problema. Existem duas formas de se aplicar as regras para se chegar a uma conclusão, que são as seguintes:

**Encadeamento reverso**

O sistema seleciona uma regra, carrega-lhe as premissas (condições) na memória de trabalho e tenta casar o objetivo (um fato a ser demonstrado como verdadeiro) com as conclusões das regras, buscando encontrar evidências que provem se tratar mesmo de um fato (LUGER, 2013). Por exemplo: um conjunto de regras para analisar problemas automotivos. Guiada por consulta ao usuário, contendo três regras simples, e mostradas na Figura 7. As regras serão executadas por ordem de prioridade.

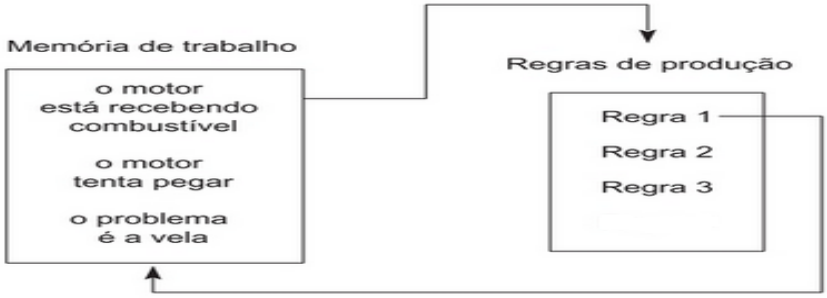
Figura 7 – Regras para sistema especialista de problemas automotivos

| | | |
|---|---|---|
| Regra1 **SE**<br>o motor recebe combustível e<br>o motor tenta pegar<br>**ENTÃO**<br>o problema é vela<br>**FIM** | Regra2 **SE**<br>o motor não pega e<br>as luzes não acendem<br>**ENTÃO**<br>o problema é bateria ou cabo<br>**FIM** | Regra 3 **SE**<br>Há combustível no tanque e<br>Há combustível no carburador<br>**ENTÃO**<br>o motor está recebendo combustível<br>**FIM** |

Fonte: Adaptado de Luger (2013).

Supondo que a Regra1 tenha prioridade maior do que as outras regras. A máquina de inferência então aciona e carrega na memória de trabalho esta Regra1 (ver Figura 8).

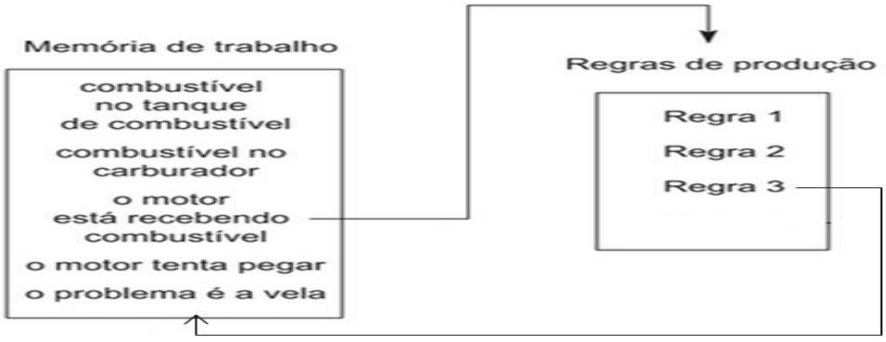
Figura 8 - Regra1 na memória de trabalho




Fonte: Adaptado de Luger (2013).

Existem duas premissas, ou condições, para a Regra1 que devem ser satisfeitas para que a conclusão seja acionada. Assim, o problema está decomposto em dois subproblemas: (i) descobrir se o *motor está recebendo combustível*, e (ii) se *o motor tenta pegar*. Podemos, então, acionar a Regra3, cuja conclusão casa com a primeira premissa da Regra1: *o motor está recebendo combustível*, e isso faz com que as premissas da Regra3 sejam colocadas na memória de trabalho (Figura 9).

Figura 9 - Regra3 acionada na memória de trabalho




Fonte: Adaptado de Luger (2013).

O cenário atual apresenta três premissas na memória de trabalho que não se relacionam com qualquer conclusão de regra: (i) *combustível no tanque de combustível*; (ii) *combustível no carburador*, e (iii) *o motor está recebendo combustível*. Dessa forma, o sistema baseado em regras passa a questionar diretamente o usuário sobre essas premissas. Uma vez o usuário tendo confirmado as três como verdadeiras, a conclusão é acionada: *o problema é v*ela. Isso conclui o diagnóstico do problema com sucesso, indicando que o carro não funciona por causa das velas.

**Encadeamento direto**

É o raciocínio guiado por dados, ainda o mais comumente encontrado. Parte-se dos fatos existentes e executa-se cada regra ativa existente, normalmente, por ordem de prioridade. Cada regra é executada apenas uma vez, e quando não houver mais regras ativas, a inferência termina (LUGER, 2013).

Por exemplo, ao se fazer uso do problema anterior, analisar problema automotivo e as regras da Figura 7. Podemos realizar o encadeamento direto da seguinte forma: se uma premissa de uma regra não for a conclusão de outra regra, então esse fato é considerado "questionável". Primeiro verificamos as premissas das regras para identificarmos qual premissa é "questionável". A premissa da Regra1 (*o motor recebe combustível*) não é questionável e vem a falhar, pois é a conclusão da Regra3 (*o motor está recebendo combustível*). O controle se move para a Regra2, que tem a sua premissa questionável (*O motor não tenta pegar).* Ao se supor que a resposta seja falsa, então é colocado na memória, *o motor tenta pegar* (Figura 10).

Figura 10 - Regra 2 na memória de trabalho

Fonte: Adaptado de Luger (2013).

A Regra2 falha, porque sua primeira premissa foi falsa; a Regra3 passa a ser considerada, visto que suas duas premissas são questionáveis: (i) *Há combustível no carburador;* e (ii) *Há combustível no tanque de combustível*. Suponha-se que as duas sejam verdadeiras, então estas são também colocadas na memória de trabalho (Figura 11).

Figura 11 - Regras 2 e 3 na memória de trabalho

Fonte: Adaptado de (Luger, 2013).

Com todas as regras consideradas (mas nem todas executadas), e a memória de trabalho com o novo conteúdo, voltam-se a considerar as regras na ordem pela segunda vez, voltando-se à Regra1. Quando a memória de trabalho (ver Figura 11) é casada com as duas premissas da Regra1: (i) *o motor recebe combustível;* e (ii) *o motor tenta pegar*, estas são correspondidas, e a conclusão da Regra1 (*o problema é vela*) então é disparada e colocada na memória de trabalho. O problema é exibido (*o problema é vela*) e o processamento das regras é finalizado.

### 2.3.1.7 Procedimentos para Construção de Base de Regras

De acordo com Spirandelli e Lennon (2011), para a construção de bases de regras, precisamos realizar quatro procedimentos:

a) **Definir as variáveis**: Como o próprio nome sugere, são variáveis que devem ser atribuídas a valores e podem ser do tipo univalorada (sim ou não) ou multivalorada, sendo que, neste último, deve-se especificar seus respectivos valores;
b) **Decretar os objetivos**: Deve-se selecionar a variável que será o objetivo, sendo que o objetivo de uma consulta feita por um especialista é encontrar a resposta para determinado problema;
c) C**riar as regras:** Seguem a estrutura SE... ENTÃO. Possuindo: (i) conectivos (não, e, ou); (ii) atributo: uma variável; (iii) operador, que define o tipo de comparação a ser realizada, como, por exemplo: =, >, <=, <>; (iv) valor, que é um item de uma lista a qual foi previamente criada e relacionada a um atributo (variável);
d) **Criar o questionamento do problema**: São perguntas, questionamento de problemas para que sejam feitas ao tomador de decisão. As respostas são associadas a cada

variável, com exceção da variável objetivo. Por exemplo, o questionamento “o cabo de rede está conectado ao computador?” com a resposta associada a variável “cabo”.

### 2.3.2 Combinando Raciocínio Baseado em Casos e Raciocínio Baseado em Regras

A utilização da combinação das técnicas RBC e RBR está se tornando cada vez mais presente para apoiar tomadas de decisões. Uma razão para esta presença é que os casos e as regras oferecem capacidades complementares. As regras representam conhecimentos gerais, enquanto os casos abrangem o conhecimento acumulado a partir de situações específicas vivenciadas. O RBR resolve problemas partindo do princípio das regras inseridas, enquanto o RBC usa informações armazenadas para tratar novas instâncias similares. Assim, a combinação deles acaba sendo natural e útil (PRENTZAS; HATZILYGEROUDIS, 2007).

Lee (2002), realizou um estudo comparativo entre estas duas técnicas de raciocínio, RBC e RBR. Sendo destacadas as seguintes características:

a) **Uniformidade**: a forma de adicionar conhecimento na base de conhecimento é uniforme;
b) **Simplicidade**: a representação do conhecimento é realizada de modo simples;
c) **Facilidade de aquisição de conhecimento**: facilidade de adicionar conhecimento na base de conhecimento;
d) **Aprendizado por experiência**: situações vivenciadas no cotidiano serão aprendidas;
e) **Facilidade de aplicar**: o conhecimento aplicado para resolver problema é próximo ao utilizado pelo ser humano, facilitando a aplicabilidade do conhecimento;
f) **Dificuldade para representar informação informal**: como as informações inseridas são uniformes, a representação de informação informal apresenta dificuldade;
g) **Dificuldade para aquisição de conhecimento**: dificuldade de inserção de conhecimento na base de conhecimento;
h) **Atualização**: atualização da base de conhecimento;
i) **alto custo na procura**: a base de conhecimento com muitos casos gera um alto custo na procura por casos similares;
j) **Problema para indexar caso**: é preciso indexar de forma precisa o caso para que seja efetuada a sua recuperação de forma eficiente.

Para apresentar o estudo comparativo foi construído o Quadro 1 a seguir, apresentando as técnicas de raciocínio e as suas respectivas vantagens e desvantagens, de acordo com as características descritas anteriormente.

Quadro 1 – Vantagens e desvantagens das técnicas de raciocínio baseado em casos e baseado em regras

| | **RBR** | **RBC** |
|---|---|---|
| **Vantagens** | 1. Uniformidade<br>2. Simplicidade | 1. Facilidade de aquisição de conhecimento<br>2. Aprendizado por experiência<br>3. Facilidade de aplicar |
| **Desvantagens** | 1. Dificuldade para representar informação informal<br>2. Dificuldade para aquisição de conhecimento<br>3. Atualização | 1. Alto custo na procura<br>2. Problema para indexar caso |

Fonte: Adaptado de Lee (2002).

De acordo com Lee (2002), o RBR representa o conhecimento do especialista, mas para converter este conhecimento em regras é necessário um tempo significativo. Afirma ainda que nem todo conhecimento pode ser convertido em regras. Sendo que até mesmo os especialistas, muitas vezes tiram conclusões das suas experiências ou de casos específicos em detrimento das regras. Por outro lado, o conhecimento em RBC é mais facilmente adquirido e atualizado.

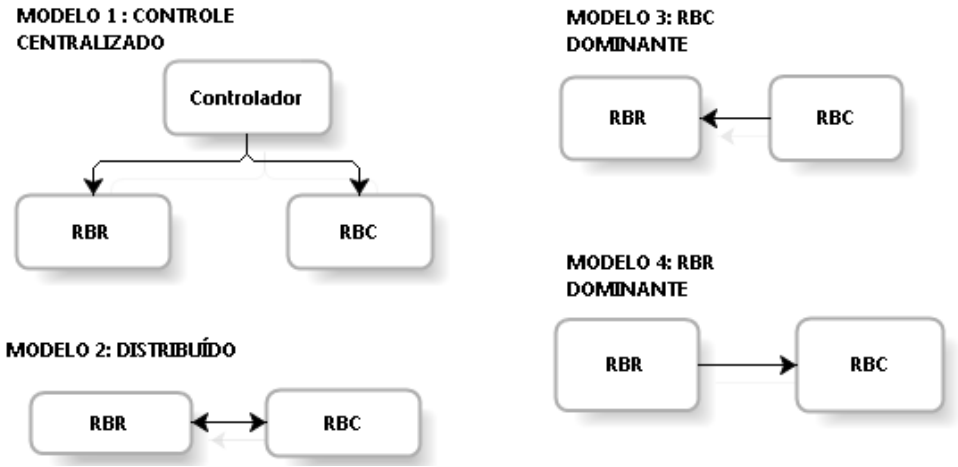
Comparando as técnicas de raciocínio, mostra que as duas podem se complementar. Sendo que a combinação de regras e casos podem trazer benefícios significativos, no momento em que, exploram-se as suas vantagens (PRENTZAS; HATZILYGEROUDIS, 2007).

Existem diferentes modos de realizar a combinação RBC com RBR para melhorar a eficiência na solução de problemas. Segundo Lee (2002), podemos classificar esta combinação em quatro modelos (Figura 12):

a) **Controle centralizado**: contém uma heurística para direcionar e intercalar os dois raciocínios, determinando e priorizando tarefas para cada raciocínio;
b) **Distribuído**: utilizam os dois raciocínios ao mesmo tempo para tentar solucionar um problema;
c) **RBC dominante**: a ideia central é de aplicar RBC para resolver problemas e de estabelecer regras para cobrir exceções dos casos;

d) **RBR dominante**: a ideia central é aplicar RBR para encontrar solução para o problema e determinar casos para cobrir exceções às regras.

Figura 12 – Classificação dos modelos de combinação RBC/RBR




Fonte: Adaptado de Lee (2002).

# 3 TRABALHOS CORRELATOS

Este capítulo destina-se à apresentação de trabalhos relacionados aos temas fundamentados no capítulo anterior. O objetivo é analisar criticamente estes trabalhos acerca das temáticas envolvidas na construção da abordagem metodológica proposta nesta dissertação, quais são: raciocínio baseado em casos, raciocínio baseado em regras, tomada de decisão, metodologia e ITIL. Além disso, é preciso se verificar a característica de originalidade deste trabalho.

Para se encontrar os trabalhos que serão apresentados a seguir, foram feitas pesquisas, nas quais se utilizou o engenho de busca Google Acadêmico, através do levantamento de produções científicas de 2008 até 2015 nas bases de dados: *IEEE Xplore; Citeseer; Springer* e *ACM*. Também se buscou referências dos principais trabalhos encontrados e que apresentavam maior relação com o tema desta dissertação.

Os trabalhos considerados mais relacionados com o tema foram os que utilizavam metodologia de apoio à decisão utilizando raciocínio baseado em casos e os que combinavam o raciocínio baseado em regras com o raciocínio baseado em casos para se apoiar a tomada de decisão.

É relevante se observar que outros trabalhos foram pesquisados, mas não foram aqui resumidos, pois o seu conteúdo estava representado em trabalhos selecionados.

Este capítulo está organizado da seguinte maneira: inicialmente, na Seção 3.1, apresenta-se uma metodologia para melhorar o ensino de programação para iniciantes, criado por Mendonça (2010). Na Seção 3.2, será analisada a automatização de geração de relatórios de segurança de redes usando-se RBC proposto por Erbacher e Hutchinson (2012). A Seção 3.3 analisa o trabalho desenvolvido por Carvalho (2008), que apresenta RBC como uma ferramenta capaz de auxiliar o processo decisório na resolução de problemas. A Seção 3.4 analisa o trabalho de Lopes e Schiel (2009), que trata de uma metodologia de apoio à decisão apoiada no ciclo do RBC para controle de penas alternativas. A Seção 3.5 traz combinação de RBR com RBC para apoio à decisão clínica de diagnóstico da Meningite Aguda, desenvolvido por Cabrera e Edye (2010). Na Seção 3.6, descreve-se uma abordagem baseada em RBC para gestão de conhecimento em sistemas de central de serviços baseados no ITIL, proposto por Tehrani e Mohamed (2011). Por fim, na Seção 3.7 apresentaremos uma análise comparativa destes trabalhos descritos.

## 3.1 PROGRAMAÇÃO ORIENTADA AO PROBLEMA: UMA METODOLOGIA PARA O ENTENDIMENTO DE PROBLEMAS E ESPECIFICAÇÃO NO CONTEXTO DE ENSINO DE PROGRAMAÇÃO PARA INICIANTES

O trabalho realizado por Mendonça (2010) descreve uma metodologia de ensino denominado Programação Orientada ao Problema (POP), voltada ao ensino de programação a iniciantes.

A motivação da autora para construção deste trabalho vem da necessidade de mudar a forma como é ensinada a disciplina introdutória de programação nas universidades. No sistema tradicional, o ensino é direcionado no desenvolvimento do raciocínio lógico e na construção de programas para solucionar problemas bem definidos, não sendo priorizadas outras habilidades, como o entendimento de problemas mal definidos e especificação de requisitos.

Para melhorar o ensino desta disciplina, a autora defende que se deve: (i) utilizar estratégias que integrem a programação como parte da engenharia de software; e (ii) desenvolver atividades para o aluno, que ao receber um problema mal definido pode ir da especificação dos requisitos à construção do programa. Portanto, é proposta a metodologia POP, com um conjunto de atividades a serem inseridas para apoiar o ensino da disciplina introdutória de programação.

A metodologia POP apresenta duas perspectivas: (i) do ensino: combinando atividades de engenharia de software e programação para os alunos resolverem problemas dos mais simples aos mais complexos; e (ii) da aprendizagem: proporcionando aos alunos uma visão mais realista do processo de construção de um programa, desenvolvendo habilidades para elicitação e especificação de requisitos, além de desenvolver outras habilidades, como o diálogo, a negociação e a escrita.

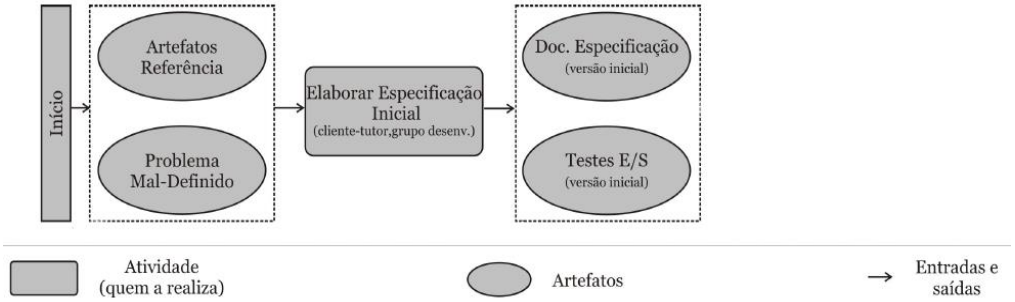
No trabalho proposto, os alunos, que serão os desenvolvedores de software, irão trabalhar em grupos, sendo acompanhados por um assistente, que fará o papel de cliente-tutor juntamente com o professor. Estes assistentes seguem uma especificação de requisitos documentada, preparada previamente pelo professor.

POP trabalha na forma de um ciclo de resolução de problemas que compreende quatro atividades: elaborar especificação inicial; iniciar implementação; concluir especificação; e concluir implementação. Estas atividades serão descritas a seguir:

- **Elaborar especificação inicial**: Uma primeira reunião entre os desenvolvedores e o cliente-tutor ocorre. Sendo apresentado textualmente um problema mal definido ao

grupo de desenvolvedores pelo cliente-tutor, estes devem gerar um documento com as especificações dos requisitos apresentados na reunião. Em seguida, o grupo deve conduzir um diálogo com o cliente-tutor para elicitar os requisitos. Ao final da reunião, uma documentação de especificação inicial e casos de testes de entrada/saída devem ter sido produzidos pelos desenvolvedores, conforme Figura 13.

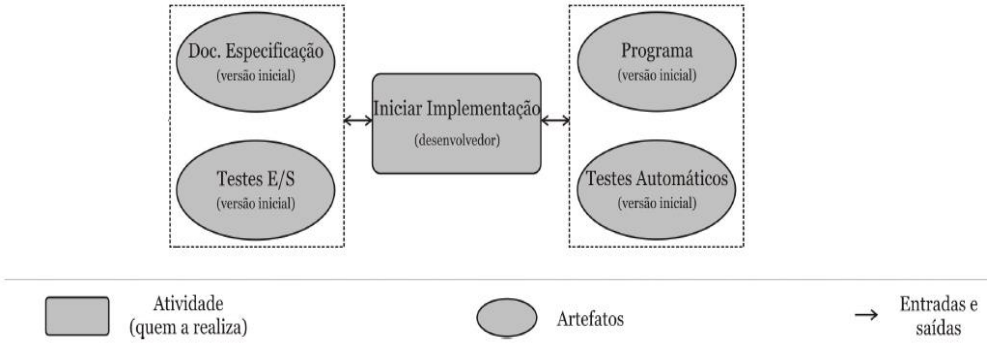
Figura 13 – Elaborar especificação inicial




Fonte: Mendonça (2010).

- **Iniciar a implementação**: Finalizada a atividade anterior, os desenvolvedores, de forma individual, irão iniciar a implementação; um protótipo do programa e de casos de testes automáticos será construído, atendendo aos requisitos especificados (Figura 14);

Figura 14 – Iniciar implementação

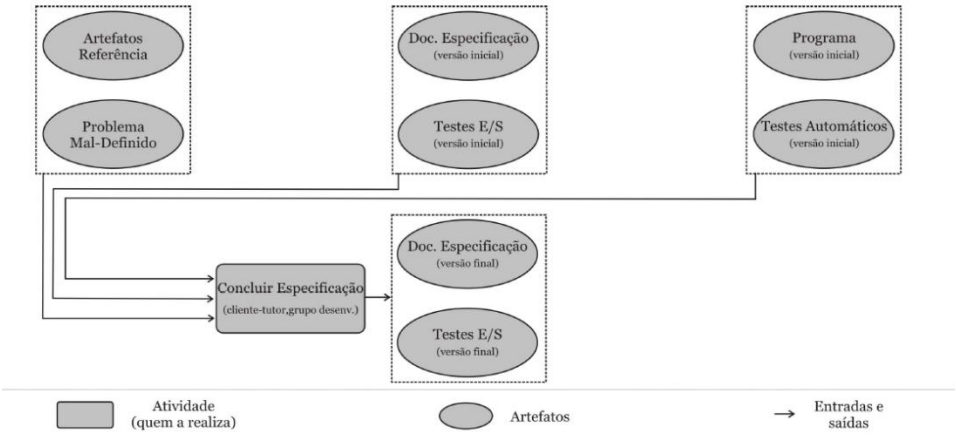



Fonte: Mendonça (2010).

- **Concluir especificação**: Uma segunda reunião entre cliente-tutor e desenvolvedores é estabelecida. Os desenvolvedores devem apresentar seu protótipo do programa e os testes automáticos ao cliente-tutor e retomar a interação sobre os requisitos e atualizá-

los juntamente com os casos de testes de entrada/saída (Figura 15). No final desta reunião, os grupos de desenvolvedores devem evoluir a especificação inicial para uma especificação final, contemplando os requisitos desejados pelo cliente-tutor;

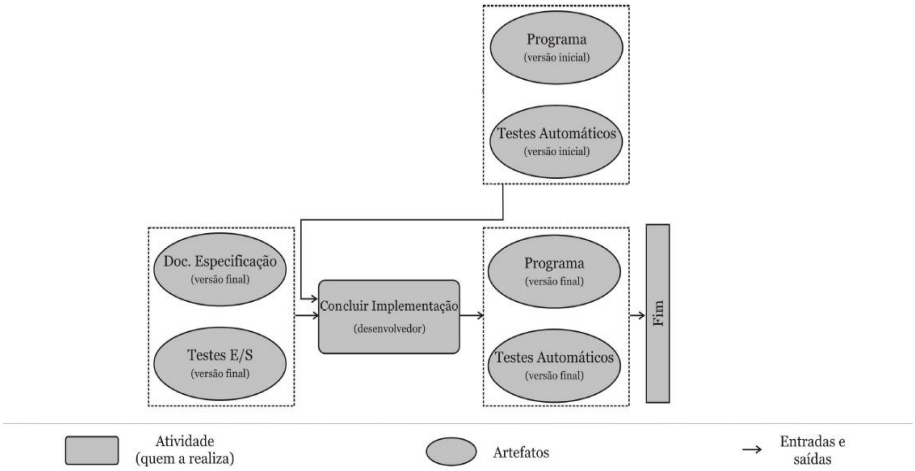
Figura 15 – Concluir especificação




Fonte: Mendonça (2010).

- **Concluir implementação**: Finalizado a atividade anterior, os desenvolvedores conhecem todos os requisitos solicitados pelo cliente-tutor. Os desenvolvedores, utilizando como referência o documento de especificação final e os casos de testes de entrada/saída, vão evoluir seus protótipos do programa e casos de testes automáticos, objetivando concluir a implementação (Figura 16). Ao finalizar esta atividade, o ciclo de resolução de problemas de POP é encerrado.

Figura 16 – Concluir implementação




Fonte: Mendonça (2010).

Com o intuito de validar a metodologia proposta, foram realizados estudos de caso e experimentos com 171 participantes, sendo 125 estudantes nos estudos de caso e 46 nos experimentos. No estudo de caso foi avaliada a aplicação de POP no ambiente de sala de aula. Nos experimentos, foram avaliados os efeitos de POP em alunos que adotaram versus alunos que não adotaram a metodologia. Como resultado destes experimentos foi demonstrado que os alunos POP conseguiram melhores resultados na: (i) documentação dos requisitos; (ii) implementação dos requisitos; e (iii) elaboração de questionamentos para a elicitação dos requisitos.

## 3.2 ESTENDENDO O RACIOCÍNIO BASEADO EM CASOS PARA GERAR RELATÓRIOS DE ALERTA DE REDES

O artigo de Erbacher e Hutchinson (2012) tem como proposta estender as abordagens existentes do raciocínio baseado em casos para apoiar as necessidades específicas de geração de relatórios de segurança cibernética, mais precisamente para o sistema de detecção de intrusão (IDS), que constantemente necessita gerar relatórios. No trabalho é explorada a habilidade de geração automática de relatórios de incidentes utilizando-se o RBC.

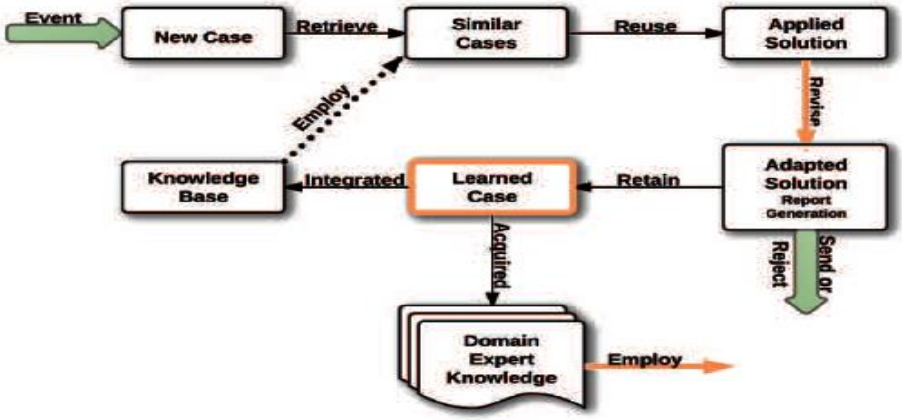
Os autores relatam que os analistas de segurança precisam de uma quantidade substancial de tempo para gerar relatórios de alerta de segurança de redes, e que muitas vezes estes alertas são bem conhecidos, por terem acontecido outras vezes.

Normalmente, a geração de relatórios requer que o analista introduza uma grande quantidade de informação, sendo essas: (i) computador afetado; (ii) revisão feita pelo analista; (iii) gravidade do ataque; (iv) evidências do ataque; e (v) identificação do usuário responsável pelo computador afetado.

O objetivo da pesquisa é reduzir os esforços necessários dos analistas de segurança para lidar com ataques considerados bem conhecidos. Para isso é aplicado um sistema RBC dentro de uma base de conhecimento de alertas, que tem a tarefa de correlacionar alertas conhecidos e novos e automatizar a geração de relatórios.

A aplicação para gerar relatórios de alertas segue as fases do Ciclo do RBC, da seguinte maneira (Figura 17):

Figura 17 - Modelo de processo RBC estendido para geração de relatórios de segurança de rede




Fonte: Erbacher e Hutchinson (2012).

Deve ser observado na Figura 17, que as ações do sistema estão em verde e as ações do analista estão em laranja.

Dado um evento (*event*), um alerta, tem-se um *new case* (novo caso) (ver Figura 17), é efetuada a recuperação (*retrieve*) de casos similares (*similar cases*) ao alerta na base de conhecimento (*knowledge base*) com o intuito de gerar o relatório deste novo caso de alerta. Esta recuperação correlaciona o novo alerta com vários alertas da base de conhecimento. Para melhorar a recuperação, a quantidade de detalhes fornecidos sobre o novo caso de alerta é fundamental. Dessa forma, ajuda a estabelecer uma melhor correlação entre os alertas e facilita a recuperação dos casos mais similares na base de conhecimento.

O reuso (*reuse*) fica por conta de reusar a melhor solução recuperada, sendo aplicada a solução (*applied solution*).

A revisão (*revise*) consiste em preencher um modelo de relatório com campos previamente definidos, informando detalhes deste novo caso de alerta. Se necessário, o analista de segurança inclui ou altera as informações preenchidas no modelo, com o intuito de garantir a exatidão do que está sendo relatado, adaptando a solução inicial (*adapted solution*). Em seguida, ele decide se distribui (*send*) o relatório gerado ou descarta (*reject*), conforme Figura 17.

O reter (*retain*) depende do caso, se ele deve ser aprendido (*learned case*) ou não. O analista de segurança vai analisar se o novo caso de alerta será um caso a ser aprendido pelo sistema (*learned case*). Se o novo caso de alerta tiver informações substanciais que o diferenciam dos outros casos existentes na base de conhecimento (*knowledge base*), este caso é aprovado pelo analista de segurança e integrado (*integrated*) na base de conhecimento. O

analista de segurança também verifica se este novo conhecimento dever ser adquirido (*acquired*) pelo domínio de conhecimento dos especialistas em segurança (*domain expert knowledge*).

O trabalho foi validado em estudos de caso. Para isso os autores construíram um modelo teórico de aplicação da solução. O cenário para exemplificar a utilização deste modelo teórico foi aplicado em três casos de alertas, que foram pesquisados em um site da internet que disponibilizam alertas de segurança.

Após os estudos de casos realizados com a aplicação do modelo teórico criado, ficou verificada a viabilidade da solução na geração de relatórios de segurança.

## 3.3 RACIOCÍNIO BASEADO EM CASOS APLICANDO NO PROCESSO DECISÓRIO

O trabalho de Carvalho (2008) apresenta o RBC como uma ferramenta capaz de auxiliar as tomadas de decisões dentro de uma organização, contribuindo para uma maior agilidade e velocidade destas decisões, além de maior grau de assertividade dos tomadores de decisão dentro de uma organização.

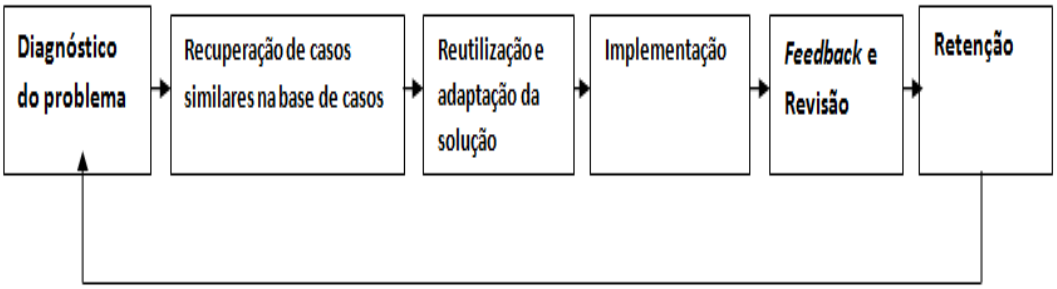
O artigo foca na decisão administrativa para se resolverem problemas e cita a ineficiência dos modelos tradicionais do processo decisório. Constata-se que o ambiente atual que envolve as organizações é constituído de inúmeras transformações que ocorrem de maneira extremamente acelerada e simultânea, exigindo tomadas de decisões em um menor tempo e de forma assertiva. Identifica-se também, que os modelos tradicionais de processo de tomada de decisão agem analítica e linearmente, reduzindo a eficácia na resolução do problema.

O autor propõe o uso do RBC para auxiliar na compreensão do ambiente e melhorar o processo decisório tradicional, tornando-o mais ágil e com um maior grau de assertividade.

Analisando-se as fases do processo decisório tradicional, o trabalho propõe a aplicação de um sistema RBC, que armazena experiências anteriores para solucionar problemas futuros, com o objetivo de realizar automaticamente, de maneira parcial ou total, as fases de levantamento de alternativas, análise de alternativas e escolha do plano de ação.

O processo decisório, proposto no trabalho com o sistema RBC, tem as seguintes fases: (i) diagnóstico do problema; (ii) recuperação de casos similares na base de casos; (iii) reutilização e adaptação da solução; (iv) implementação; (v) *feedback* e revisão; e (vi) retenção (Figura 18):

Figura 18 – Fases do processo decisório com RBC




Fonte: Autor deste trabalho (2015).

O diagnóstico e a contextualização do problema são realizados pelo tomador de decisão, fornecendo-se ao sistema RBC toda a descrição do problema.

As fases de recuperação e reutilização são efetuadas pelo sistema RBC automaticamente - através de busca na base de casos -, que retorna a solução mais similar encontrada e prontamente adaptada ao problema encontrado.

A implementação da solução cabe ao tomador de decisão, que pode recusar ou aceitar a solução proposta pelo sistema.

Um *feedback* dos resultados da solução é realizado e posteriormente decidido se o caso será retido ou não, a critério do tomador de decisão.

O trabalho propõe que a aplicação de um sistema RBC auxilia o processo decisório e reduz significativamente o tempo demandando para a tomada de decisão, visto que o levantamento de alternativas, a análise de alternativas e a escolha do plano de ação, que comumente consomem tempo em demasia, serão realizados de forma automática pelo sistema. Além disso, os autores afirmam que o mecanismo de aprendizagem do RBC faz com que o grau de assertividade das decisões dentro da organização aumente consideravelmente.

## 3.4 METODOLOGIA DE APOIO À DECISÃO PARA O CONTROLE DE PENAS ALTERNATIVAS: UMA ABORDAGEM UTILIZANDO RACIOCÍNIO BASEADO EM CASOS

O trabalho de Lopes e Schiel (2009) descreve sobre processos do subsistema de penas alternativas no Brasil e a complexidade da estrutura organizacional para julgamentos de processos jurídicos.

Os autores propõem uma metodologia de apoio à decisão para auxiliar os juízes com o objetivo de propiciar um controle e aplicar aos infratores penas alternativas das seguintes

unidades Judiciárias: (i) Juizado Criminal Especial; (ii) Departamento de execuções penais; e (iii) Departamento de Crimes Comum.

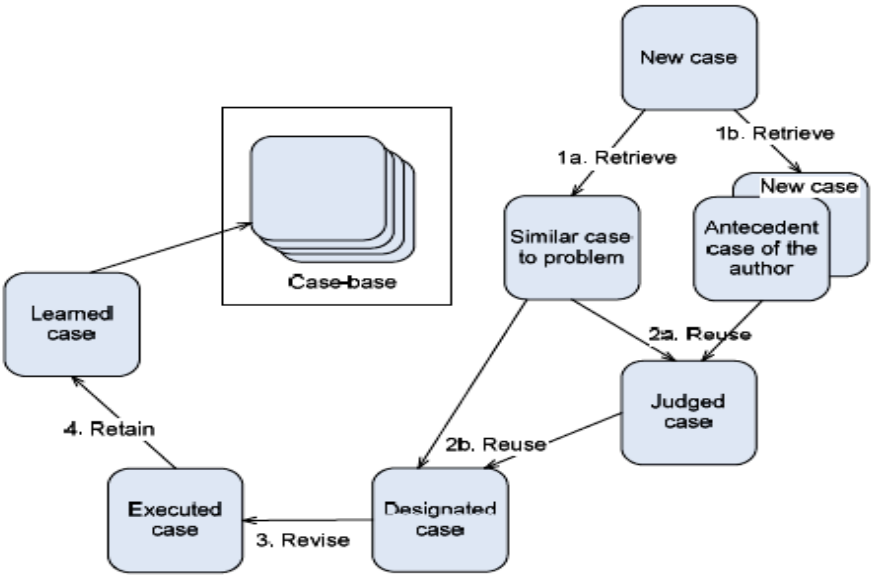
Esta metodologia é descrita através de um processo decisório geral e de tratamentos específicos de exceções, sendo desenvolvida sobre o modelo clássico de decisão de Simon e auxiliada pela técnica do Raciocínio Baseado em Casos. O objetivo é facilitar a tomada de decisão dos juízes, aumentando a produtividade e garantindo a interoperabilidade das penas alternativas aplicadas entres as diferentes unidades Judiciárias.

No sistema de penas alternativas, o processo de tomada de decisão geral envolve: (i) o julgamento; (ii) a designação do serviço de programas / instituições para atendimento; e (iii) execução penal das sentenças, sobre diferentes responsabilidades dos Juízes das unidades Judiciárias, exigindo sincronismo entre eles. A metodologia está apoiada no Ciclo adaptado do RBC para auxiliar este sincronismo.

Para o domínio especificado, o ciclo básico do RBC não satisfaz as atividades de recuperar e reutilizar. Estas atividades são adaptadas para acompanharem o sincronismo da tomada de decisão dos juízes.

Os juízes, ao se depararem com um novo julgamento de penas alternativas, devem seguir os seguintes passos da metodologia proposta para a tomada de decisão (Figura 19):

Figura 19 – Ciclo RBC adaptado para o sistema de penas alternativas




Fonte: Lopes e Schiel (2009).

a) **Recuperar (*retrieve*):** efetua duas recuperações: a primeira para casos similares ao problema (*similar case to problem*) que está sendo julgado (*1a*

*retrieve*), e a segunda para os casos que indiquem os antecedentes criminais do autor (*antecedent case of the author*) que está sendo julgado (*1b retrieve*);

b) **Reusar (*reuse*):** executa a fase de reuso, que também é composta de dois passos: (i) o primeiro para o julgamento (*judged case*), que é baseado na reutilização de sentença condenatória após se analisarem as informações das duas recuperações realizadas (*2a reuse*); (ii) o segundo é para a designação do serviço (*designated case*) a ser prestado com base em uma designação anterior bem sucedida (*2b reuse)*;

c) **Revisão *(revise)*:** efetua a revisão, que consiste no acompanhamento da execução penal da sentença do novo caso (*executed case*);

d) **Reter *(retain)*:** corresponde ao armazenamento do caso aprendido.

No trabalho foi apresentado um protótipo, construído com o *framework* Jcolibri, implementado na linguagem de programação Java, construído com o objetivo de demonstrar a viabilidade técnica do trabalho.

Os autores enfatizam que a aplicação da metodologia proposta para o sistema Brasileiro de penas alternativas auxilia a diversidade das tomadas de decisão com base na recuperação de experiências passadas, além de facilitar a operação e sincronismo entre os juízes.

## 3.5 INTEGRAÇÃO DE SISTEMA ESPECIALISTA BASEADO EM REGRAS E RACIOCÍNIO BASEADO EM CASOS PARA UM SISTEMA DE APOIO À DECISÃO CLÍNICA DA MENINGITE BACTERIANA AGUDA.

Este artigo, dos autores Cabrera e Edye (2010), apresenta um sistema de apoio à decisão de diagnostico médico aplicado à meningite bacteriana aguda usando-se a metodologia RBC. A pesquisa foi focada na execução da fase de adaptação do ciclo RBC. A proposta é integrar Raciocínio Baseado em Casos e Raciocínio Baseado em Regras, tendo-se como objetivo oferecer um diagnóstico mais preciso da meningite bacteriana aguda.

O diagnóstico médico consiste em traduzir um conjunto de sintomas e sinais encontrados em um determinado paciente, e as patologias que este paciente pode sofrer. Na pesquisa realizada para diagnosticar a Meningite Bacteriana Aguda (MBA), foram considerados como referência um total de 81 sintomas, tais como: convulsões, depressão, febre, hipertensos fontanela, rigidez de nuca, rigidez no tronco, febre, vómitos, sonolência.

A combinação da presença de alguns destes sintomas permite identificar a doença em estudo, mas a depender da combinação destes sintomas pode caracterizar diferentes tipos de MBA, quais são: meningite viral aguda, meningite tuberculosa, encefalite, abscesso cerebral,

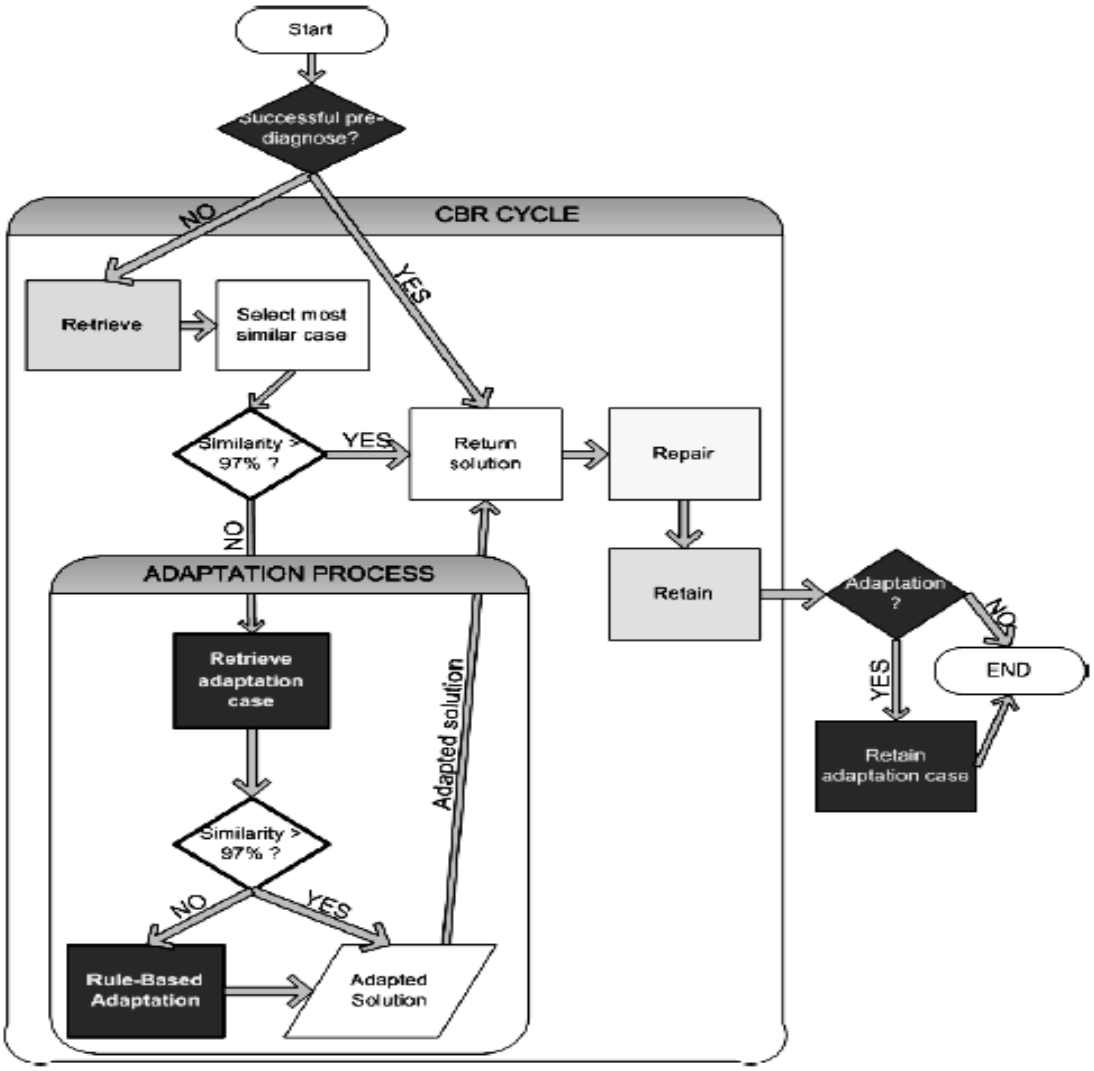
meningismo, reação à inflamação meníngea nas proximidades, meningeal hemorrágica e tumor cerebral. O grande desafio do médico é diagnosticar com precisão qual o tipo de MBA afetou o paciente.

Para auxiliar o médico no diagnóstico de MBA, o trabalho propõe um sistema híbrido desenvolvido para apoiar esta tomada de decisão do médico, combinando-se RBC e RBR.

O sistema proposto recebe como entrada informações dos sintomas do paciente, um novo caso. O sistema é inicializado (*start*) aplicando-se a etapa de pré-diagnóstico (ver Figura 20). Nesta etapa, um conjunto de regras básicas de diagnóstico contendo informações dos sintomas da doença pesquisada possibilita identificar a presença da doença. Estas informações são fornecidas por médicos especialistas da área de meningite bacteriana aguda.

As regras do pré-diagnóstico seguem a seguinte estrutura: *SE* sistema nervoso central apresenta aspecto nublado *ENTÃO* MBA.

Figura 20 - Imagem explicativa do sistema desenvolvido




Fonte: Cabrera e Edye (2010).

O objetivo desta etapa é aumentar o desempenho do sistema, pois passa pelo processo de inferência completa fornecida pelo RBR. Além disso, os autores definem esta etapa como fundamental, principalmente quando a base de conhecimento que contém os casos é pequena ou não contém caso algum.

Se a fase de pré-diagnóstico é bem sucedida, por exemplo, se o paciente apresenta "aspecto nebuloso no líquido cefalorraquidiano" então a conclusão da regra é acionada, indicando MBA. Dessa forma, o pré-diagnóstico retorna uma solução (*return solution*) para o sintoma apresentado. A reparação, bem sucedida ou não, pode ser retida (*retain*) na base de conhecimento para auxiliar em futuros diagnósticos.

Se o caso não é tão óbvio ou simples, o pré-diagnostico falha, e o sistema segue para processar o novo caso, utilizando-se o método RBC. Uma nova consulta é construída a partir dos sintomas do novo caso. O sistema recupera (*retrieve*), utilizando o método do vizinho mais próximo, os três casos mais similares. Em seguida, o médico seleciona o caso que ele acredita ser o mais similar (*select most similar case*) para ser reutilizado. Os casos apresentam três informações: (i) a descrição da situação problemática; (ii) a solução para tal situação; e (iii) e um indicador de sucesso para solução.

Após a recuperação, o sistema avalia o grau de similaridade do caso selecionado com a relação ao novo caso. Caso esteja acima do limiar determinado de 97% (*similanty >97?*) a solução é reutilizada diretamente sem adaptação (Figura 20).

Caso não ocorra o limiar determinado para similaridade, o processo de adaptação que integra RBC e RBR é aplicado. Funciona da seguinte forma: o sistema recebe como entrada o que foi diferente entre os sintomas do caso recuperado e o novo caso, bem como a solução oferecida pelo caso recuperado. A partir dessa entrada, o sistema RBC recupera uma nova solução adaptada com relação à solução inicial (*retrieve adaption case*), que, se satisfizer o limiar de similaridade de 97% (*similanty >97?*), será imediatamente reutilizada.

Caso a adaptação utilizando o RBC não satisfaça o limiar de similaridade de 97%, uma adaptação combinando RBC e RBR é realizada. Este processo consiste de um conjunto de regras que permite adaptar as soluções (*rule-based adaptation*). Além disso, utilizando RBC é possível reutilizar (*return solution*) e armazenar a nova experiência (*retain*). O processo recebe como entrada a diferença entre os sintomas do último caso recuperado para o novo caso, assim como a solução oferecida pelo último caso recuperado. A partir destas entradas, este processo, através das regras, oferece uma solução adaptada ao novo problema.

Para a fase de retenção (*retain*), o médico é questionado se pretende reter o novo caso ou não. Além disso, se um processo de adaptação foi realizado durante a resolução (*adaption?*), o usuário deve reter este caso adaptado (*retain adaptation case*), tornando-o disponível para uma nova consulta.

O desempenho do sistema desenvolvido foi avaliado considerando-se trinta casos diferentes. A sua utilização obedecendo à abordagem desenvolvida apresentou um resultado de precisão de diagnóstico em 80% dos casos.

Os autores consideraram os resultados do trabalho desenvolvido como excelente, por apresentar diagnósticos mais precisos e robustos para o diagnóstico da meningite bacteriana aguda.

## 3.6 UMA ABORDAGEM BASEADA EM RBC PARA CENTRAL DE SERVIÇOS BASEADAS NO ITIL

No trabalho de Tehrani e Mohamed (2011), os autores relatam que o ITIL vem evoluindo e em sua versão mais recente, é proposta a Gestão de conhecimento, que se trata de uma tecnologia que aumenta nossa compreensão e ajuda as organizações a tomarem decisões e resolverem problemas de forma mais eficaz, possibilitando compartilhar informações e experiências.

Por outro lado, quanto aos sistemas de Central de Serviços, detentores de uma grande quantidade de dados e informações, principalmente proeminentes de incidentes e problemas de TI, poucos adotam tecnologia da informação para melhorar a aquisição de conhecimento, ou essa adoção em muitos casos, não acontece de forma prática.

Os autores afirmam ainda que profissionais que resolvem incidentes de TI e que não possuem o conhecimento sobre um determinado domínio, apresentam dificuldades para encontrar soluções e resolver os incidentes. E que isso poderia ser resolvido com um método eficaz para procurar soluções em base de dados.

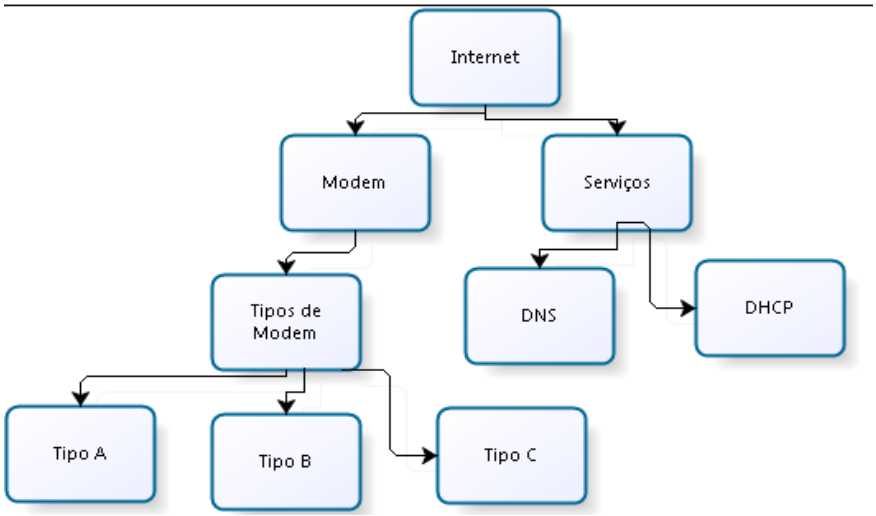
O objetivo do trabalho é melhorar o funcionamento dos sistemas de Central de serviços, utilizando tecnologias para melhorar a aquisição de conhecimento e, consequentemente, oferecer melhores soluções para incidentes de TI. Para isso, a estrutura do sistema de Central de Serviços proposto no ITIL, será integrando com uma biblioteca de regras e o Raciocínio Baseado em Casos (RBC).

O funcionamento desta nova abordagem de sistemas de Central de serviços, proposta pelos autores, é descrita em partes, sendo estas:

a) **Categorização:** A proposta descreve que inicialmente todos os componentes e serviços de TI da entidade devem ser categorizados de forma hierárquica, em categorias e subcategorias, com o intuito é melhorar o desempenho do sistema nas buscas e recuperações futuras. Para esclarecer, como funciona a hierarquia de categorias, foi ilustrada a árvore de categorização relacionada à internet (ver Figura 21);

b) **Caminho compartilhado**: esta segunda parte tem o objetivo de tornar a recuperação de casos mais rápida. Considerando a Figura 21, o funcionamento ocorre da seguinte forma: se um determinado incidente foi relacionado na categoria (internet*)*, mas, na realidade ele está relacionado com a subcategoria (Modem), uma investigação preliminar pode encaminhar a pesquisa diretamente para subcategoria (Modem);

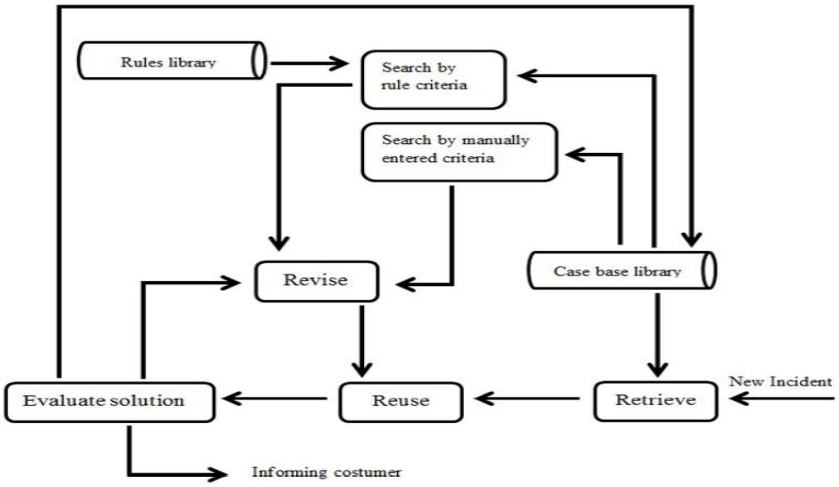
Figura 21 – Categorização dos componentes relacionados ao modem




Fonte: Adaptado de Tehrani e Mohamed (2011).

c) **Fluxo RBC**: após o projeto e a categorização dos serviços é aplicado o seguinte fluxo de trabalho, (ver Figura 22).

Figura 22 – Novo fluxo de trabalho RBC




Fonte: Tehrani e Mohamed (2011).

Ao deparar-se com um novo incidente (*New incident*), é realizada uma pesquisa utilizando o sistema RBC para tentar solucioná-lo, realizando o seguinte: recupera (*Retrieve*) casos anteriores similares, reusa (*Reuse*) o melhor caso similar recuperado, revisa (Revise) a solução, validando-a através de testes e, por fim, avalia a solução (*Evalute solution*) para determinar se deve ser aplicada ao incidente sem necessidade de adaptação.

Caso não seja encontrada uma solução direta para o problema, é necessário um método de adaptação. Para cumprir essa etapa, a biblioteca de casos (*Case base library*) é pesquisada de acordo com regras de critérios estabelecidos (*Search by rule criteria*) (*Search by manually entered criteria*) a serem pesquisados na biblioteca de regras *(Rules library)*, utilizando RBR. Biblioteca esta que contém regras de como podem ser substituídos os valores dos atributos de uma categoria ou subcategoria para se obter uma nova solução com a probabilidade maior de se resolver o novo incidente.

Por exemplo: considerando-se a Figura 21, se o sistema estiver pesquisando na subcategoria (Tipos de Modem) da subcategoria (Modem), uma regra definida na *(Rules library)* para pesquisar no vizinho mais próximo pode ser aplicada para procurar melhores soluções em outro tipo de modem. Dessa forma, adaptando a solução inicial.

d) **Aprendizado:** Os autores constatam que, à medida que novos casos resolvidos são adicionados, na *case base library*, é menor o tempo necessário para se resolverem os problemas e os incidentes de entrada, o que torna o sistema mais eficaz.

Um sistema de Central de serviços foi construído com o intuito de validar a nova abordagem proposta. Os testes com o sistema foram realizados com mais de 850 registros, sobre problemas de hardware e software.

Foram obtidos os seguintes resultados com os testes realizados: (i) utilizando o método de categorização, a busca e a apresentação dos resultados aconteceram em um tempo menor; e (ii) a biblioteca de regras (*rules library*), criada para adaptação automaticamente da solução inicial, foi construída cuidadosamente com o conhecimento do especialista do domínio, produzindo resultados eficazes para encontrar o vizinho mais próximo ao incidente atual.

Os autores observaram algumas limitações com o novo sistema de Central de serviços. Por exemplo, encontrar soluções depende de casos resolvidos e, em casos de incidentes inéditos, essa nova abordagem pode falhar.

O trabalho conclui que pesquisas futuras na aplicação de técnicas de gestão do conhecimento em sistemas de Central de serviços são recomendadas para que se possa obter cada vez mais, melhores resultados.

## 3.7 ANÁLISE COMPARATIVA DOS TRABALHOS RELACIONADOS

Ao analisarmos as obras relatadas aqui nesta seção, apresentamos o Quadro 2 com os critérios que servem de comparação com a proposta desta dissertação. Os artigos selecionados, que foram descritos neste capítulo, foram pesquisados satisfazendo pelo menos um dos cincos critérios adotados, considerados mais significantes para este trabalho. Os critérios usados foram: (i) utiliza o raciocínio baseado em casos, (ii) utiliza o raciocínio baseado em regras, (iii) detém uma *metodologia de apoio à decisão*, (iv) *baseia-se nos conceitos da ITIL*, (v) relata incidentes de TI.

Quadro 2 - Análise comparativa dos trabalhos correlatos

| Trabalho/ Critério | RBC | RBR | Metodologia de apoio à decisão | ITIL | Incidentes de TI |
|---|---|---|---|---|---|
| (1) Mendonça (2010) | | | x | | |
| (2) Erbacher e Hutchinson (2012) | x | | | | x |
| (3) Carvalho (2008) | x | | x | | |
| (4) Lopes e Schiel (2009) | x | | x | | |
| (5) Cabrera e Edye (2010) | x | x | | | |
| (6) Tehrani e Mohamed (2011) | x | x | | x | x |

Fonte: Autor deste trabalho.

A análise comparativa dos seis trabalhos apresentados no Quadro 2 mostra que o trabalho (6) aborda quatro dos cinco critérios adotados para o estudo da dissertação, demonstrando ser o artigo com maior completude para este trabalho. Porém não apresenta uma abordagem metodológica de apoio à tomada de decisão.

Somente o trabalho (1) não abordou RBC. As abordagens (1), (3) e (4) apresentam uma metodologia com o objetivo de auxiliar usuários tomadores de decisão. Destaque para o trabalho (4) que descreveu uma metodologia de apoio à decisão utilizando o RBC para o domínio judiciário, que tem a proposta de guiar os Juízes através de passos a serem seguidos para tomada de decisão. Esta proposta aproxima-se do que desejamos desenvolver neste trabalho, pois envolve um guia com passos apoiado por RBC para tomada de decisão.

Por sua vez, o trabalho (3), aborda a composição do processo de tomada de decisão com RBC. Esta composição é de grande relevância, pois possui uma similaridade direta com a metodologia proposta neste trabalho.

Os trabalhos (5) e (6) combinam o uso do RBR com RBC para apoiar a tomada de decisão, o (5) no domínio médico, e o (6) na área de incidentes de TI. Ambos, com a combinação dos raciocínios, oferecem diagnósticos mais precisos e melhores resultados.

O trabalho (2) apresentou o desenvolvimento de sistema, baseado em RBC, para geração de relatórios de incidentes relacionados à segurança da rede. Estes sistemas RBC facilitam as tomadas de decisões com o reuso de experiências vivenciadas.

Os trabalhos descritos neste capítulo auxiliam os objetivos desta dissertação, porém de forma isolada. Faz-se necessário uma abordagem metodológica de apoio à decisão para o domínio de resolução de incidente de TI utilizando RBC e RBR, com o objetivo de guiar e apoiar a tomada de decisão dos integrantes da equipe de suporte.

# 4 ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PROPOSTA

Neste capítulo é apresentada a Abordagem Metodológica de apoio à decisão proposta, que foi construída respeitando o processo de tomada de decisão de Simon, seguindo as atividades de resolução de incidentes de TI descrita no ITIL, e utilizando raciocínio por regras, raciocínio por analogia, e a combinação destes raciocínios para auxiliarem os integrantes das equipes de suporte nas tomadas de decisões.

Para a construção desta abordagem, a aquisição de conhecimento efetivou-se através de:

a) pesquisa bibliográfica sobre a resolução de incidentes de TI, em especial a biblioteca ITIL (OGC, 2011a) e (OGC, 2011b);
b) análise do processo de tomada de decisão geral das equipes de suporte de duas organizações; uma do setor privado que presta serviço social aos comerciários no Estado da Bahia e outra do setor público do Estado da Bahia;
c) análise de casos de incidentes coletados nestas organizações;
d) pesquisa bibliográfica sobre o processo de tomada de decisão de Simon;
e) pesquisa bibliográfica sobre técnicas de apoio a decisão, principalmente, RBC e RBR;
f) entrevistas com especialistas da área de suporte de TI.

Dessa forma, foi possível:

a) conhecer o domínio de suporte para resolução de incidentes de TI;
b) conhecer as principais atividades dos seus atores (técnicos da central de serviços, gerentes, supervisores, analistas, etc.);
c) organizar as atividades e passos em um processo decisório.

Inicialmente, na Seção 4.1, apresentaremos algumas considerações iniciais para entendermos a representação dos casos de incidentes e das regras, possibilitando assim a utilização do RBC e RBR. Na Seção 4.2 é exibida uma visão geral da abordagem metodológica. Na Seção 4.3 são apresentadas de forma específica e detalhada as atividades auxiliadas por RBR e/ou RBC, com os passos necessários para a resolução de incidentes de TI. Por fim, uma visão integrada das atividades e passos que foram especificados nas Seções 4.2 e 4.3 é apresentada na Seção 4.4.

## 4.1 CONSIDERAÇÕES INICIAIS

Antes de apresentarmos a abordagem metodológica proposta, é interessante fazermos algumas considerações sobre como serão representados o caso e as regras para apoiarem a resolução de incidentes de TI. Dessa forma, possibilita-se a utilização de RBC e RBR.

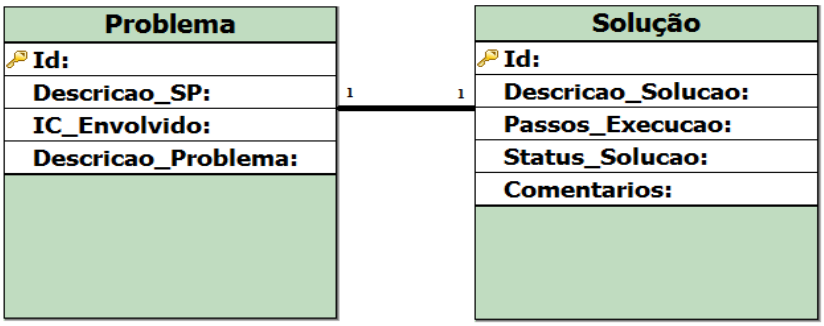
### 4.1.1 Representação de caso para utilização do RBC

Para utilizarmos RBC no apoio à decisão deste trabalho é necessário representar o caso de incidentes de TI para que ele seja armazenado e recuperado em uma base de casos.

No contexto de resolução de incidentes de TI, o caso pode ser representado com informações da situação-problema e do problema do incidente, incluindo a solução técnica empregada pelo integrante da equipe de suporte para solucioná-lo.

Para este trabalho, a representação do caso para controle de resoluções de incidentes foi criado de acordo com a literatura sobre RBC, a análise do processo de tomada de decisão no ambiente de duas organizações citadas anteriormente e a avaliação de casos coletados em tais organizações. O modelo lógico apresentado na Figura 23 exibe esta representação do caso.

Figura 23 - Representação do caso




Fonte: Autor deste trabalho (2015).

O elemento problema tem os seguintes atributos: (i) descrição da situação-problema (Descricao_SP); (ii) item de configuração envolvido no incidente (IC_Envolvido); (iii) descrição do problema do incidente (Descricao_Problema).

O elemento solução é descrito com os seguintes atributos: (i) descrição da solução para o incidente (Descricao_Solucao); (ii) passos necessários para execução da solução (Passos_Execucao); (iii) status da solução (Status_Solucao), indicando sucesso ou falha para aplicação da solução proposta; e (iv) comentários (Comentários).

Para implementação das funções de similaridade, foi utilizado o algoritmo de recuperação do vizinho mais próximo (*Nearest Neighbour)*. Este algoritmo realiza a comparação entre o caso com problema atual e os casos da base de casos, efetuando cálculos de similaridade. Inicialmente deve-se calcular a similaridade local, partindo de uma função que estabelece o quão semelhantes são os valores para um atributo específico. Sendo que, para cada atributo é determinado um peso, que representa seu grau de relevância. Por conseguinte, é calculada a similaridade global determinada pela média ponderada das similaridades locais.

Para o cálculo da similaridade local dos atributos, foi utilizada a função *Max_string*, uma adaptação do coeficiente de *Jaccard*, que retorna o valor da maior *substring* pertencentes as duas *strings* e divide esse valor pelo tamanho da maior *string* dentre elas.

No Quadro 3 são apresentadas a função de similaridade local e o peso dado a cada atributo representado no caso. Os atributos que serão comparados são: (i) Descrição_SP; (ii) Descrição_Problema. Os pesos atribuídos para os atributos foi 1.0.

Quadro 3 - Similaridade local para os atributos do caso

| **Atributo** | **Similaridade Local** | **Peso** |
|---|---|---|
| Descrição_SP | *MaxString()* | 1.0 |
| Descrição_Problema | *MaxString()* | 1.0 |

Fonte: Autor deste trabalho (2015).

Cada atributo será comparado de forma separada, em duas pesquisas distintas. Assim, para este caso específico, a definição do peso dos atributos não é relevante. Foi determinado o peso 1.0, exclusivamente, para realização do cálculo de similaridade.

### 4.1.2 Representação das regras para utilização do RBR

A construção das bases de regras aqui apresentadas seguirão os procedimentos mencionados no item 2.3.2.3.

Foram modeladas duas bases de regras de situações-problema e uma base de regras de problema do incidente, com o intuito de exemplificar a utilização do RBR e estão descritas da seguinte forma:

a) Situação-problema 1 : sem acesso aos sistemas da organização;
b) Situação-problema 2 : não ingressa as pastas de rede;
c) Problema de incidente 1: desbloqueio de porta de rede web

Para a **situação-problema 1 "**sem acesso aos sistemas da organização". Os seguintes parâmetros foram definidos:

a) **Variáveis:** ***univaloradas*** (cabo, ip_manual, parâmetro_configuraçao, link_dados); ***multivaloradas*** (endereço_ip);

b) **Variável objetivo:** diagnóstico_inicial;

c) **Regras criadas:** de acordo com o Quadro 4;

d) **Questionamento de problema**: (i) "o cabo de rede está conectado ao computador?"; (ii) "como aparece o endereço IP?"; (iii) "as configurações TCP/IP do computador cliente encontram-se configuradas de forma manual?"; (iv) "ao executar um *ping* para o Gateway da unidade a resposta é positiva?"; (v) "Ao clicar para abrir o sistema aparece a MSG: Parâmetro de Configuração Inválido".

Quadro 4 – Regras implementadas para situação-problema "sem acesso aos sistemas da organização"

| | | |
|---|---|---|
| **REGRA 1** | SE cabo = não | ENTÃO "conecte o cabo de rede" |
| **REGRA 2** | SE end_ip = 169.254.x.x ou 127.0.0.1 | ENTÃO "Verificar o serviço DHCP do servidor." |
| **REGRA 3** | SE ip_manual = sim | ENTÃO "Configurar IP automático nas propriedades TCP/IP" |
| **REGRA 4** | SE link_dados = não | ENTÃO "Reiniciar gateway e entrar em contato com a operadora do link." |
| **REGRA 5** | SE parametro_configuraçao = sim | ENTÃO "Configurar o DB2 cliente para o servidor: srvdb2 e porta de rede: 50000" |

Fonte: Autor deste trabalho (2015).

Para a **situação-problema 2 "**não ingressa as pastas de rede". Os seguintes parâmetros foram definidos:

a) **Variáveis:** ***univaloradas***: (cabo, ip_manual, mapeamento, permissão); ***multivaloradas***: (endereço_ip);

b) **Variável objetivo:** diagnóstico_inicial;

c) **Vegras criadas:** de acordo com o Quadro 5;

d) **Questionamento de problema**: As três primeiras perguntas foram iguais às da situação-problema 1, sendo acrescentado: (i) "acessando Meu Computador. Aparecem as pastas mapeadas?"; (ii) "ao acessar as pastas. É apresentada a MSG: "Acesso negado"".

Quadro 5 - Regras implementadas para situação-problema “não ingressa as pastas de rede”

| | | |
|---|---|---|
| **REGRA 1** | SE cabo = não | ENTÃO “conecte o cabo de rede” |
| **REGRA 2** | SE end_ip = 169.254.x.x ou 127.0.0.1 | ENTÃO “Verificar o serviço DHCP do servidor.” |
| **REGRA 3** | SE ip_manual = sim | ENTÃO “Configurar IP automático nas propriedades TCP/IP” |
| **REGRA 4** | SE mapeamento = não | ENTÃO “Executar o script de mapeamento de pasta de rede. \\190.100.1.1\script\script.bat.” |
| **REGRA 5** | SE permissão = não | ENTÃO “Conceder permissão se autorizado.” |

Fonte: Autor deste trabalho (2015).

Para o **problema de incidente 1 “**desbloqueio de porta de rede web”. Os seguintes parâmetros foram definidos:

a) **Variáveis: *multivaloradas***: sistema_operacional e gerenciador_web;

b) **Variável objetivo:** problema;

c) **Regras criadas:** de acordo com o Quadro 6;

d) **Questionamento de problema**: (i) “o gerenciador de conteúdo web encontra-se instalado em um Sistema operacional Windows ou Linux?”; (ii) “qual o gerenciador de conteúdo web utilizado?”.

Quadro 6 – Regras para desbloqueio de porta web

| | | |
|---|---|---|
| **REGRA 1** | **SE** Sistema_Operacional = Linux **E** Gerenciador_Web = Apache | **ENTÃO** “Desbloqueio no Firewall IPTABLES da porta 8080” |
| **REGRA 2** | **SE** Sistema_Operacional = Windows **E** Gerenciador_Web = IIS | **ENTÃO** “Desbloqueio no Firewall do Windows da porta 80” |
| **REGRA 3** | **SE** Sistema_Operacional = Windows **E** Gerenciador_Web = Apache | **ENTÃO** “Desbloqueio no Firewall do Windows da porta 8088” |
| **REGRA 4** | **SE** Sistema_Operacional = Linux **E** Gerenciador_Web = JBOSS | **ENTÃO** “Desbloqueio no Firewall IPTABLES da porta 8192” |

Fonte: Autor deste trabalho (2015).

## 4.2 VISÃO GERAL DA ABORDAGEM METODOLÓCIA

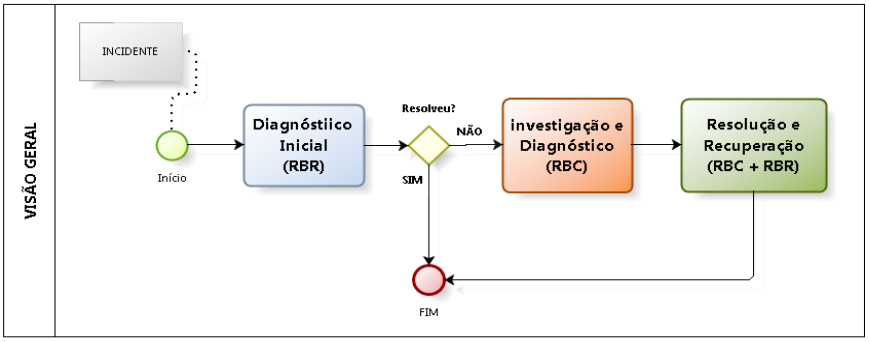
A Abordagem Metodológica de apoio à decisão voltada para a resolução de incidentes de TI segue as principais atividades determinadas no ITIL para resolver incidentes, quais são: (i) diagnóstico inicial; (ii) investigação e diagnóstico; e (iii) resolução e recuperação. Cada uma destas atividades utilizará RBC e/ou RBR para apoiar as tomadas de decisões.

Na Figura 24 é apresentada a visão geral do processo decisório a ser detalhado em passos ordenados. Foi utilizado BPMN[1] (*Business Process Model and Notation*) para sua representação.

A resolução de incidentes é inicializada com a identificação de uma ocorrência de incidente, normalmente fornecida pelo usuário dos serviços de TI.

A partir deste incidente relatado, a primeira atividade a ser efetuada para tentar estabelecer imediatamente o serviço de TI é o diagnóstico inicial, sendo efetuada, exclusivamente, pelos integrantes do suporte de 1º nível. O raciocínio por regra (RBR) será utilizado para apoiar as tomadas de decisões relacionadas a esta atividade, visto que RBR tende a produzir conclusões imediatas contendo informações para resolver o problema de incidente, sendo indicado para a tomada de decisões repetitivas e estabelecimento de diagnósticos.

Figura 24 – Visão geral da abordagem metodológica proposta




Fonte: Autor deste trabalho (2015).

Caso não se resolva o incidente no diagnóstico inicial, deve-se seguir para as atividades: investigação e diagnóstico, e resolução e recuperação. Estas atividades podem ser realizadas tanto por integrantes de suporte de 1º nível quanto de 2º nível.

Para o trabalho proposto, utilizaremos o raciocínio por analogia (RBC) afim de auxiliar as tomadas de decisões da atividade investigação e diagnóstico.

Por sua vez, para auxiliar as tomadas de decisões da atividade resolução e recuperação, usaremos a combinação de RBC com RBR, mais precisamente, o modelo de combinação RBC dominante definido por Lee (2002).

[1] BPMN auxilia na criação de modelos de processos. É uma notação gráfica para desenho de processos. BPMN permite a utilização de subprocesso, que consiste em permitir que o processo atual inicie outra instância de outro processo. A representação gráfica de subprocessos são quadrados com as bordas arredondadas com um sinal de + em seu interior. Ver em http://www.bpmn.org/.

## 4.3 DETALHAMENTO DA ABORDAGEM METODOLÓGICA DE APOIO À DECISÃO PARA RESOLUÇÃO DE INCIDENTES DE TI

Os passos para resolução de incidentes de TI desta abordagem metodológica estão descritos nas atividades: (i) diagnóstico inicial; (ii) investigação e diagnóstico e (iii) resolução e recuperação. Sendo auxiliadas por RBC e/ou RBR. Estas atividades são detalhadas a seguir.

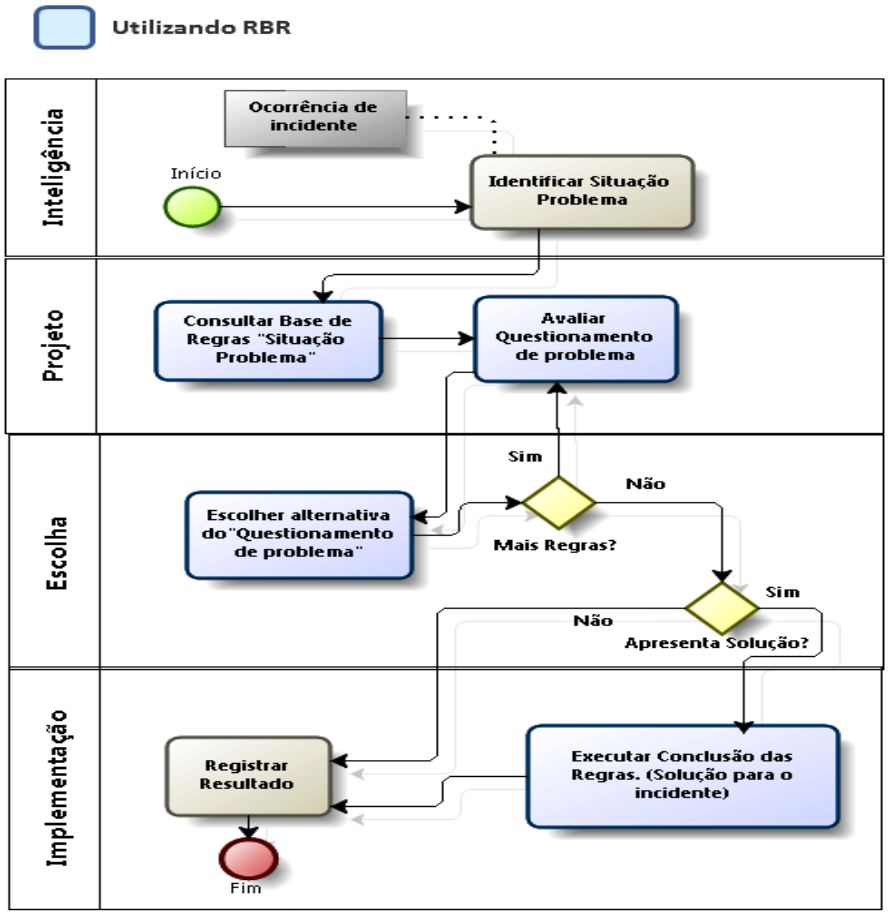
### 4.3.1 Diagnóstico Inicial Auxiliado por RBR

Aqui, serão descritos os passos que têm o objetivo de guiar o integrante da equipe de suporte na atividade diagnóstico inicial, para a resolução de incidentes de TI utilizando o raciocínio por regras com RBR.

A Figura 25, modelada em BPMN, apresenta o processo de tomada de decisão na resolução de incidentes em TI que foi construído após a análise da tomada de decisão no domínio das empresas aqui mencionadas. Ela é baseada no que é descrito no ITIL, em especial a atividade diagnóstico inicial para a resolução de incidentes, de acordo com as fases da tomada de decisão de Simon, quais são: (i) inteligência; (ii) projeto; (iii) escolha; e (iv) implementação.

A Central de Serviços, 1º nível do suporte, recebe uma ocorrência de incidente que, em geral, é uma descrição textual feita pelo usuário com as próprias percepções vivenciadas no ambiente de TI. A descrição textual do incidente ocorrido é uma situação-problema. Por exemplo: é descrito em uma ocorrência de incidente “Todos os usuários sem acesso aos sistemas da organização”. Esta descrição é a situação-problema manifestada através do “mau” funcionamento de um servidor de banco de dados que apresentou problema em uma porta de rede. Outro exemplo: a central de serviços recebe a ocorrência de incidente “Sem conexão com as pastas de rede”, sendo esta a situação-problema manifestada através do “não” funcionamento do servidor de arquivos que apresentou problema na fonte de alimentação de energia.

Figura 25 - Processo de tomada de decisão para o diagnóstico inicial utilizando RBR




Fonte: Autor deste trabalho (2015).

Por sua vez, o primeiro passo que o integrante da equipe de suporte precisa realizar para a resolução do incidente de TI é **identificar a situação-problema** da ocorrência de incidente relatada pelo usuário, passo referente à fase de inteligência, de acordo com a Figura 25. Caso esta descrição não esteja tão compreensível para o integrante da equipe de suporte, é necessário entrar em contato com o usuário e solicitar mais informações (OGC, 2011a).

Com a situação-problema identificada, os próximos passos são: c**onsultar a base de regras correspondente à situação-problema** identificada, que apresentará questionamentos de problemas sobre o funcionamento ou configuração de algum *hardware*/*software*, e **avaliar este questionamento de problema** – passos referentes à fase de projeto, conforme Figura 25.

Esta consulta será realizada através de *software*, utilizando RBR. Ao consultar a base de regras da situação-problema, se apresentará informações, por exemplo: (i) o questionamento de um problema "o cabo de rede está conectado ao computador?"; e (ii) alternativas (sim ou não) para responder este questionamento. Estas informações devem ser avaliadas pelo

integrante da equipe de suporte para que em seguida se selecione a melhor alternativa, correspondendo ao próximo passo **escolher a alternativa do questionamento de problema -** referente à fase escolha (Figura 25).

Caso a alternativa escolhida seja correspondida (igual à condição da regra *se*) a conclusão *então* será acionada. Por exemplo, temos a seguinte regra estabelecida para a situação-problema: *se* cabo = "não" *então* = "conecte o cabo de rede". Presumindo que a alternativa escolhida seja "não" a conclusão da regra apresentando a solução "conecte o cabo de rede" é acionada, e armazenada na memória de trabalho do sistema de regras para ser apresentada posteriormente. Neste caso, ficou constatado um problema de incidente "cabo de rede não conectado ao computador".

Sendo verificada a existência de mais regras para a situação-problema em questão, os passos **avaliar questionamento** e **escolher alternativa do questionamento de problema** serão novamente efetuados (ver Figura 25).

Quando não existirem mais regras a serem avaliadas, a memória de trabalho do software de regras, existindo conclusões acionadas, apresenta todas as conclusões das regras que foram armazenadas. Devem, pois, ser informados o problema do incidente e a sua respectiva solução, descrevendo-se a solução ou soluções a serem executadas para se resolver o problema do incidente.

Por conseguinte, com a solução apresentada, ocorrem os passos **executar conclusão (solução) das regras** e **registrar resultado** – correspondente à fase implementação (Figura 25).

O integrante da equipe de suporte, ao finalizar a execução da solução para o incidente, realiza o registro do resultado desta execução, onde será informado e armazenado na base de casos de incidentes: (i) a descrição da situação-problema, (ii) a descrição do problema, (iii) IC (item de configuração) envolvido no incidente, (iv) a descrição da solução do problema com seus passos, (v) status (sucesso ou falha) para resolução do incidente, e (vi) comentários (opcional).

Ao armazenar estas informações em uma base de casos, aumenta-se a quantidade de casos que poderão ser reutilizados para apoiar futuras tomadas de decisões quando utilizarmos o raciocínio por analogia (RBC). Como veremos nas próximas atividades para resolução de incidentes.

No registro do resultado, para não apresentação de solução (o que pode ocorrer devido ao não acionamento de qualquer uma das conclusões das regras), devem ser acrescentados na

ocorrência de incidentes maiores informações da situação-problema e comentários. O intuito é facilitar o entendimento do incidente para as próximas atividades.

Caso não se resolva o incidente, seja por ter obtido falha na execução, ou por não ter sido apresentada solução, deve-se seguir para as próximas atividades de resolução de incidentes de TI: investigação/diagnóstico e resolução/recuperação. Sendo possível, antes de se seguir para as atividades citadas, ocorrer o escalonamento para o 2º nível, como previsto no ITIL.

### 4.3.2 Investigação e Diagnóstico Auxiliados por RBC

As atividades descritas neste item têm o objetivo de guiar o integrante da equipe de suporte para resolução de incidentes de TI na atividade investigação e diagnóstico como previsto no ITIL.

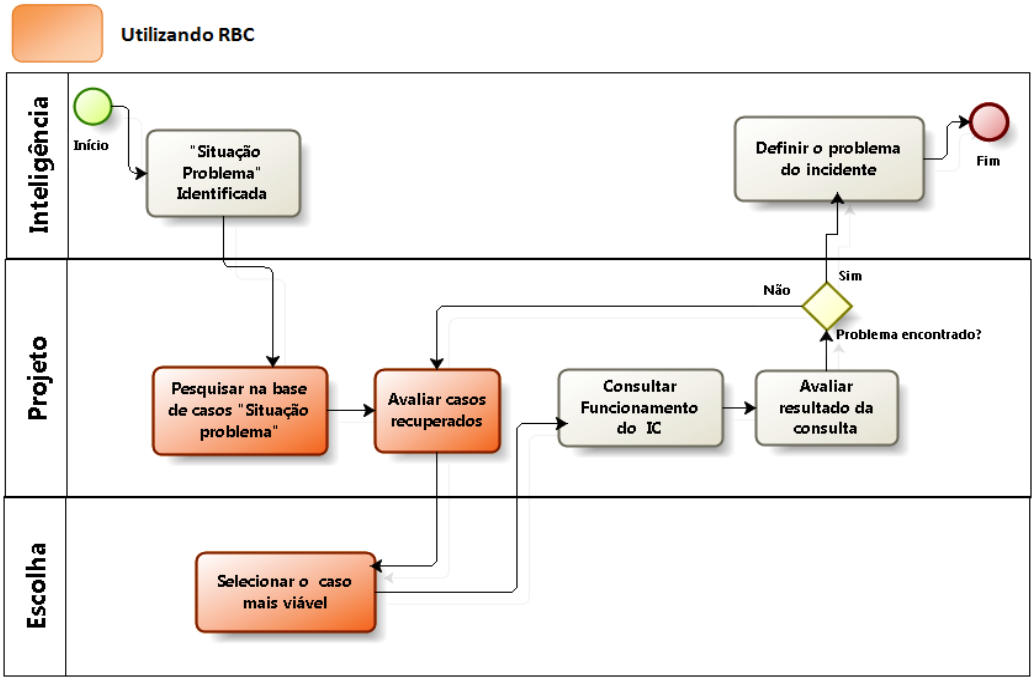
A Figura 26, modelada em BPMN, apresenta o processo de tomada de decisão na resolução de incidente em TI referente à atividade investigação e diagnóstico, que foi construído após a análise da tomada de decisão no domínio das empresas mencionadas anteriormente, respeitando três fases do modelo de decisão de Simon (inteligência, projeto, e escolha) e considerando o apoio do RBC para tomada de decisão.

O integrante da equipe de suporte com a **situação-problema identificada** na atividade diagnóstico inicial**,** correspondendo ao primeiro passo referente à fase de inteligência, deve continuar com os passos necessários para investigar o que ocorreu e diagnosticar o incidente.

Seguindo os passos teremos: **pesquisar na base de casos os casos similares à situação-problema** e **avaliar estes casos recuperados** - passos referentes à fase de projeto, conforme Figura 26.

Esta pesquisa será realizada através de software, baseado em RBC, que recupera da base de casos, os casos similares ao pesquisado, utilizando-se de métodos de cálculo de similaridade e deve exibir: a descrição da situação-problema, a descrição do problema do incidente e o respectivo item de configuração (IC) envolvido, que irão apoiar a tomada de decisão no sentido de identificar o problema do incidente.

Figura 26 - Processo de tomada de decisão para a atividade investigação e diagnóstico utilizando RBC




Fonte: Autor deste trabalho (2015).

Acerca de avaliar os casos recuperados, foi realizado contato com especialistas da área para determinar um limiar (*threshold*) da quantidade de casos a serem recuperados e exibidos para avaliação. Sendo determinado o número desejável para exibição de 10 casos. Desta forma, realizando uma melhor filtragem dos casos recuperados.

Após se avaliarem os casos que mais se aproximam do que foi apresentado, deve-se seguir para o próximo passo referente à fase escolha - **selecionar o caso mais viável**, que será o mais similar ao pesquisado.

Por exemplo: ao se pesquisar na base de casos a seguinte situação-problema "cliente não consegue acessar a intranet", são recuperados casos semelhantes. Para efeito de exemplificação, listamos apenas 3 casos:

Caso 1:

a) **Descrição da situação-problema:** "usuário não consegue entrar na intranet";
b) **Descrição do problema do incidente:** "servidor web com a porta de rede 8080[2] bloqueada";
c) **IC envolvido:** "servidor web".

[2] É importante salientar que a porta de rede número 80 é a padrão para comunicação com os servidores de serviço de conteúdo web que gerenciam o conteúdo, por exemplo, da intranet. Porém este serviço pode ser configurado para oferecer o serviço por outra porta, por exemplo, na porta 8080 como apresentado no caso 1.

Caso 2:

a) **Descrição da situação-problema:** “administrador não consegue atualizar o conteúdo da intranet”;
b) **Descrição do problema do incidente:** “atualizar as permissões de segurança no gerenciador de conteúdo web”;
c) **IC envolvido:** “servidor web”.

Caso 3:

a) **Descrição da situação-problema:** “cliente não ingressa nas pastas de rede”;
b) **Descrição do problema do incidente:** “servidor de domínio (AD) com o serviço NTDS desativado”;
c) **IC envolvido:** “servidor AD”.

O integrante da equipe de suporte avalia e seleciona o caso 1, por ser o mais viável, semelhante, ao pesquisado.

Por conseguinte, são realizados os passos **consultar o funcionamento do IC**, envolvido no caso selecionado, e **avaliar o resultado desta consulta** - fase projeto, conforme Figura 26.

É valioso lembrar que, segundo Pinheiro (2014), cada incidente está conectado a um IC (Item de Configuração). Exemplos de IC: servidores, estação de trabalho (*desktop*), roteadores, switches, etc. Pode ser consultado sobre o IC, o status do funcionamento dos serviços deste IC ou uma documentação completa sobre as configurações dele.

Para o caso 1 selecionado, será consultado o funcionamento do serviço de *firewall* do servidor web (IC envolvido). Caso seja encontrado algum problema, como, por exemplo, “porta de rede do serviço de conteúdo web bloqueada”. Assim, o passo de **definir o problema** do incidente é feita - fase inteligência na Figura 26, concluindo-se a atividade de investigar o que ocorreu e estabelecer o diagnóstico do incidente.

Caso não se encontre falha de funcionamento no IC consultado, volta-se ao passo de avaliar casos recuperados, e segue-se o fluxo dos passos até se encontrar e definir o problema (ver Figura 26).

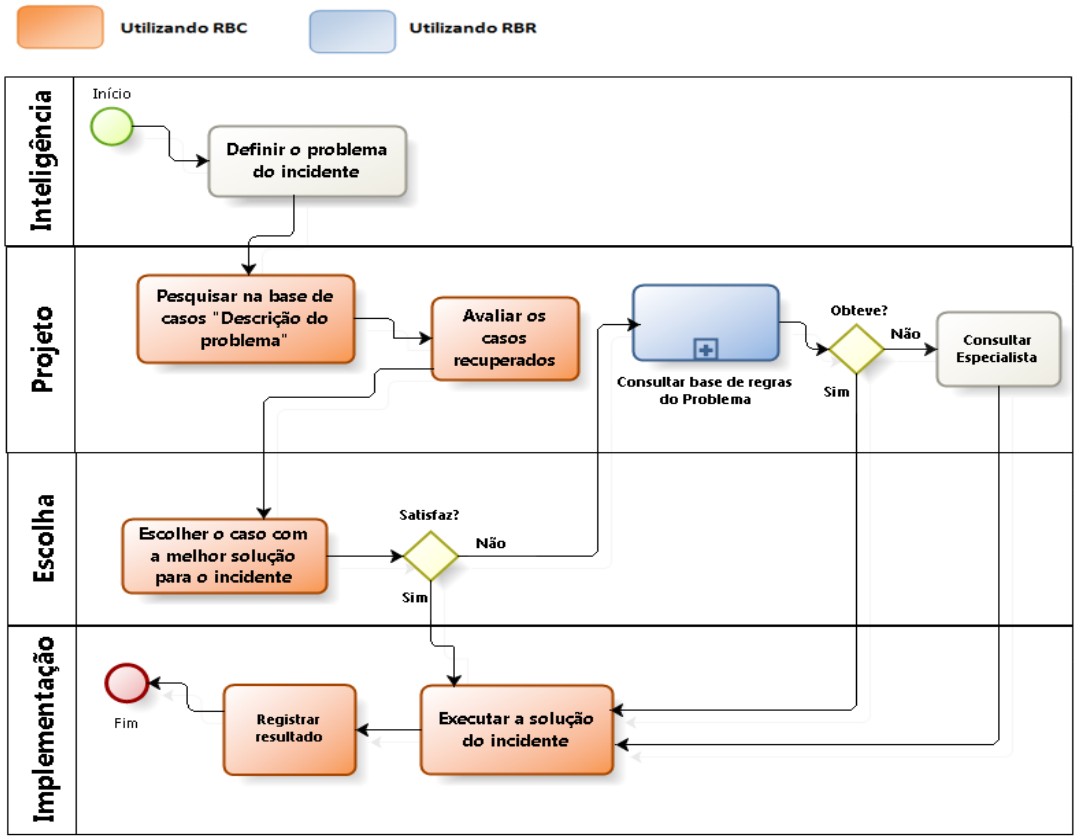
Com a atividade investigação e diagnóstico concluída, deve-se seguir para a próxima atividade: resolução e recuperação.

### 4.3.3 Resolução e Recuperação Auxiliadas pela Combinação RBC com RBR

Os passos descritos neste item têm o objetivo de guiar o integrante da equipe de suporte para a resolução de incidentes de TI na atividade, resolução e recuperação como previsto no ITIL.

A Figura 27, desenvolvida em BPMN, exibe o processo de tomada de decisão na resolução de incidente em TI referente à atividade resolução e recuperação, que foi construído após a análise da tomada de decisão no domínio das empresas aqui mencionadas, respeitando-se as fases do modelo de decisão de Simon e considerando-se o apoio da combinação RBC com RBR para tomada de decisão.

Figura 27 - Processo de tomada de decisão para a atividade resolução e recuperação utilizando RBC com RBR




Fonte: Autor deste trabalho (2015).

Com a **definição do problema do incidente (**fase inteligência) ocorrida na atividade anterior, os próximos passos são: **pesquisar na base de casos,** descrevendo-se o problema definido e **avaliar os casos recuperados** para se identificar a melhor solução para o incidente (fase projeto, conforme Figura 27).

Esta pesquisa também é realizada por *software* baseado em RBC e o resultado deve mostrar casos recuperados similares que irão exibir: descrições do problema do incidente, descrição da solução, passos para execução da solução e os resultados apresentados (status da solução e comentários). Estas descrições devem apoiar a tomada de decisão para se definir qual a melhor solução para o problema novo. Em seguida, deve-se avaliar qual o caso mais similar ao pesquisado. Mais uma vez, é desejável considerarmos o limiar (*threshold*) de 10 casos recuperados similares ao pesquisado. Após avaliação dos casos recuperados, acontece o passo **escolher a melhor solução para o problema do incidente** (fase escolha, conforme Figura 27).

Por exemplo: inserida pelo integrante da equipe de suporte a descrição do problema do incidente "porta web bloqueada" - para corresponder à etapa recuperar do ciclo básico do RBC – casos são apresentados. Para efeito de exemplificação, listamos apenas 3 casos:

Caso 2:

a) **Problema do incidente**: "porta de rede web bloqueada";
b) **Solução**: "reiniciar servidor web (Linux)";
c) **Passos para execução da solução**: "no terminal executar: shutdown –r now";
d) **Resultados apresentados**: "FALHA. Solução não obteve êxito".

Caso 3:

a) **Problema do incidente**: "porta 8080 do servidor web bloqueada";
b) **Solução**: "desbloquear a porta 8080 do apache[3] no servidor web (Linux)";
c) **Passos para execução da solução**: "conectar no Server-web (Linux), apache, e executar o comando: iptables[4] –A INPUT –p tcp –destination-port 8080 –j ACCEPT";
d) **Resultados apresentados**: "Sucesso. Solução resolveu o problema".

Caso 1:

a) **Problema do incidente**: "porta RDP bloqueada";
b) **Solução**: "desbloqueio da porta 3389 no firewall do servidor de terminal remoto";
c) **Passos para execução da solução**: "conectar no Server RDP (Windows) acessar o painel de controle, *firewall*, e liberar porta 3389";
d) **Dos resultados apresentados**: "Sucesso. Solução resolveu o problema".

---

[3] Servidor web de código aberto para sistemas operacionais Linux e Windows. Ver em http://httpd.apache.org/
[4] O *Iptables* é um firewall em nível de pacotes. Ver em http:// iptables.org/projects/iptables/index.html

Supondo que o integrante da equipe de suporte avalia e escolhe o caso 3, por ser o mais viável ao problema de incidente pesquisado e apresentar resultado de sucesso para solução.

Se o integrante da equipe de suporte, tomador de decisão, se der por satisfeito com a solução oferecida e acreditar que aplicando a solução do caso 3 resolva o novo incidente, ocorre o passo de **execução da solução do problema do incidente,** seguindo-se as descrições dos passos para execução da solução e as informações coletadas. Em seguida, é efetuado o passo **registro do resultado** – passos correspondentes à fase implementação da Figura 27.

O resultado é o registro (se obteve sucesso ou falha) para resolução do incidente, juntamente com comentários sobre a solução executada.

Independente do resultado (com ou sem sucesso), os casos devem ser armazenados na base de casos para auxiliar tomada de decisão futura. O novo caso será retido com as informações aprendidas, sendo instanciados os seguintes campos:

a) Descrição da situação-problema;
b) Descrição do problema do incidente;
c) IC envolvido no incidente;
d) Descrição da solução;
e) Descrição dos passos para solução e
f) Resultados obtidos.

Supondo que a solução e os passos para execução da solução apresentada no caso 3 escolhido não satisfaça por completo para solucionar o incidente. O integrante da equipe de suporte então resolve reusar parte da informação recuperada, e percebe a necessidade de obter mais informações para adaptar o caso e resolver o incidente. Neste caso, é possível obter mais informações ao **consultar uma base de regras de problemas** de incidentes, correspondendo a um subprocesso – referente à fase de projeto (ver Figura 27).

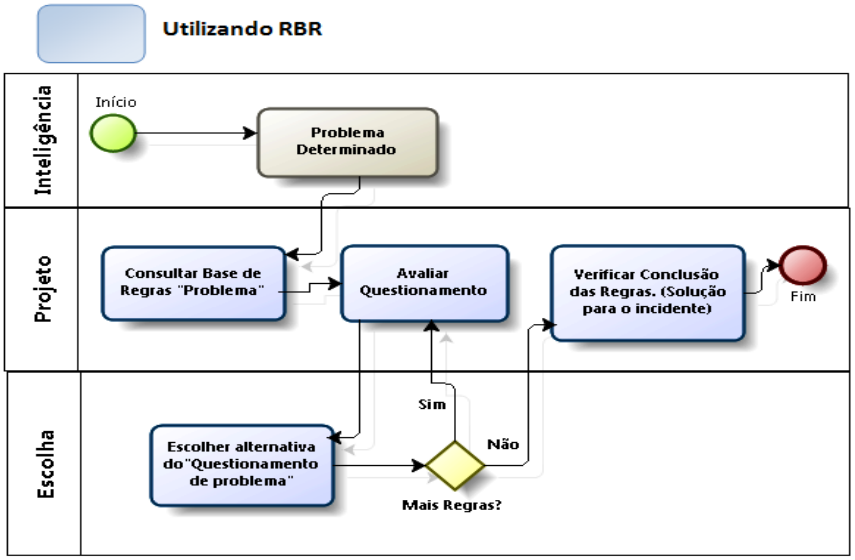
A combinação de RBC com RBR melhora a geração de conhecimento para apoiar decisões. Segundo Eshete (2009), cada um destes raciocínios lida com as limitações do outro e a sua combinação aumenta a precisão no tratamento de problemas.

A Figura 28, modelada em BPMN, apresenta os passos do subprocesso **consultar uma base de regras de problema de incidentes**, que foi construído respeitando-se as três primeiras fases do modelo de decisão de Simon e considerando-se o apoio do RBR para tomada de decisão.

Objetivando obter mais informações para o **problema determinado** – fase de inteligência –, ocorrem os passos **consultar a base de regras do problema** determinado e **avaliar questionamento** - fase de projeto. Em seguida, é efetuado o passo **escolher a**

**alternativa do questionamento de problema** – fase de escolha (Figura 28). Caso existam mais regras, os passos de avaliar questionamento e escolher a alternativa de questionamento de problema são repetidos. Estes passos são iguais aos dos primeiros passos, descritos na Figura 25 (item 4.3.1). Diferenciando que neste momento tratamos de problema e não de situação-problema.

Figura 28 - Consulta base de regras do Problema do incidente




Fonte: Autor deste trabalho (2015).

Esta consulta será realizada através de *software* utilizando RBR. Por exemplo, é consultada a base de regras do seguinte problema “Desbloqueio de porta de rede web”, sendo apresentados dois questionamentos de problema, a saber, (i) “O gerenciador de conteúdo web encontra-se instalado em um Sistema operacional Windows ou Linux?”; e (ii) “Qual o gerenciador de conteúdo web utilizado?”. O integrante da equipe de suporte avalia os questionamentos e escolhe, respectivamente, para cada questionamento, as seguintes alternativas: (i) Windows; e (ii) Apache. Ou seja, o IC com problema (Server web) contém: (i) o sistema operacional Windows e (ii) o gerenciador de conteúdo web Apache.

Seguindo o exemplo, as condições da seguinte regra serão atendidas e a conclusão acionada: *se (condições)* sistema operacional = “Windows” *e* gerenciador web = “Apache” *então (conclusão)* “Desbloqueio no Firewall do Windows da porta 8088”.

Por fim o passo **verificar a conclusão da regra é** acionado, fase de projeto, e efetuado (Figura 28). A conclusão a ser verificada informa “Desbloqueio no Firewall do Windows da porta 8088”.

A melhor solução inicial do caso escolhido anteriormente apresentou: “desbloquear a porta 8080 do apache no servidor web (Linux)”. O tomador de decisão tem neste caso que: (i) o sistema operacional é Linux; e (ii) o gerenciador de conteúdo web é apache. Conclui-se que, o sistema operacional do IC com problema é diferente do caso inicial escolhido (Windows versus Linux) e o gerenciador de conteúdo web apesar de ser o mesmo (Apache) estão configurados em número de portas diferentes (8088 versus 8080). Assim é necessário adaptar a solução inicial apresentada com as novas informações obtidas para solucionar o novo caso.

Dessa forma, através das regras (RBR) é possível obter informações complementares de como solucionar o problema e melhor satisfazer a solução para o incidente.

Finalizado os passos do subprocesso de consultar base de regras do problema e obtido informações complementares capazes de auxiliar na solução do incidente, ocorrem os seguintes passos: **execução da solução do problema do incidente** e o **registro do resultado,** descritos anteriormente (ver Figura 27). Caso não se obtenha informações complementares através das regras para auxiliar na solução do incidente, o integrante da equipe de suporte, antes de realizar estes dois passos, deve efetuar o passo de **consultar um especialista -** fase de projeto (Figura 27). O objetivo é obter, com o especialista, informações que auxiliem na solução do incidente.

## 4.4 ABORDAGEM METODOLÓGICA - PROPOSTA FINAL

Nesta seção, apresentaremos na integra a proposta final da abordagem metodológica para resolução de incidentes de TI.

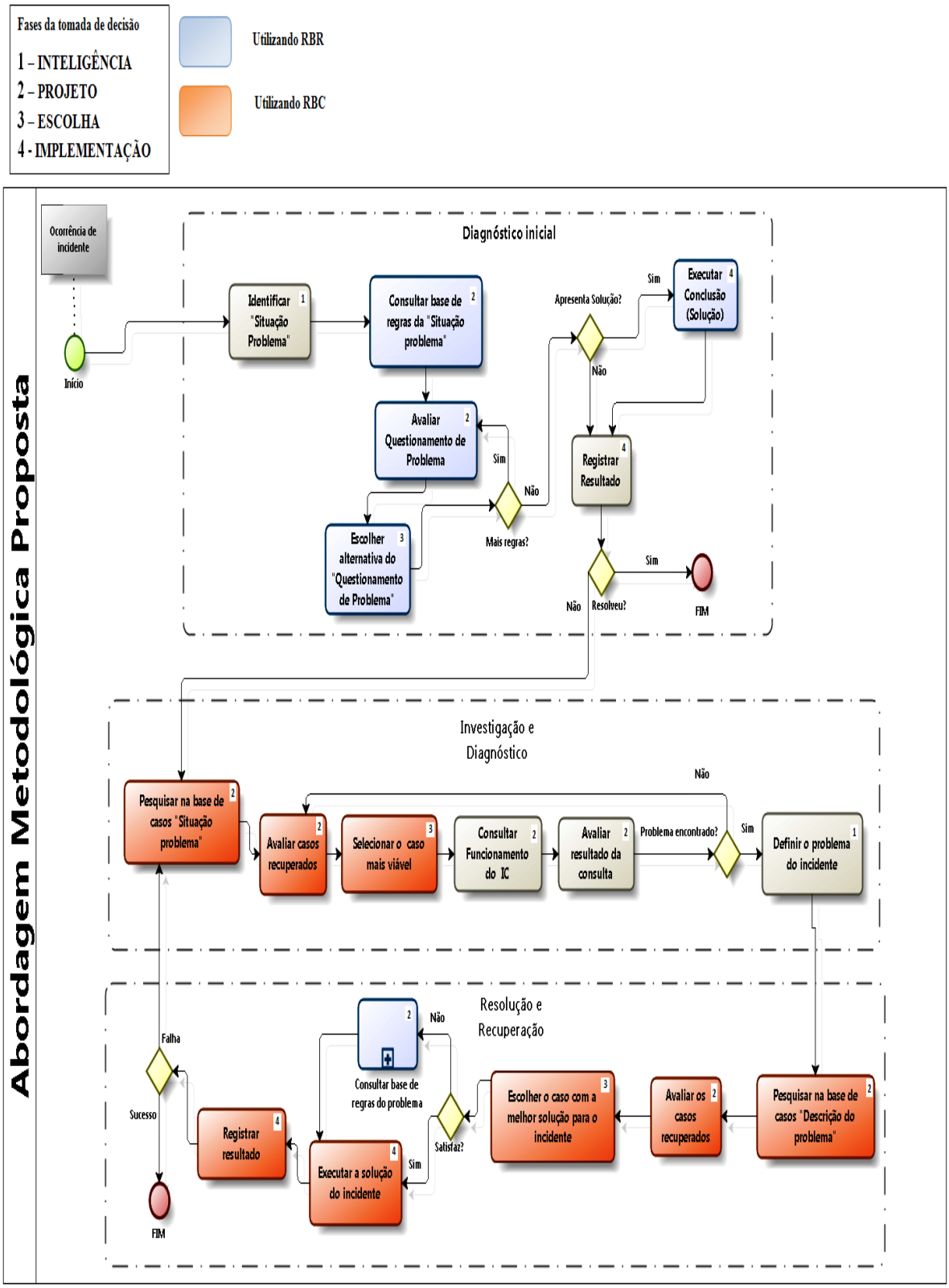
A Figura 29 apresenta o processo completo de tomada de decisão para resolução de incidentes em TI, proposto neste trabalho. São apresentados os passos, descritos na seção anterior, agrupados de acordo com as principais atividades para resolução de incidentes de TI, e numerados de acordo com as quatro fases da teoria da decisão de Simon.

As atividades para resolução de incidente é inicializada quando se identifica a ocorrência de um incidente. A primeira atividade para diagnosticar e tentar reparar imediatamente o serviço que sofreu incidente é o diagnóstico inicial. Se no final dos passos presentes nesta atividade, o integrante da equipe de suporte obtiver êxito com a reparação do serviço, as atividades são finalizadas.

Caso contrário, deve-se realizar a atividade de investigação e diagnóstico seguindo os passos indicados, para entender o que está errado e diagnosticar o problema do incidente. Em seguida é preciso efetuar a atividade de resolução e recuperação, guiada pelos passos

apresentados, com o objetivo de identificar uma solução e executá-la para solucionar o incidente. Se no final destas atividades, o integrante da equipe de suporte obtiver sucesso na resolução do incidente, as atividades serão finalizadas; quando não, uma nova investigação e diagnóstico, resolução e recuperação serão realizadas.

Figura 29 - Atividades completas para Abordagem metodológica proposta




Fonte: Autor deste trabalho (2015).

# 5 ESTUDO DE CASO APLICADO À RESOLUÇÃO DE INCIDENTES DE TI

Neste capítulo, será apresentado na Seção 5.1, o resultado de um questionário aplicado para avaliar a compreensão e aceitabilidade da abordagem proposta. Na Seção 5.2, serão exibidos os artefatos utilizados para o desenvolvimento de um protótipo que utiliza RBR e RBC para apoiar as tomadas de decisões para o domínio de resolução de incidentes de TI. Em seguida, na Seção 5.3, será apresentado um estudo de caso com casos de incidentes coletados em uma organização, nos quais aplicaremos a Abordagem Metodológica proposta com o auxílio do protótipo desenvolvido. O intuito deste estudo de caso é avaliar esta abordagem. Por fim, na Seção 5.4, serão descritas as considerações finais.

O ambiente escolhido foi uma organização privada que presta serviço social aos comerciários da Bahia, mais especificamente na Assessoria de Tecnologia da Informação. A estrutura da Assessoria é composta por coordenadores de suporte e sistemas, uma equipe de técnicos de suporte de 1º e de 2º nível e uma equipe de analistas de suporte e de sistemas de 1º e 2º nível.

A organização possui uma central administrativa localizada na capital e mais 17 unidades: onze na capital baiana e seis no interior. Todas unidades estão interligadas por circuito de dados de fibra ótica.

Na central administrativa, concentram-se a equipe técnica responsável pelo suporte de TI da empresa e os principais servidores e recursos de rede tais como, servidores de banco de dados, e de arquivos, web, e-mail, controlador de domínio.

A infraestrutura disponibilizada para este estudo de caso apresentou-se na forma de um ambiente composto de: (i) um sistema informatizado no qual os usuários descrevem as ocorrências de incidentes vivenciadas; (ii) um servidor de rede para alocação da base de casos e base de regras, além da instalação do protótipo; (iii) computadores em rede para os técnicos e analistas acessarem toda a infraestrutura dos serviços de TI da organização.

## 5.1 QUESTIONÁRIO APLICADO PARA AVALIAR A ABORDAGEM METODOLÓGICA PROPOSTA

O objetivo da aplicação do questionário é avaliar com especialistas da área de suporte o grau de aceitabilidade e da compreensão da abordagem proposta.

Esta etapa foi efetuada por meio de entrevistas e questionários, dotados de material previamente preparado para esclarecer detalhes da abordagem proposta. O questionário aplicado encontra-se no Apêndice A desta dissertação.

As questões objetivaram trazer à tona situações de problemas vivenciados pelos entrevistados, e saber se os participantes conheciam as técnicas de inteligência artificial (RBC/RBR) apresentadas e contempladas em nossa pesquisa, além de verificar se os especialistas concordam com a estratégia adotada para resolução de incidentes em TI.

Responderam ao questionário quatro especialistas da área de suporte. Dois dos participantes trabalham em organizações privadas e dois em organizações públicas. Três deles são técnicos de suporte do 1º nível e um é analista de suporte do 2º nível. Os participantes tinham entre 5 a 10 anos de experiência na área de suporte.

Todos os participantes entenderam as técnicas aplicadas na abordagem. Além disso, declararam que a combinação das técnicas pode apoiar a resolução de incidentes de TI, e que aplicariam a proposta na organização em que trabalham. Três deles informaram que a abordagem atende de forma integral, enquanto que um deles julga que ela colabora de forma parcial para resolução de incidentes.

Acerca das sugestões, um deles considerou a inclusão de pesquisar em sites confiáveis da internet informações adicionais para adaptar soluções incompletas retornadas. Porém, devido ao tempo para conclusão deste trabalho, não foi possível acatar esta sugestão, sendo incluída em trabalhos futuros.

## 5.2 ARTEFATOS PARA O DESENVOLVIMENTO DO PROTÓTIPO

Aqui, serão apresentados os artefatos produzidos para a criação do protótipo, a saber: casos de uso, projeto arquitetural e tecnologias utilizadas.

O protótipo tem o objetivo de auxiliar os usuários que são os integrantes das equipes de suporte do 1º nível e também do 2º em atividades de resolução de incidentes em TI.

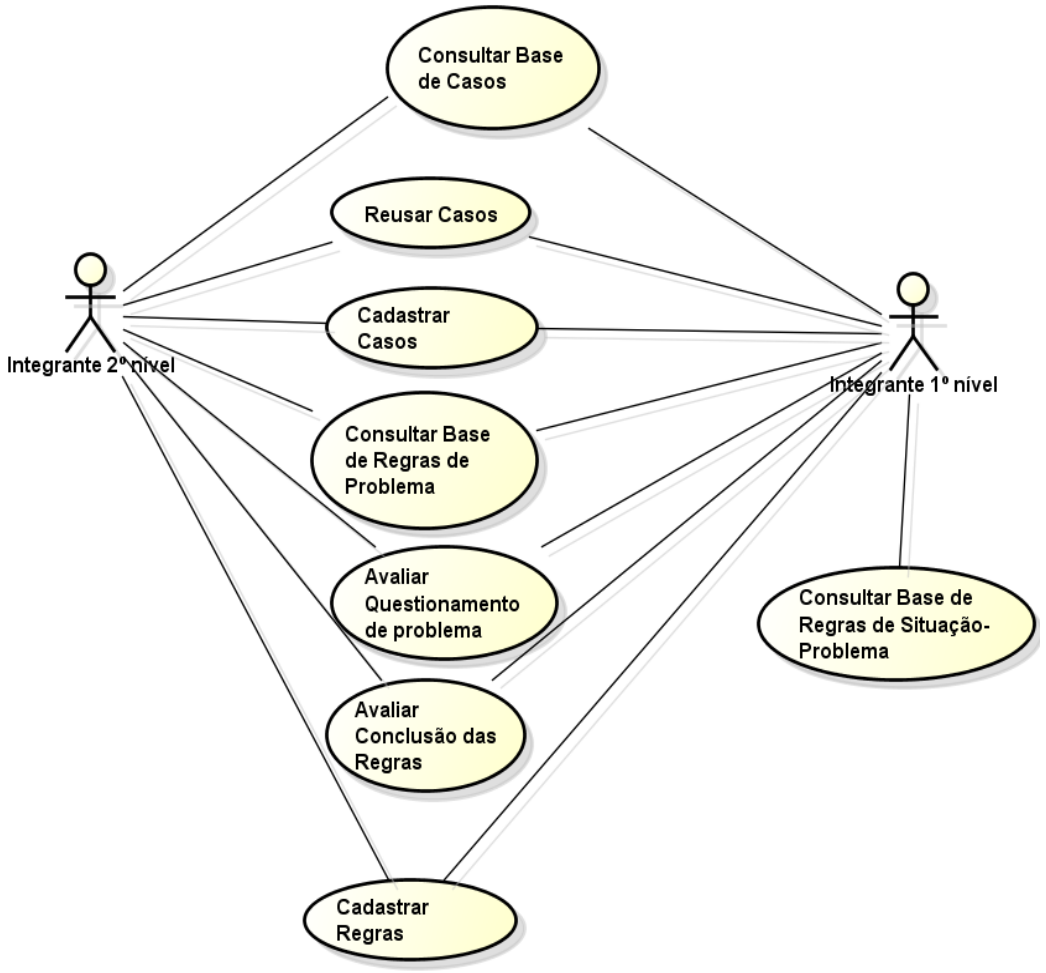
### 5.2.1 Casos de Uso

Na Figura 30, está ilustrado o diagrama de casos de uso da UML em que se descreve a interação entre os atores e o protótipo.

O protótipo possui dois atores: (i) integrante do 2º nível de suporte; e (ii) integrante do 1º nível de suporte. O integrante do 1º nível utiliza as mesmas funções do integrante do 2º nível, e a funçã0 de consultar a base de regras de situação-problema.

No caso de uso *Consultar Base de Casos*, deve ser descrito uma situação-problema ou um problema de incidente para ser consultado na base de casos e uma lista de casos similares deve ser apresentada. Compatível com a fase recuperar do ciclo básico do RBC.

Figura 30 – Diagrama de casos de uso do protótipo




Fonte: Autor deste trabalho (2015).

Em *Reusar Casos*, os casos recuperados serão avaliados pelos integrantes da equipe de suporte, que em seguida podem selecionar o caso mais viável.

No caso de uso *Cadastrar Casos*, novos casos serão inseridos na base de casos para auxiliarem tomadas de decisões futuras. Equivalente a fase reter do ciclo básico do RBC.

Em *Consultar Base de Regras de Problema*, os integrantes da equipe de suporte consultam estas bases para obter informações complementares sobre um determinado problema de incidente.

Em *Avaliar Questionamento de problema*, as bases de regras ao serem consultados apresentam questionamentos de problema que devem ser avaliadas e respondidas, selecionando uma alternativa.

*Avaliar conclusão das regras* finalizam todos os questionamentos de problema e todas as regras que tiveram suas condições satisfeitas apresentarão suas conclusões (soluções) que serão avaliadas pelos integrantes da equipe de suporte com a finalidade de aplicar as soluções apresentadas.

No caso de uso *Cadastrar Regras*, novas regras com o conhecimento geral do especialista podem ser inseridas.

Em *Consultar Base de Regras de Situação-Problema*, o integrante de suporte do 1º nível, responsável por realizar a atividade diagnóstico inicial, consulta estas bases de regras com o intuito de obter informações e solucionar o incidente.

### 5.2.2 Projeto Arquitetural

O projeto arquitetural para o protótipo apresenta-se nas três camadas clássicas (apresentação, negócio e dados) em conformidade com a Figura 31.

A camada de apresentação, detém *interfaces de recomendação* entre os especialistas do ambiente (integrantes da equipe de suporte 1º e 2º níveis) e aplicação do negócio, onde serão inseridos e apresentados os dados para tomada de decisão.

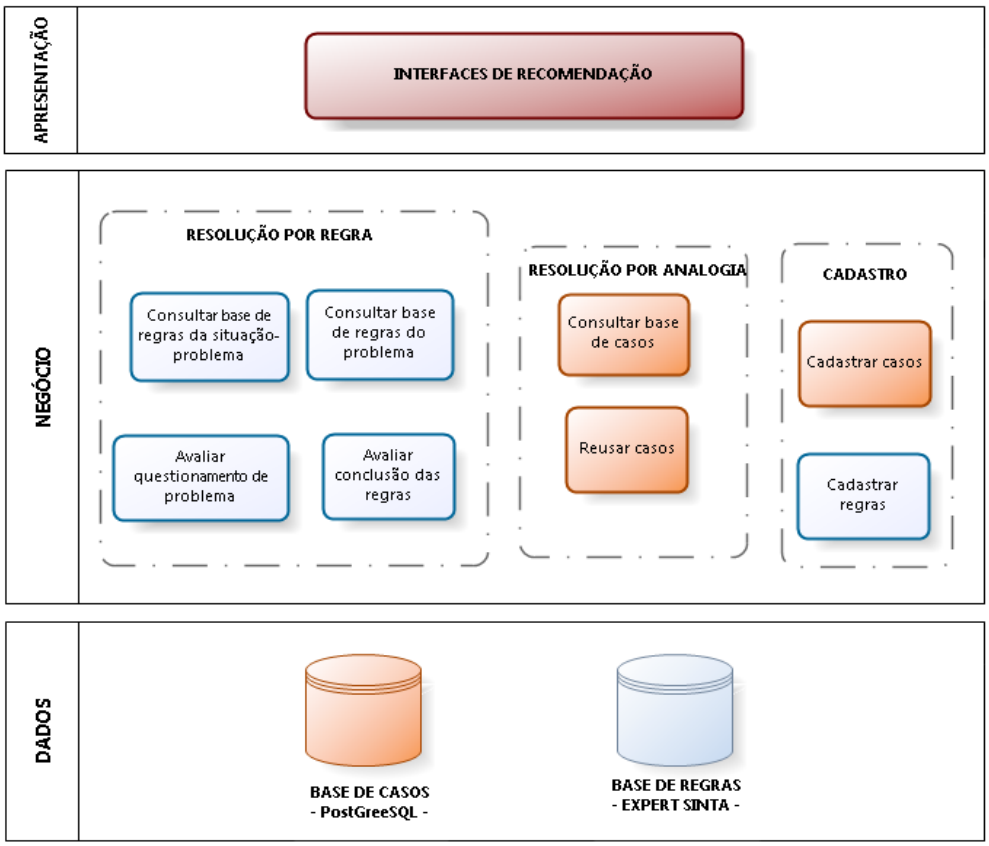
Na camada de negócio encontram-se três módulos para realizarem as funcionalidades do protótipo:

a) **Módulo de cadastro:** gerencia a manutenção dos casos e das regras para a resolução de incidentes de TI, permitindo cadastrar informações sobre casos e regras;

b) **Módulo de resolução por regra:** Este módulo tem o objetivo de disponibilizar o raciocínio por regra para auxiliar nas tomadas de decisões. Ele é composto por quatro submódulos: (i) consultar base de regras da situação-problema; (ii) consultar base de regras do problema; (iii) avaliar questionamento de problema; e (iv) avaliar conclusão das regras;

c) **Módulo de resolução por analogia:** tem objetivo equivalente ao anterior, sendo responsável pelo raciocínio por analogia. É composto por dois submódulos: (i) consultar base de casos e (ii) recuperar casos.

Figura 31 – Arquitetura do protótipo




Fonte: Autor deste trabalho (2015).

Na terceira camada, localiza-se a infraestrutura para o armazenamento persistente de dados e conhecimentos sobre incidentes. A camada é composta de duas bases distintas:

a) **Base de casos**: que contém casos de resolução de incidentes de TI efetuados pelos integrantes da equipe de suporte, sendo implementado no sistema de gerenciamento de banco de dados PostgreSQL;

b) **Base de regras**: que armazena as informações cadastradas por um especialista, contendo informações e procedimentos para solucionar problemas habituais no domínio de resolução de incidentes de TI. Foi implementada no *framework* ExpertSinta que será apresentado na seção a seguir.

### 5.2.3 Tecnologias utilizadas

O protótipo foi implementado utilizando a linguagem Java e o software Sinta. A seguir, as justificativas pelas escolhas destas tecnologias:

a) Ferramentas de desenvolvimento gratuitas, como o eclipse *IDE* para o desenvolvimento em Java e o Expert Sinta;
b) Disponibilidade de *framework* para implementação do sistema de raciocínio baseado em casos – jColibri – e para o sistema especialista baseado em regras – Expert Sinta.

#### 5.2.3.1 JCOLIBRI

O Jcolibri[5] é um *framework open-source*, desenvolvido na linguagem Java e mantido pelo GAIA (*Group For Artificial Intelligence Applications*) da Universidade Complutense de Madrid. O objetivo do Jcolibri é disponibilizar uma plataforma para concepção e geração de sistemas de Raciocínio Baseado em Casos (GARCÍA, 2005).

Este *framework* é implementado na linguagem Java e tem alto potencial para as seguintes expansões: pesquisas textuais, buscas semânticas, formatação de casos utilizando-se técnicas de extração de informação, entre outras vantagens (BELLO TOMÁS, 2004).

Jcolibri possui uma série de classes abstratas e interfaces gráficas que permitem e facilitam a criação de aplicações de RBC. Ele define uma arquitetura para projetar sistemas RBC, além de uma implementação de referência dos componentes necessários para poderem ser construídos. Esse *framework* baseia-se no ciclo básico do RBC e permite: (i) estruturar a base de casos; (ii) definir os casos; e (iii) definir as funções de similaridade global e local.

GAIA disponibiliza o Jcolibri para download e uma grande documentação sobre o *framework* tais como, tutorial, guia de instalação, artigos acadêmicos e exemplo de código.

Para o desenvolvimento do raciocínio por analogia do protótipo foi utilizado este *framework* com o intuito de auxiliar nas tomadas de decisões de resolução de incidentes de TI. Assim, foi necessário modelar o caso nesse domínio, definir os índices e os atributos, analisar as funções de similaridade locais, ajustar os pesos para cada atributo e, por fim, alimentar a base com o conjunto de casos iniciais composto por incidentes resolvidos pelos integrantes da equipe de suporte.

---

[5] http://gaia.fdi.ucm.es/research/colibri/jcolibri

No Apêndice B é apresentado o código fonte do protótipo para o módulo de resolução por analogia.

#### 5.2.3.2 Expert Sinta

O Expert SINTA[6] é uma ferramenta computacional que utiliza técnicas de Inteligência Artificial para geração automática de sistemas com a utilização de regras. Ele foi desenvolvido pelo Laboratório de Inteligência Artificial (LIA) da Universidade Federal do Ceará (Brasil). Esta ferramenta utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades, tendo como objetivo principal simplificar o trabalho de implementação de sistemas baseado em regras através do uso de uma máquina de inferência compartilhada.

A ferramenta tem como objetivo principal simplificar ao máximo as etapas de criação de um sistema baseado em regras, sem a necessidade de conhecimento em programação, facilitando assim o desenvolvimento deste tipo de sistema.

As características principais dessa ferramenta são: (i) utilização do encadeamento para trás (*backward chaining*); (ii) utilização de fatores de confiança; (iii) ferramentas de depuração; e (iv) possibilidade de incluir ajudas on-line para cada base.

O funcionamento básico do Expert SINTA ocorre da seguinte forma: o projetista do conhecimento prepara uma sequência de perguntas (assertivas) e define as regras para cada pergunta no modelo “se” (condição) “então” (conclusão). As perguntas são apresentadas através de uma interface para o tomador de decisão, com algumas opções para respostas: se a resposta corresponder com a condição, a conclusão é disparada para a memória de trabalho. Ao se finalizarem todos os questionamentos das regras, exibem-se todas as conclusões armazenadas na memória de trabalho.

De acordo com Spirandelli e Lennon (2011), o software Expert SINTA proporciona uma imensa facilidade com sua programação e a sua linguagem é adaptável às pessoas sem o conhecimento na área de softwares, podendo ser utilizada em diversos domínios. Para o desenvolvimento do raciocínio por regra do protótipo foi utilizada esta ferramenta. No Apêndice C é apresentado partes das regras construídas.

---

[6] http://www.lia.ufc.br/

## 5.3 APLICANDO A ABORDAGEM METODOLÓGICA PROPOSTA PARA RESOLUÇÃO DE INCIDENTES DE TI

Esta seção realiza um estudo de caso prático da aplicação da Abordagem Metodológica proposta para apoio à decisão de resolução de incidentes de TI.

O objetivo é avaliar o trabalho proposto e demonstrar, por meio de quatro casos selecionados de incidentes de TI, as principais ideias e contribuições discutidas no Capítulo 4.

Os casos selecionados são:

Caso 1 – Usuário informa que ele está sem acesso a alguns sistemas da organização. É resolvido com o raciocínio por regras (RBR) na atividade diagnóstico inicial;

Caso 2 – Usuário informa que todos os usuários da unidade estão sem acesso aos sistemas. É resolvido com o raciocínio por analogia (RBC);

Caso 3 – Usuário informa que não consegue ingressar nas suas pastas de rede. É resolvido com o raciocínio por analogia (RBC);

Caso 4 – Usuário informa que não é possível atualizar a página da intranet da empresa. Este caso foi resolvido com a combinação do raciocínio por analogia (RBC) com o raciocínio por regra (RBR).

### 5.3.1 Caso 1 - Sem acesso aos sistemas financeiro e recibos

Este caso 1 foi resolvido com a atividade diagnóstico inicial, executando os passos: (1) identificar situação-problema; (2) consultar base de casos; (3) avaliar questionamento de problema; (4) escolher alternativa do questionamento de problema; (5) executar conclusão (solução); e (6) registrar resultado. Caso existam mais regras a serem questionadas, os passos 2 e 3 se repetem. O passo 5 será executado caso as regras apresentem conclusão.

Considera-se o caso do usuário da organização “João Ricardo Santos”, de uma unidade do interior, que ao tentar acessar o sistema financeiro e o sistema de recibos do seu computador na organização, não obtêm êxito.

Este usuário descreve uma ocorrência de incidente, através de uma ferramenta web com as percepções vivenciadas por ele e, para este caso 1, foi descrita a seguinte ocorrência de incidente: “Estou sem acesso aos sistemas financeiro e recibos” (ver Figura 32).

Figura 32 - Ocorrência de incidente 1 - aberta por ferramenta web

| Problema | Contato Telefone \| Ramal | Unidade | Status |
|---|---|---|---|
| | **Joao Ricardo Santos** 5689 | **BARREIRAS** Estou sem acesso aos sistemas financeiro e recibos. | Em atendimento |

Fonte: Autor deste trabalho (2015)..

É valioso lembrar que caso esta descrição, realizada pelo usuário, não esteja precisa para o entendimento da situação-problema pelo integrante da equipe de suporte responsável, este deve entrar em contato com o usuário para mais esclarecimentos (OGC, 2011a).

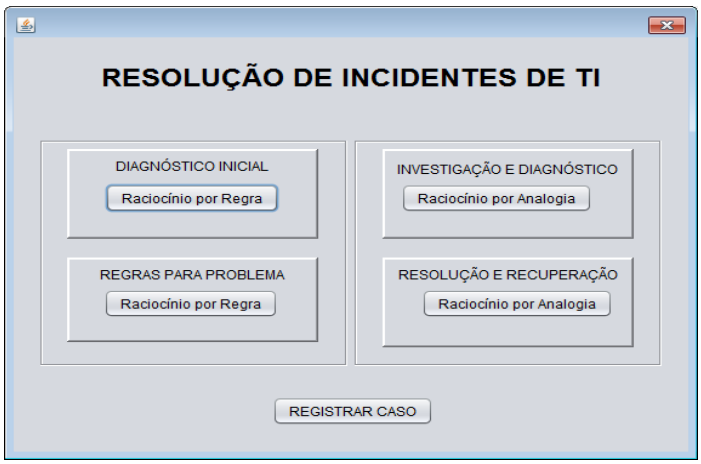
Seguindo a abordagem proposta, a primeira atividade com o objetivo de solucionar o incidente é o diagnóstico inicial que define como primeiro passo identificar a situação-problema, referente à fase de inteligência. A **situação-problema identificada** aqui foi "Sem acesso aos sistemas".

Os próximos passos serão auxiliados pelo protótipo. Com o intuito de facilitar a usabilidade do protótipo e o acionamento dos sistemas de raciocínio por regras ou analogia, foi desenvolvida na linguagem Java uma interface de apresentação das suas funcionalidades, constituída das seguintes informações:

a) **Identificação do tipo de raciocínio e função:** apresenta as funções de resolução de incidente de TI que podem ser acionadas pelo integrante da equipe de suporte juntamente com escolha do tipo de raciocínio a ser empregado (regra ou analogia) para auxiliar na tomada de decisão (Figura 33);

b) **Registrar caso**: para acionar o formulário de cadastro de um novo caso, uma nova experiência aprendida (Figura 33).

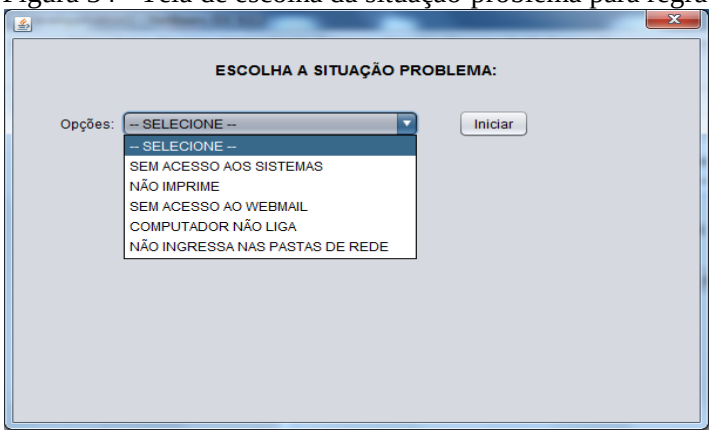
Figura 33 – Tela inicial do protótipo




Fonte: Autor deste trabalho (2015).

Seguindo a abordagem, o próximo passo é **consultar a base de regras da situação-problema** – referente à fase projeto. O protótipo, então, é acionado pelo integrante da equipe de suporte que escolhe a atividade **diagnóstico inicial** e aciona o módulo do raciocínio por regra clicando no botão Raciocínio por Regra (Figura 33) e, em seguida, seleciona a base de regras da situação-problema "Sem acesso aos Sistemas", a ser consultada, e clica no botão iniciar, conforme ilustrado na Figura 34.

Figura 34 - Tela de escolha da situação-problema para regra




Fonte: Autor deste trabalho (2015).

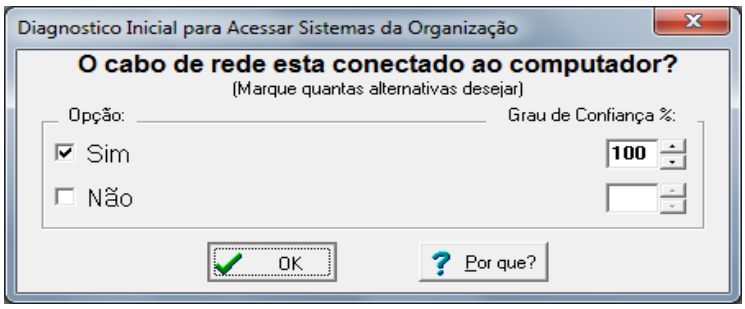
As regras serão apresentadas exibindo-se o questionamento de problema da regra e opções para as respostas. O integrante da equipe de suporte deve efetuar os passos **avaliar o questionamento de problema**, referente à fase de projeto, e **escolher a alternativa do questionamento de problema,** respondendo ao questionamento com a escolha das opções apresentadas – correspondendo à fase de escolha.

Para se efetuarem estes dois passos, neste caso, existem a necessidade de se entrar em contato com o usuário para se verificarem os questionamentos de problemas – fato que pode ocorrer por telefone e/ou acessando-se remotamente o computador que apresentou o incidente.

A Figura 35 apresenta o primeiro questionamento de problema "O cabo de rede está conectado ao computador?", referente à situação-problema selecionada. Para acessar os recursos de rede e, consequentemente, acessar os sistemas da organização, é necessário que o cabo de rede esteja conectado. O integrante, após avaliação com o usuário, escolhe a resposta com a opção "sim" e a conclusão da Regra 1 (*se* cabo = não *então* "conecte o cabo de rede") não é acionada.

As interfaces para as regras foram construídas com a ferramenta Expert Sinta. Estas interfaces permitem informar o grau de confiança da resposta do tomador de decisão, que pode variar de 0 a 100%. Para todos os casos apresentados utilizaremos sempre o grau de confiança padrão fornecido pela ferramenta que é de 100%.

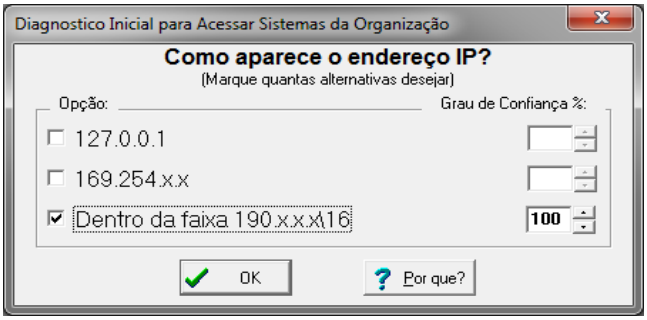
Figura 35 - Regra 1 apresentada




Fonte: Autor deste trabalho (2015).

Em seguida, é efetuado o questionamento de problema “Como aparece o endereço IP?” (Figura 36). A faixa de IP válida para ter acesso aos recursos de rede para esta organização pesquisada deve estar dentro da faixa de ip “190.100.x.x/16”. O integrante avalia o questionamento de problema e escolhe a opção de resposta “Dentro da faixa 190.100.x.x/16”. Com esta resposta, a conclusão da Regra 2 (*se* end_ip = 169.254.x.x ou 127.0.0.1 *então* “Verificar o serviço DHCP do servidor.”) não é acionada.

Figura 36 - Regra 2 apresentada

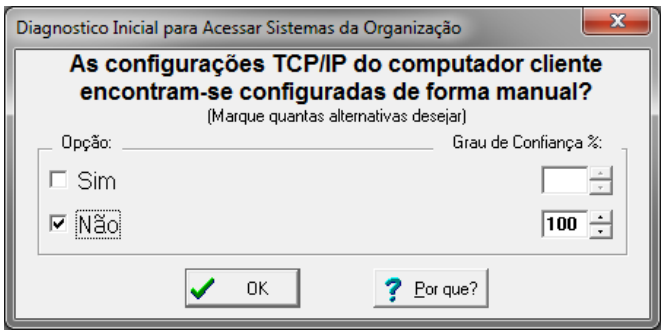



Fonte: Autor deste trabalho (2015).

É apresentado o próximo questionamento de problema “As configurações TCP/IP do computador cliente encontram-se configuradas de forma manual?” (Figura 37). Existe uma política de grupo criada pelo suporte desta organização que caso as configurações TCP/IP do computador estejam configuradas de forma manual, este fica sem acesso aos serviços de rede.

Realizada a avaliação, é escolhida a opção de resposta “Não” e a conclusão da Regra 3 (*se* ip_manual = sim *então* “Configurar IP automático nas propriedades TCP/IP”), não é acionada.

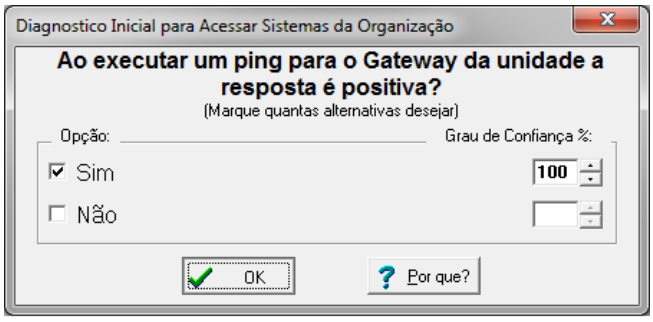
Figura 37 - Regra 3 apresentada




Fonte: Autor deste trabalho (2015).

A Figura 38 apresenta mais um questionamento de problema “Ao executar um ping para o Gateway da unidade, a resposta é positiva?”. Esse teste é realizado para avaliar se há comunicação com o circuito de dados entre a central administrativa onde se encontram os servidores de redes e a unidade. Feito o teste para avaliar o questionamento, é escolhida a resposta “Sim” e a conclusão da Regra 4 (*se* link_dados = não *então* “Reiniciar gateway e entrar em contato com a operadora do link.”) não é acionada.

Figura 38 - Regra 4 apresentada



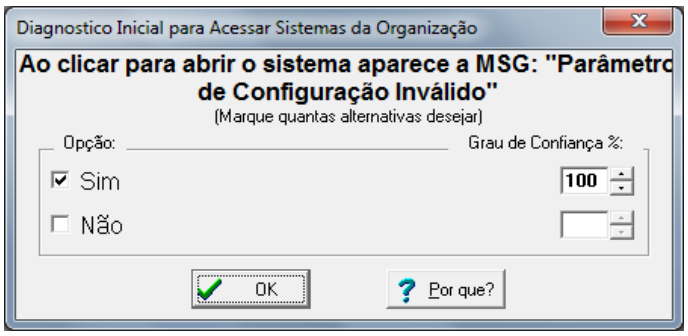

Fonte: Autor deste trabalho (2015).

O próximo questionamento de problema apresenta “Ao clicar para abrir o sistema aparece a Mensagem: Parâmetro de configuração inválido” (Figura 39). Para os computadores clientes terem acesso aos sistemas nesta organização, é necessário estar configurado o IBM/DB2 cliente em todos os computadores.

O integrante da equipe de suporte faz a avaliação e escolhe a opção de resposta “Sim”. Dessa forma, aciona-se a conclusão da Regra 5 (*se* parametro_configuraçao = sim *então*

“Configurar o DB2 cliente para o servidor: srvdb2 e porta de rede: 50000”) que será armazenada na memória de trabalho e será exibida para o tomador de decisão, somente no final dos questionamentos de todas as regras.

Figura 39 - Regra 5 apresentada

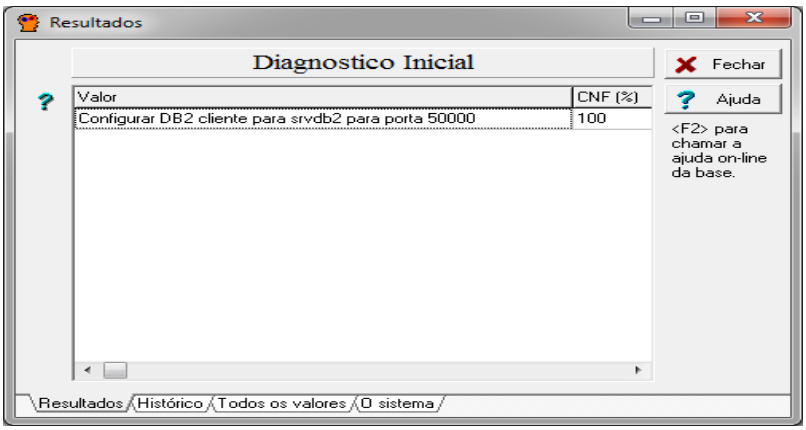



Fonte: Autor deste trabalho (2015).

Para esta situação-problema foram modeladas estas cinco regras apresentadas. Dessa forma, todas as regras foram questionadas e as conclusões, indicando-se a solução para o incidente, serão exibidas, de maneira a apoiar a tomada de decisão do integrante da equipe de suporte.

A Figura 40 apresenta a única conclusão que foi acionada das regras questionadas para este caso pesquisado, com a seguinte descrição de solução para o incidente: “Configurar o DB2 cliente para o servidor: srvdb2 e porta de rede: 50000”.

Figura 40 - Conclusão com solução para o incidente

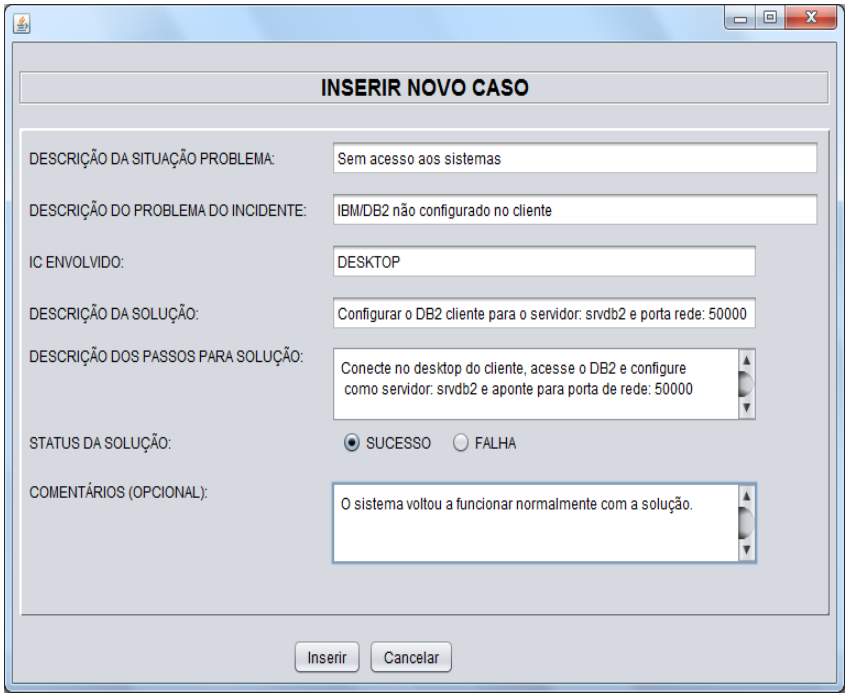



Fonte: Autor deste trabalho (2015).

Como foi apresentada solução, o próximo passo é **executar a conclusão das regras** – referente à fase de implementação. O integrante da equipe de suporte pode conectar remotamente no computador do cliente e executar a conclusão indicada.

Por fim, é realizado o passo **registar resultado** – fase de implementação, e este deve ser armazenado na base de casos. Serão instanciados os seguintes campos (Figura 41):

Figura 41- Registro do caso 1 na base de casos




Fonte: Autor deste trabalho (2015).

O objetivo de armazenar estas informações do caso é para que futuramente elas possam ser usadas para solucionarem novos casos similares.

Se obtiver sucesso com a solução, o incidente é fechado. Se não obtiver sucesso com a execução da solução ou não tiver sido apresentada conclusão (solução), deve-se seguir para as próximas atividades: (i) investigação e diagnóstico; e (ii) resolução e recuperação, como previsto no ITIL e nesta abordagem.

### 5.3.2 Caso 2 - Sem acesso aos sistemas da organização

O caso 2 foi resolvido com as atividades investigação e diagnóstico, e resolução e recuperação, executando os passos: (1) identificar situação-problema; (2) pesquisar na base de casos por situação-problema; (3) avaliar casos recuperados; (4) selecionar o caso mais viável; (5) consultar funcionamento do IC; (6) avaliar resultado da consulta do IC; (7) definir o problema do incidente; (8) pesquisar na base de casos por problema; (9) avaliar casos recuperados; (10) escolher o caso com a melhor solução para o incidente; (11) consultar base de regras de problema; (12) executar a solução do incidente; (13) registrar resultado.

Caso não tenha encontrado o problema do incidente após o passo 6, os passos 3,4 e 5 se repetem. O passo 11 será executado no momento em que a solução do caso escolhido não satisfaça completamente a resolução do incidente.

Considera-se o caso do usuário “Franklin” – lotado em uma unidade do interior (Paulo Afonso - BA), que verifica que todos os usuários da sua unidade não conseguem acessar os sistemas da organização.

Este usuário descreve uma ocorrência de incidente, através de uma ferramenta web, com as percepções vivenciadas por ele. Para este caso 2, foi descrita a seguinte ocorrência de incidente “Todos os usuários desta unidade estão sem acesso aos sistemas da organização. Favor verificar” (ver Figura 42).

Figura 42 – Ocorrência de incidente 2

| Problema | Contato Telefone \| Ramal | Unidade | Status |
|---|---|---|---|
| | **Franklin** 8656 | **PAULO AFONSO** Todos os usuários desta unidade estão sem acesso aos sistemas da organização. Favor verificar. | Em atendimento |

Fonte: Autor deste trabalho (2015).

Assim, foi realizada, inicialmente, a atividade diagnóstico inicial com o raciocínio por regra, como em 5.3.1 e nenhuma conclusão (solução) foi apresentada. O integrante da equipe de suporte do 1º nível verificou que não é mais da responsabilidade dele o incidente descrito e resolve escalar o incidente para o 2º nível, mais especificamente para o suporte técnico.

O integrante de suporte do 2º nível passa a realizar a próxima atividade prevista para a resolução do incidente de TI (investigação e diagnóstico).

Prontamente, este integrante **identifica a situação-problema** referente à fase de inteligência “Sem acesso aos sistemas”, correspondendo ao primeiro passo.

Os próximos passos aconteceram com o apoio do raciocínio por analogia. Para isso, ele vai iniciar o protótipo e escolher a atividade investigação e diagnóstico, clicando no botão (raciocínio por analogia) desta atividade.

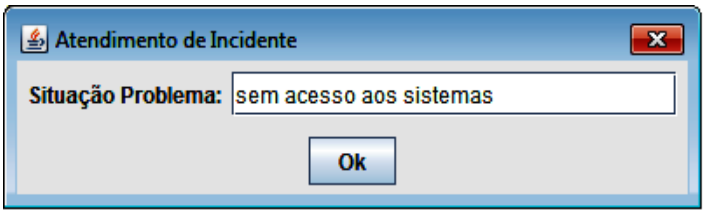
Seguindo-se a abordagem proposta, teremos os passos **pesquisar na base de casos a situação-problema** e **avaliar estes casos recuperados –** passos referentes à fase de projeto.

Para isso, a investigação é inicializada inserindo-se no campo situação-problema a pesquisa “sem acesso aos sistemas” (ver Figura 43). Em seguida, o sistema RBC realiza o seguinte procedimento: utilizando-se do algoritmo *Nearest Neighbour* do *JColibri* (função de similaridade *maxstring* e peso (1.0) para os atributos), recupera e apresenta os casos mais

semelhantes armazenados na base de casos, ordenando-os por grau de similaridade, (Figura 44). Todas as recuperações apresentadas neste trabalho utilizaram este procedimento.

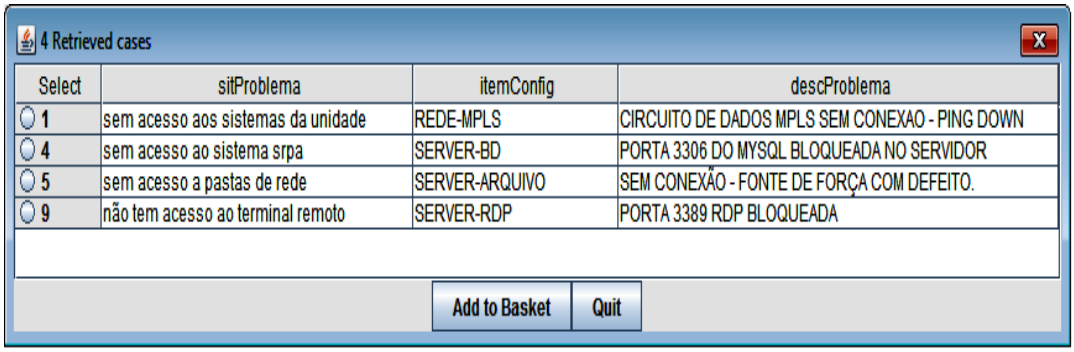
As interfaces foram construídas em Java, sendo compostas das seguintes telas: (i) situação-problema: permite inserir a descrição da situação-problema que deverá ser pesquisada por similaridade na base de casos (Figura 43); e (ii) casos recuperados da situação-problema: apresentam os casos mais similares à situação-problema, ordenados por grau de similaridade, exibindo a identificação (id) do caso, a situação-problema, o item de configuração envolvido e a descrição do problema do incidente (Figura 44).

Figura 43 - Pesquisa situação-problema




Fonte: Autor deste trabalho (2015).

Figura 44 - Resultado da recuperação da situação-problema




Fonte: Autor deste trabalho (2015).

Para demonstrar o cálculo de similaridade global, exibe-se a memória de cálculo para os casos apresentados na Figura 44.

*SimGlobal*(consulta, caso(1)) = ((23/34)*1)/1= 0,676 (comparando as *strings* foi determinado 23 caracteres – “sem acesso aos sistemas” – sendo a maior *string* a do caso de id (1), com 34 caracteres);

*SimGlobal*(consulta, caso(4)) = ((10/26)*1)/1= 0,384 (comparando as *strings* foi determinado 10 caracteres – “sem acesso” – sendo a maior *string* a do caso de id (4), com 26 caracteres);

*SimGlobal*(consulta, caso(5)) = ((10/27)*1)/1= 0,370 (comparando as *strings* foi determinado 10 caracteres – “sem acesso” – sendo a maior *string* a do caso de id (5), com 27 caracteres);

*SimGlobal*(consulta, caso(9)) = ((6/33)*1)/1= 0,181 (comparando as *strings* foi determinado 6 caracteres – “acesso” – sendo a maior *string* – caso de id (9), com 33 caracteres);

Inicialmente, o integrante da equipe de suporte, tomador de decisão, avalia os casos recuperados e seleciona um caso – caso de id (1) - por se aproximar mais ao que foi pesquisado, realizando-se, pois, o passo de **selecionar o caso mais viável** – fase de escolha. Neste exemplo, é apontado que o problema pode estar no item de configuração (IC) REDE-MPLS (circuito de dados para conectar a unidade do interior à central administrativa).

Em seguida, são realizados os passos **consultar funcionamento do IC** e **avaliar resultado da consulta** – fase de projeto. Esta consulta tem o intuito de obter informações de funcionamento deste item de configuração envolvido. No exemplo, estas podem ser auxiliadas por uma ferramenta de monitoramento de infraestrutura de TI. Observando-se e avaliando-se o resultado, percebe-se que o circuito de dados da unidade do interior Paulo Afonso (Figura 45) encontra-se com suas operações normais - Ping ok - indicando-se que o item de configuração não tem problema de funcionamento.

Figura 45 - Consulta informação do IC (circuito de dados de Paulo Afonso)

| Host | Service | Status |
|---|---|---|
| MPLS_PauloAfonso | PING | OK |

Fonte: Autor deste trabalho (2015).

Assim, como indicado na abordagem metodológica proposta, o integrante da equipe de suporte retorna ao passo de **avaliar os casos recuperados.** Após a segunda avaliação dos casos, ele **seleciona outro caso** que mais se aproxime do pesquisado – caso id (4) (ver Figura 44). No exemplo, ficou indicado que o problema pode estar no IC, SERVER-BD (servidor de banco de dados da empresa) e a **consulta do funcionamento deste IC**, com o auxílio da ferramenta de monitoramento, indica um problema de funcionamento (*critical*) na porta MySql (ver Figura 46). **Avaliando-se o resultado desta consulta**, chega-se ao diagnóstico de que o problema do incidente é na porta MySql do SERVER-BD.

Figura 46 - Consulta informações do IC (SERVER_BD)

| Host | Service | Status | Last Check | Duration | Attempt |
|---|---|---|---|---|---|
| SERVER_BD | Current Load | OK | 11-14-2014 16:36:41 | 29d 22h 47m 28s | 1/4 |
| | Current Users | OK | 11-14-2014 16:36:23 | 29d 22h 42m 46s | 1/4 |
| | HTTP | OK | 11-14-2014 16:36:50 | 29d 22h 47m 19s | 1/4 |
| | MYSQL | CRITICAL | 11-14-2014 16:37:18 | 29d 4h 55m 51s | 3/3 |
| | PING | OK | 11-14-2014 16:36:32 | 29d 22h 42m 37s | 1/4 |
| | SSH | OK | 11-14-2014 16:36:40 | 29d 22h 42m 29s | 1/4 |
| | Swap Usage | OK | 11-14-2014 16:36:53 | 29d 22h 47m 2s | 1/4 |
| | Total Processes | OK | 11-14-2014 16:36:49 | 29d 22h 42m 20s | 1/4 |

Fonte: Autor deste trabalho (2015).

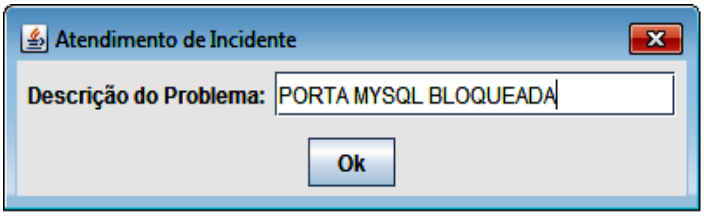
Como o problema do incidente foi encontrado, é efetuado o passo de **definir o problema do incidente** “Porta Mysql bloqueada” – referente à fase inteligência e finaliza-se, assim, a atividade investigação e diagnóstico.

A seguir, é necessário efetuar a atividade resolução e recuperação. O integrante da equipe de suporte acessa a tela inicial do protótipo e clica no botão raciocínio por analogia da atividade resolução e recuperação (ver Figura 33). O objetivo aqui é encontrar uma solução para o incidente e recuperar o serviço de TI.

Para esta atividade, são executados, inicialmente, os passos de **pesquisar na base de casos** os casos similares, descrevendo-se o problema definido e **avaliar os casos recuperados** para identificar a melhor solução para o incidente - fase projeto-. Esta pesquisa ocorre também através de *software* baseado em RBC, Jcolibri.

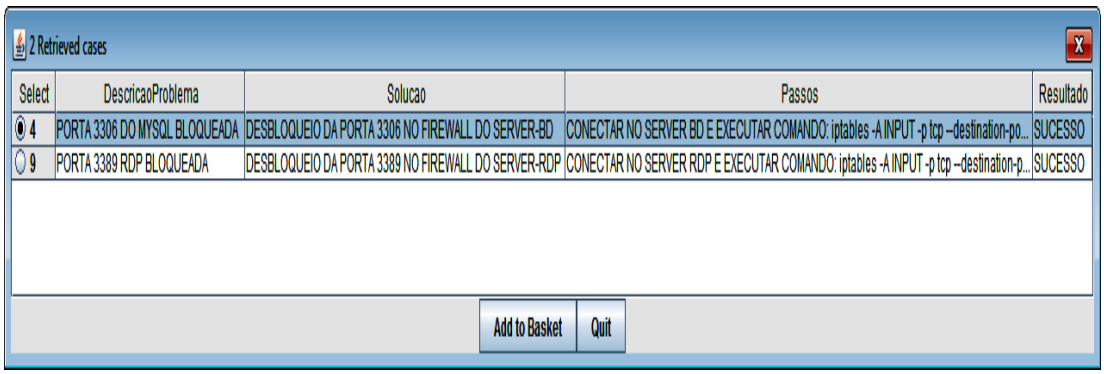
Para estes passos é inserida para pesquisa no *software* RBC a seguinte descrição para o campo descrição do problema: “PORTA MYSQL BLOQUEADA” (ver Figura 47). Em seguida ocorre a recuperação dos casos na base de casos para serem avaliados, apresentando-se: (i) descrição do problema de incidente; (ii) descrição da solução para o incidente; (iii) respectivos passos para execução; e (iv) resultados com aplicação da solução proposta (sucesso ou falha), conforme Figura 48. O atributo comentários previstos no sistema é opcional. Para estes dois casos recuperados este atributo não foi inserido. A tela da figura 47 mencionada foi desenvolvida em Java.

Figura 47 - Consulta problemas similares




Fonte: Autor deste trabalho (2015).

Figura 48 – Resultado da recuperação para descrição do problema



2 Retrieved cases

| Select | DescricaoProblema | Solucao | Passos | Resultado |
|---|---|---|---|---|
| 4 | PORTA 3306 DO MYSQL BLOQUEADA | DESBLOQUEIO DA PORTA 3306 NO FIREWALL DO SERVER-BD | CONECTAR NO SERVER BD E EXECUTAR COMANDO: iptables -A INPUT -p tcp --destination-po... | SUCESSO |
| 9 | PORTA 3389 RDP BLOQUEADA | DESBLOQUEIO DA PORTA 3389 NO FIREWALL DO SERVER-RDP | CONECTAR NO SERVER RDP E EXECUTAR COMANDO: iptables -A INPUT -p tcp --destination-p... | SUCESSO |

Add to Basket | Quit

Fonte: Autor deste trabalho (2015).

Para demonstrar o cálculo de similaridade global, exibe-se a memória de cálculo para os casos apresentados na Figura 48.

*SimGlobal*(consulta, caso(4)) = ((21/29)*1)/1= 0,724 (comparando as *strings* foi determinado 21 caracteres – “PORTA MYSQL BLOQUEADA” – sendo a maior *string* a do caso de id (4), com 29 caracteres); e

*SimGlobal*(consulta, caso(9)) = ((15/24)*1)/1= 0,625 (comparando as *strings* foi determinado 15 caracteres – “PORTA BLOQUEADA” – sendo a maior *string* a do caso de id (4), com 24 caracteres);

Ao avaliar os casos recuperados (Figura 48), o integrante da equipe de suporte realiza o próximo passo **escolher o caso com a melhor solução para o incidente –** fase de escolha. É escolhido o caso de id (4) com o intuito de reusar-lhe a solução, visto que a descrição do problema apresentado é equivalente ao do pesquisado e o campo resultado informa o status de sucesso para a solução executada.

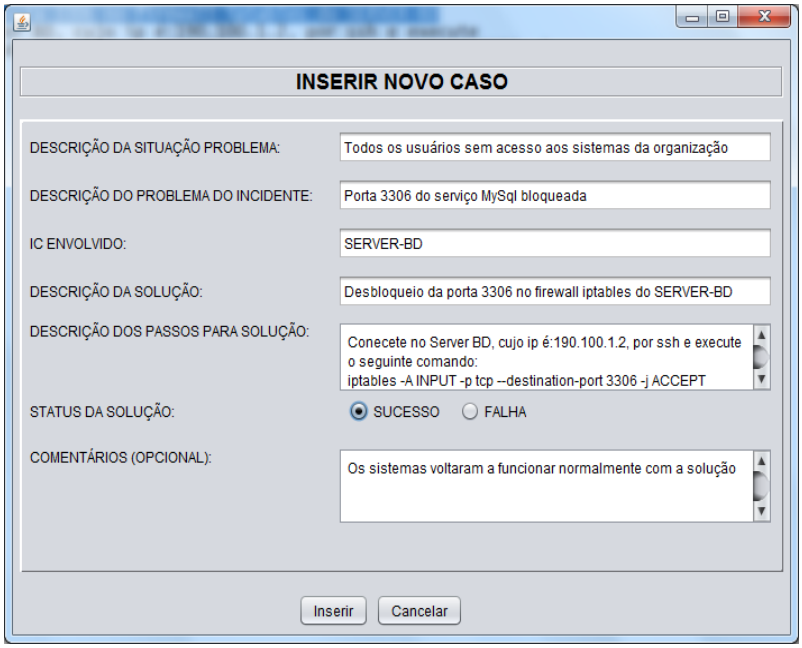
Seguindo os passos previstos, devem-se efetuar os passos **executar a solução do incidente** e **registrar resultado** – referentes à fase de implementação.

A solução é executada com o intuito de resolver e recuperar o serviço de TI, seguindo a descrição dos passos do caso selecionado (Figura 48), que orienta efetuar o seguinte “Conectar no SERVER BD e executar o comando iptables –A INPUT –p tcp –destination-port 3306 –j ACCEPT”. Este servidor de banco de dados encontra-se em um servidor com sistema operacional Linux e com o firewall iptables ativo. Este comando serve para liberar no firewall a porta 3306, referente à porta de comunicação de rede do Serviço de Gerenciamento de Banco de Dados MySql.

Ao se executar esta solução proposta é obtido êxito e o incidente é solucionado. Portanto, o serviço de TI passa a operar normalmente. Por fim, o passo **registrar resultado** é

realizado, sendo esta experiência retida na base de casos com as informações aprendidas. Serão instanciados os campos da base de casos, conforme (Figura 49):

Figura 49 – Registro do novo caso




Fonte: Autor deste trabalho.

Foi observado, ao se reter um novo caso similar a um existente na base de casos, que informações mais detalhadas são inseridas, deixando o novo caso mais esclarecido com relação ao similar. Por exemplo: foi adicionado o ip “190.100.1.2” do SERVER BD e orientada a utilização da ferramenta “ssh” para realizar a conexão com o servidor. Dessa forma, melhorando a tomada de decisão do integrante da equipe de suporte para as futuras resoluções de incidentes.

### 5.3.3 Caso 3 – Não ingressa nas pastas de rede

Considera-se o caso do usuário “Adenilson” – lotado na central administrativa (Salvador - BA), o qual verificou que ao tentar ingressar nas próprias pastas de rede compartilhadas do servidor de arquivos da organização, não obteve êxito.

Este usuário descreve uma ocorrência de incidente através de uma ferramenta web, com as percepções vivenciadas por ele. Para este caso 3, foi descrita a seguinte ocorrência de incidente “Não consigo ingressar nas minhas pastas de rede” (ver Figura 50).

Figura 50 - Ocorrência de incidente 3 - aberta por ferramenta web

| N.º | Problema | Contato Telefone \| Ramal | Unidade | Status |
|---|---|---|---|---|
| 6736 | | **CCDM - ASTIN - Adenilson** 8715 | **CCDM** Não consigo ingressar nas minhas pastas de rede. | Em atendimento |

Fonte: Autor deste trabalho (2015).

O integrante da equipe de suporte realiza a atividade diagnóstico inicial. Efetuando o primeiro passo **identificar a situação-problema**, referente à fase de inteligência "Não ingressa nas pastas de rede".

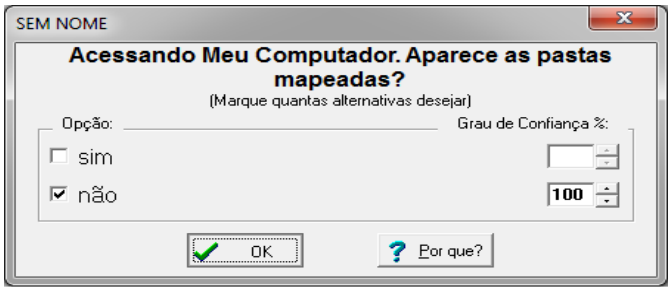
Os próximos passos utilizaram o apoio do raciocínio por regra para tomada de decisão. Dessa forma, o protótipo é iniciado e selecionado o raciocínio por regra, clicando no botão correspondente para atividade diagnóstico inicial. Em seguida, é efetuada a **consulta da base de regras da situação-problema,** passo referente à fase projeto, sendo selecionado para consulta a base de regras da situação-problema "Não ingressa nas pastas de rede".

As regras serão carregadas. O integrante da equipe de suporte efetua os passos **avaliar o questionamento de problema** apresentado – referente à fase de projeto – e **escolher a alternativa do questionamento de problema,** respondendo ao questionamento com a **escolha das alternativas apresentadas –** correspondendo à fase de escolha.

Para este caso, as três primeiras regras e os seus questionamentos de problema com as respectivas respostas foram idênticas ao do caso 1, Seção 5.3.1. Por isso, são descritos os procedimentos adotados para as regras subsequentes.

É apresentado o questionamento de problema "Acessando meu computador. Aparecem as pastas mapeadas?" O padrão de acesso às pastas compartilhadas dentro da organização é acessando a opção "meu computador" do sistema operacional cliente Windows 7. O integrante avalia o questionamento de problema e escolhe a opção "Não" (Figura 51). Com esta resposta a conclusão da regra é acionada - Regra 4 (*se* mapeamento = não *então* "Executar o script de mapeamento de pasta de rede. \\190.100.1.1\script\script.bat.")

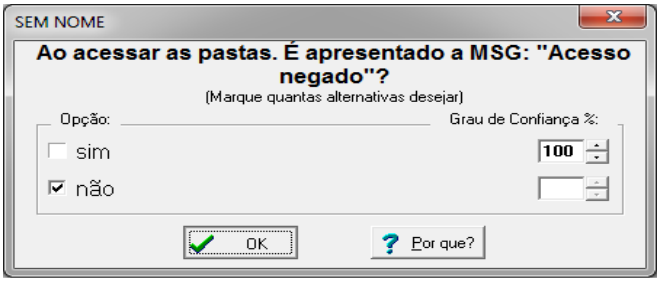
Figura 51 - Regra 4 apresentada




Fonte: Autor deste trabalho (2015).

Por conseguinte, é efetuado o questionando "Ao acessar as pastas, é apresentada a mensagem: Acesso negado?" – se a resposta é igual a "sim" indica que o usuário não tem permissão de segurança para acessar a pasta. Desse modo, o tomador de decisão realiza a avaliação e escolhe a opção "não" (ver Figura 52). A conclusão da regra não é acionada - Regra 5 (*se* permissão = não *então* "Conceder permissão se autorizado.")

Figura 52 - Regra 5 apresentada




Fonte: Autor deste trabalho (2015).

Para esta situação-problema foram modeladas cinco regras. Como todas as regras foram questionadas, a conclusão (solução) será apresentada, com a seguinte descrição de solução: "Executar o *scprit* de mapeamento de pasta de rede. \\190.100.1.1\script\script.bat"

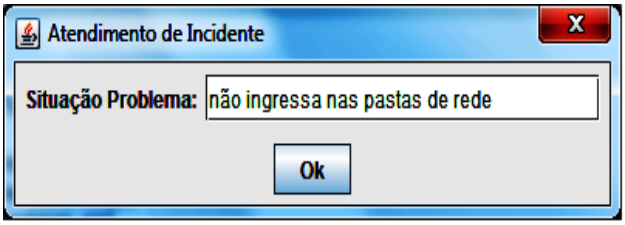
Como foi apresentada solução, o próximo passo é **executar a conclusão das regras** – referente à fase de implementação. O integrante, ao finalizar a execução das soluções apresentadas, consegue mapear as pastas de rede para que o usuário visualize o atalho para as pastas. Porém, o usuário não consegue entrar nas pastas para acessar o conteúdo existente, persistindo o incidente.

Dessa forma, o integrante da equipe de suporte precisa tomar novas decisões. Ele toma a decisão de seguir com as outras atividades para tentar solucionar o incidente e não o escala para outro nível.

O integrante do suporte, dessa forma, passa a realizar a próxima atividade prevista para a resolução do incidente de TI, investigação e diagnóstico. É iniciado o protótipo, e acionado o raciocínio por analogia da atividade investigação e diagnóstico.

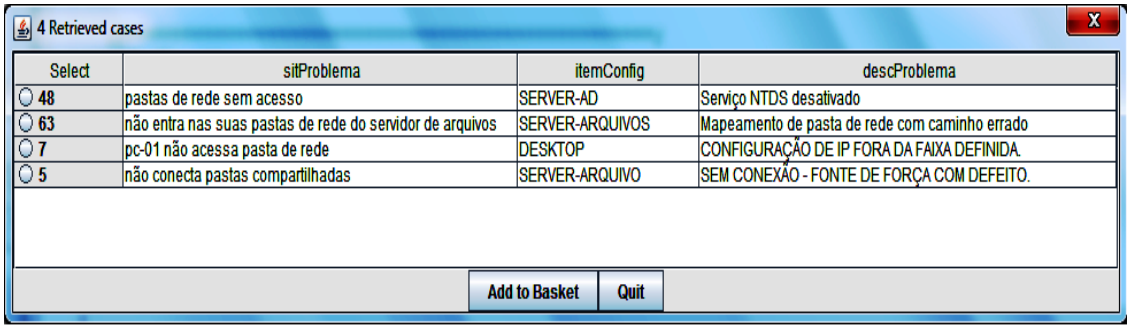
Com a situação-problema identificada na atividade anterior, os próximos passos seguindo o trabalho proposto são: **pesquisar na base de casos a situação-problema** e **avaliar estes casos recuperados –** fase de projeto. Para isso, a investigação é inicializada ao inserir-se no campo situação-problema a pesquisa "Não ingressa nas pastas da rede" (Figura 53). O sistema RBC recupera e apresenta os casos semelhantes armazenados na base de casos, ordenando-os por similaridade (Figura 54).

Figura 53 - Pesquisa situação-problema – caso3




Fonte: Autor deste trabalho (2015).

Figura 54 - Casos recuperados da situação-problema – caso 3




Fonte: Autor deste trabalho (2015).

Para demonstrar o cálculo de similaridade global, exibe-se a memória de cálculo para os casos apresentados na Figura 53.

*SimGlobal*(consulta, caso(48)) = ((14/31)*1)/1= 0,451 (comparando as *strings* foi determinado 14 caracteres – "pastas de rede" – sendo a detentora da maior *string* a da consulta, com 31 caracteres);

*SimGlobal*(consulta, caso(63)) = ((21/57)*1)/1= 0,368 (comparando as *strings* foi determinado 21 caracteres – “não nas pastas de rede” – sendo a maior *string* a do caso de id (63), com 57 caracteres);

*SimGlobal*(consulta, caso(7)) = ((11/31)*1)/1= 0,354 (comparando as *strings* foi determinado 11 caracteres – “não de rede” – sendo a maior *string* – sendo a detentora da maior *string* a da consulta, com 31 caracteres); e

*SimGlobal*(consulta, caso(5)) = ((10/33)*1)/1= 0,303 (comparando as *strings* foi determinado 10 caracteres – “não pastas” – sendo a maior *string* – caso de id (9), com 33 caracteres);

Após se avaliarem os casos recuperados, é tomada a decisão de selecionar o caso de id (48) - por se aproximar mais ao que foi pesquisado, realizando-se o passo **selecionar o caso mais viável** – fase de escolha. É indicado que o problema pode estar no item de configuração (IC) SERVER-AD (Servidor de *Active Directory*, responsável por autenticar os usuários, permitindo acessar os recursos da rede).

Posteriormente, são realizados os passos **consultar funcionamento do IC** e **avaliar resultado da consulta** – fase de projeto. Ao se consultar o serviço NTDS[7] (banco de dados do *Active Directory*) do SERVER-AD, como indicado no caso selecionado, é avaliado que o serviço encontra-se parado (*stopped*) (ver Figura 55).

Figura 55 - Consulta serviço NTDS

Fonte: Autor deste trabalho (2015).

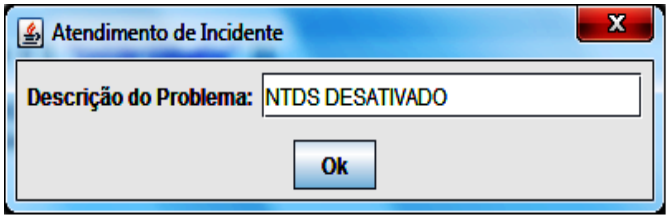
[7] Ver em https://technet.microsoft.com/en-us/library/cc779676(v=ws.10).aspx

O problema do incidente foi encontrado e é efetuado o passo **definir o problema do incidente** “NTDS desativado” – referente à fase inteligência. Desse modo, finaliza-se a atividade investigação e diagnóstico.

A próxima atividade é a resolução e recuperação com o apoio do raciocínio por analogia, acionado na tela principal do protótipo. Seguindo-se os passos propostos, é preciso **pesquisar na base de casos** os casos similares, descrevendo-se o problema definido, e **avaliar os casos recuperados** para se identificar a melhor solução para o incidente - fase projeto. A pesquisa acontece utilizando-se o Jcolibri.

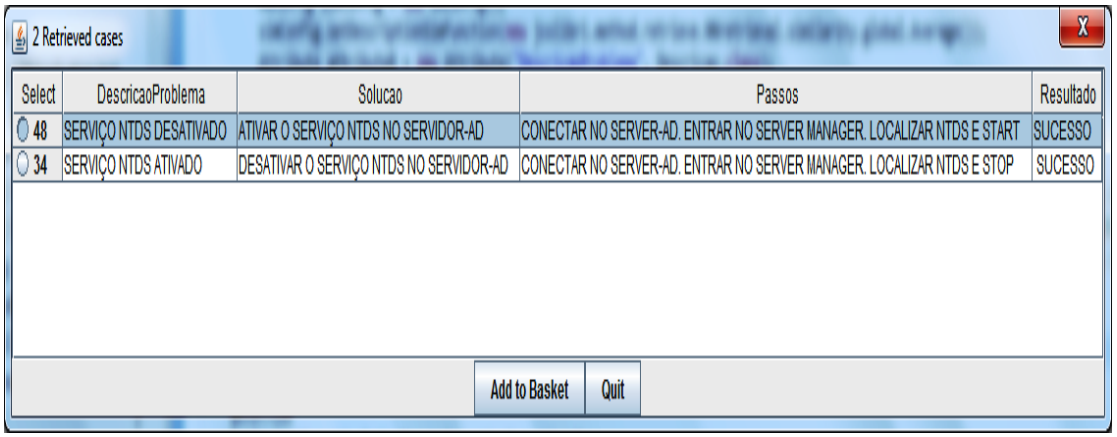
É inserida para pesquisa a seguinte descrição para o campo descrição do problema: “NTDS DESATIVADO” (ver Figura 56). Por conseguinte, ocorre a recuperação dos casos na base de casos para serem avaliados (Figura 57).

Figura 56 – Consulta problemas similares - caso 3




Fonte: Autor deste trabalho (2015).

Figura 57 - Resultado da recuperação para descrição problema - caso 3




Fonte: Autor deste trabalho (2015).

Para demonstrar o cálculo de similaridade global, exibe-se a memória de cálculo para os casos apresentados na Figura 57.

*SimGlobal*(consulta, caso(48)) = ((15/23)*1)/1= 0,652 (comparando as *strings* foi determinado 15 caracteres – “NTDS DESATIVADO” – sendo a maior *string*, o caso de id (48), com 23 caracteres); e

*SimGlobal*(consulta, caso(34)) = ((4/20)*1)/1= 0,200 (comparando as *strings* foi determinado 4 caracteres – “NTDS” – sendo a maior *string*, o caso de id (34), com 20 caracteres);

Ao se avaliarem os casos recuperados (Figura 57), é efetuado o passo **escolher o caso com a melhor solução para o incidente** – fase de escolha. É escolhido o caso de id (48) - com o intuito de reusar-lhe a solução, por se aproximar mais ao pesquisado e apresentar resultado com sucesso.

Em seguida, os passos **executar a solução do incidente** e **registrar resultado** – referente à fase de implementação, são processados. Para isso, o caso indica o seguinte: “CONECTAR NO SERVER-AD. ENTRAR NO SERVER MANAGER. LOCALIZAR O NTDS E START”.

Ao se executar esta solução proposta é obtido sucesso e o incidente solucionado. Para finalizar, o passo de **registrar resultado** é realizado. Serão instanciados os seguintes campos:

a) Descrição da situação-problema: “Não consegue ingressar nas pastas de rede”;
b) Descrição do problema do incidente: “Serviço NTDS desativado”;
c) IC envolvido no incidente: “SERVER-AD”;
d) Descrição da solução: “Iniciar o serviço NTDS”;
e) Descrição dos passos para solução: “Conectar no SERVER-AD, cujo ip é: 190.100.1.1, Clicar em ferramentas administrativas – Server manager. Localizar o serviço NTDS e start.”;
f) Status da solução: SUCESSO;
g) Comentários: *Não foram adicionados comentários*.

Informações mais detalhadas em relação ao caso similar foram inseridas, deixando este novo caso melhor explicado para apoiar uma tomada de decisão futura. Por exemplo: foi adicionado o ip “190.100.1.1” do SERVER AD e orientado como encontrar o serviço NTDS a ser iniciado.

### 5.3.4 Caso 4 – Intranet da organização não atualiza conteúdo

Para este caso, considera-se a abertura de ocorrência realizada pelo usuário Thales – lotado na central administrativa (Salvador - BA) – informando-se que a intranet da empresa não atualiza.

Este usuário descreve uma ocorrência de incidente, através de uma ferramenta web, com as percepções vivenciadas por ele, sendo descrito o seguinte: “Não consigo atualizar a página da intranet” (ver Figura 58).

Figura 58 - Ocorrência de incidente 4 - aberta por ferramenta web

| Problema | Contato<br>Telefone \| Ramal | Unidade | Status |
|---|---|---|---|
| | **THALES FERRAZ OLIVEIRA**<br>1234 | **CCDM**<br>Não consigo atualizar a página da intranet | Em atendimento |

Fonte: Autor deste trabalho (2015).

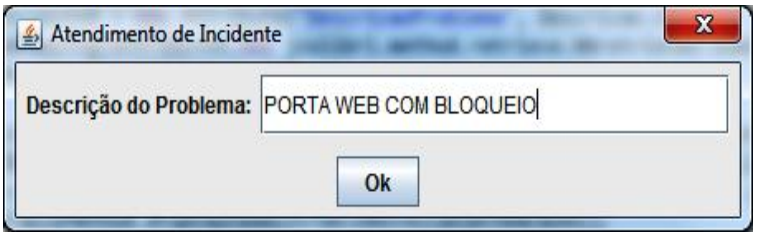
Para este caso, foi efetuada inicialmente a atividade **diagnóstico inicial**. Após a realização de todos os passos presentes nesta atividade, não foi encontrada qualquer solução.

Como não foram apresentadas soluções e o incidente persistiu, o integrante da equipe de suporte continuou com as outras atividades: (i) investigação e diagnóstico e (ii) resolução e recuperação.

Ao efetuar todos os passos presentes para a atividade investigação e diagnóstico, foi diagnosticado o problema do incidente “porta de rede do serviço web bloqueada”.

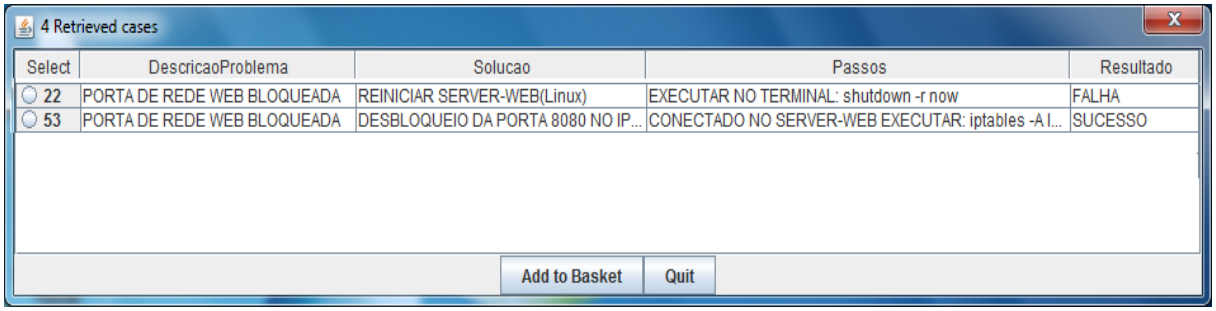
Com o problema definido, a próxima atividade é a resolução e recuperação, que serão apoiadas, incialmente, pelo raciocínio por analogia, acionado na tela principal do protótipo, clicando no botão raciocínio por analogia da atividade resolução e recuperação. Seguindo-se os passos propostos, é realizada a **pesquisa na base de casos**, descrevendo-se o seguinte no campo descrição do problema: “PORTA WEB COM BLOQUEIO” (ver Figura 59). Por conseguinte, ocorre a recuperação de **casos a serem avaliados** (Figura 60). A pesquisa acontece utilizando-se o Jcolibri.

Figura 59 - Consulta problemas similares - caso 4




Fonte: Autor deste trabalho (2015).

Figura 60 - Resultado da recuperação para descrição problema - caso 4




Fonte: Autor deste trabalho (2015).

Para demonstrar o cálculo de similaridade global, exibe-se a memória de cálculo para os casos apresentados na Figura 60.

*SimGlobal*(consulta, caso(22)) = ((9/27)*1)/1= 0,200 (comparando as *strings* foi determinado 9 caracteres – "PORTA WEB" – sendo a maior *string*, o caso de id (22), com 20 caracteres); e

*SimGlobal*(consulta, caso(53)) = ((9/27)*1)/1= 0,200 (comparando as *strings* foi determinado 9 caracteres – "PORTA WEB" – sendo a maior *string*, o caso de id (53), com 20 caracteres);

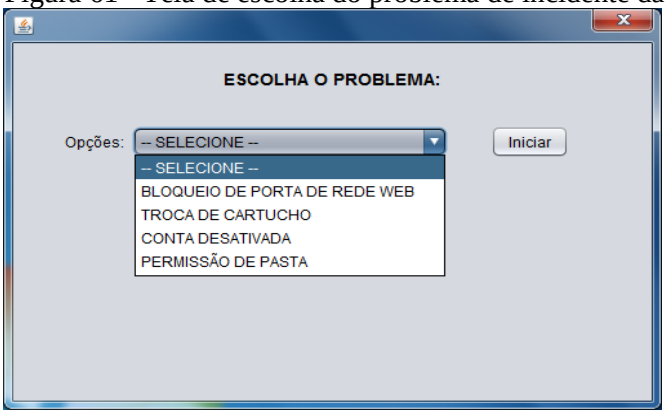
Como o cálculo de similaridade global foi igual (0,200) para ambos os casos, o sistema ordenou de forma ascendente pelo id.

Ao se avaliarem os casos recuperados (Figura 60) é preciso **escolher o caso com a melhor solução para o incidente** – passo da fase de escolha. É escolhido o caso de id (53) que no campo passos, informam os passos para a solução "CONECTADO NO SERVER-WEB EXECUTAR: iptables -A INPUT -p tcp --destination-port 8080 -j ACCEPT. Servidor apache." Este caso, apesar do *Jcolibri* ter ordenado como segundo do *ranking*, é escolhido pelo integrante da equipe de suporte devido ao fato do primeiro – caso de id (22) - apresentar no campo resultado "FALHA" para execução da solução (Figura 60). Assim, o caso id (53) é o mais similar ao pesquisado, tendo a solução registrada com resultado de sucesso.

Por outro lado, a solução escolhida não satisfaz completamente para resolver o novo incidente ocorrido. Logo, o integrante da equipe de suporte observa que o gerenciador de conteúdo web (Apache), do caso de id (53) escolhido, é diferente do gerenciador de conteúdo web (JBOSS[8]) utilizado para gerenciar o conteúdo da intranet. Dessa forma, decide-se que sejam obtidas informações adicionais sobre o **problema do incidente definido** - fase de inteligência. Para tanto, será **consultada uma base de regras de problemas** de incidentes – fase de projeto – contendo um conjunto de passos.

Assim é acionado, na tela inicial do protótipo, o raciocínio por regra das regras para o problema. A base de regras "bloqueio de porta de rede web" é iniciada para consulta (Figura 61), tela desenvolvida em Java.

Figura 61 - Tela de escolha do problema de incidente da regra

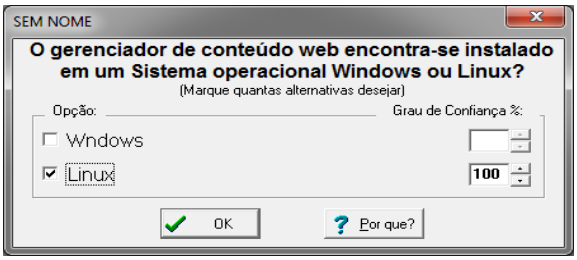



Fonte: Autor deste trabalho (2015).

Após acionar a base de regras escolhida o primeiro questionamento de problema é apresetado "O gerenciador de conteúdo web encontra-se instalado em um Sistema operacional Windows ou Linux?". Ao se **avaliar o questionamento de problema** – fase de projeto, é **escolhida a alternativa do questionamento** "Linux" (Figura 62) – fase de escolha.

[8] JBoss *web server*. Ver em https://www.redhat.com/pt-br/technologies/jboss-middleware/web-server

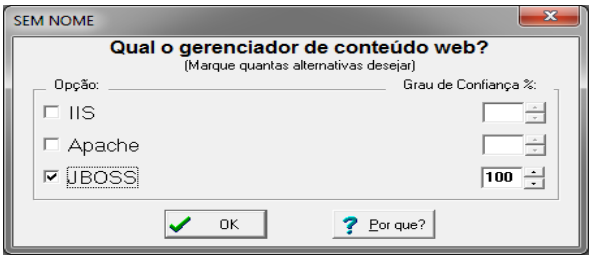
Figura 62 – Questionamento de problema 1




Fonte: Autor deste trabalho (2015).

Mais um questionamento de problema é apresentado “Qual o gerenciador de conteúdo web utilizado?”, e, ao se avaliar o questionamento, é escolhida a alternativa “JBOSS” (Figura 63).

Figura 63 - Questionamento de problema 2

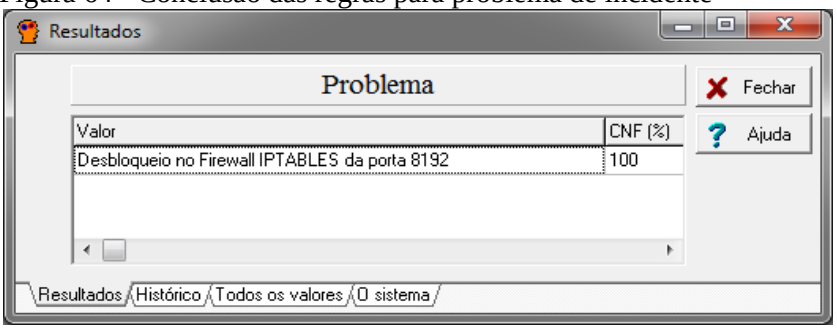



Fonte: Autor deste trabalho (2015).

A condição da Regra 4 (*se* Sistema_Operacional = Linux *E* Gerenciador_Web = JBOSS *então* “Desbloqueio no Firewall IPTABLES da porta 8192”) é satisfeita e a conclusão carregada para memória de trabalho.

Por fim, as conclusões que foram acionadas são apresentadas e **verificadas** – fase de projeto, oferecendo-se informações adicionais para solucionar o incidente “Desbloqueio no Firewall IPTABLES da porta 8192” (Figura 64).

Figura 64 - Conclusão das regras para problema de incidente




Fonte: autor deste trabalho (2015).

O integrante da equipe de suporte verifica que apesar de o sistema operacional (Linux) do caso inicial escolhido ser igual ao do novo caso, o gerenciador de conteúdo web é diferente (Apache versus JBOSS). Sendo assim, é necessário adaptar a solução inicial apresentada com a conclusão da regra exibida com o intuito de solucionar o novo caso.

Para este caso, é preciso mudar o número da porta nos passos informados para solucionar o incidente. A solução inicial escolhida informa o desbloqueio do apache - porta "8080". A conclusão das regras aponta o desbloqueio do JBOSS - porta "8192" (Figura 64).

Adaptando-se a solução inicial, são **executados os passos para a solução** – fase de implementação - "CONECTADO NO SERVER-WEB EXECUTAR: iptables -A INPUT -p tcp --destination-port **8192** -j ACCEPT", sendo solucionado o problema do incidente ocorrido.

Em seguida é **registrado o resultado** da solução executada – fase de implementação. Serão instanciados os seguintes campos:

a) Descrição da situação-problema: "Não atualiza intranet";
b) Descrição do problema do incidente: "Porta de rede web JBOSS bloqueada";
c) IC envolvido no incidente: "SERVER-WEB";
d) Descrição da solução: "Desbloqueio da porta 8192 do serviço JBOSS";
e) Descrição dos passos para solução: "CONECTADO NO SERVER-WEB EXECUTAR: iptables -A INPUT -p tcp --destination-port **8192** -j ACCEPT";
f) Status da solução: SUCESSO;
g) Comentários: "A solução implementada resolveu o problema de acesso à intranet".

A inclusão desse novo caso na base de casos irá auxiliar nas tomadas de decisões futuras e aumentar a base de conhecimento da organização.

## 5.4 CONSIDERAÇÕES FINAIS

Para amparar a avaliação deste trabalho, foi realizado um questionário com especialistas cuja intenção era demonstrar o grau de aceitação e compreensão da abordagem proposta, além da construção de um protótipo que possibilitasse a utilização dos raciocínios baseado em casos e baseado em regras, permitindo assim, demonstrar a utilização do raciocínio por analogia e por regra na resolução de incidentes.

Para avaliar a viabilidade da abordagem proposta, esta foi aplicada em casos de incidentes reais que aconteceram em uma organização pesquisada. Foram escolhidos

diferentes incidentes, com o intuito de garantir possibilidades distintas para resolução dos incidentes seguindo a abordagem proposta.

Para o incidente do caso 1 apresentado, a resolução ocorreu na primeira atividade (diagnóstico inicial), utilizando o raciocínio por regra. Os casos 2 e 3 de incidentes descritos, não obtiveram êxito na primeira atividade para resolução, sendo solucionados os incidentes com as atividades investigação e diagnóstico, e resolução e recuperação, utilizando o raciocínio por analogia.

O incidente do caso 4 foi solucionado utilizando a combinação do RBC com RBR, com as atividades, investigação e diagnóstico, e resolução e recuperação. Para se determinar uma solução, inicialmente, casos foram recuperados utilizando raciocínio por analogia (RBC), tendo a solução de um deles escolhida para solucionar o incidente, porém a solução precisava ser adaptada. Utilizando informações obtidas com o raciocínio por regra (RBR), a solução inicial foi adaptada e o incidente solucionado.

# 6 CONCLUSÃO

Neste capítulo, é apresentado, inicialmente, um resumo de toda a dissertação. Em seguida, são relatadas as principais contribuições desenvolvidas na pesquisa e sugestões para trabalhos futuros.

## 6.1 REVISÃO DO TRABALHO

O desenvolvimento dos negócios das organizações atuais está cada vez mais vinculado aos serviços de TI. Quando tais serviços sofrem incidentes, precisam ser solucionados o mais rápido possível, diminuindo os prejuízos que possam vir a causar as organizações. ITIL propõe a segmentação de atividades e a utilização de alguns métodos para auxiliar a tomada de decisão na resolução de incidentes de TI. Porém, não é proposto nas suas melhores práticas a utilização das técnicas da inteligência artificial, RBR e RBC, para apoiar as tomadas de decisões para resolução de incidentes de TI.

Considerando a utilização e combinação das técnicas RBR e RBC para apoiar e melhorar os procedimentos das atividades definidas pelo ITIL, faz-se necessário a definição de forma ordenada de quando e em qual momento devemos utilizar cada uma delas e quais os passos presentes no processo de solucionar o incidente.

Pesquisas bibliográficas foram realizadas para definição e entendimento sobre os temas abordados nesta pesquisa: processo de tomada de decisão, resolução de incidentes de TI e técnicas de inteligência artificial capazes de apoiar tomadas de decisão. Alguns trabalhos relacionados com as temáticas envolvidas para construção da abordagem também foram pesquisados, os quais trouxeram relevantes contribuições para esta dissertação.

Este trabalho foi motivado pelas lacunas encontradas acima. Assim, foi feita a construção de uma Abordagem Metodológica de apoio à decisão para resolução de incidentes de TI. A construção ocorreu respeitando o processo de tomada de decisão proposto por Simon e com a utilização de técnicas da inteligência artificial RBR e RBC para apoiar as tomadas de decisões dos integrantes das equipes de suporte, sendo descrito, em detalhes, todos os passos necessários para solucionar o incidente. A biblioteca ITIL auxiliou no entendimento do domínio de resolução de incidentes de TI e de como estruturar as atividades.

Como resultado prático, e com o objetivo de avaliar o trabalho proposto, foi realizado um estudo de caso dentro de uma organização. Para isto, primeiramente foi feito o desenvolvimento de um protótipo que utilizou o *framework jColibri* para realizar o raciocínio

por analogia e a ferramenta *Expert Sinta* para efetuar o raciocínio por regra. Uma base de casos e bases de regras foram criadas de acordo com os incidentes coletados na organização e com a literatura sobre incidentes. Em seguida, foram apresentados casos de incidentes coletados em uma organização e a aplicação da abordagem metodológica proposta para resolvê-los, auxiliada por um protótipo construído.

Objetivando verificar a compressão e aceitação da abordagem proposta, um questionário contendo 10 questões foi aplicado aos especialistas da área.

## 6.2 PRINCIPAIS CONTRIBUIÇÕES

Com o objetivo de ser avaliado pela comunidade acadêmica a cerca das contribuições deste trabalho, um artigo foi publicado e apresentado na Conferência Ibérica de Sistemas e Tecnologias da Informação (MELLO; LOPES, 2015), descrevendo uma metodologia de apoio à decisão aliada ao raciocínio por analogia (RBC), respeitando as fases do processo decisório de Simon, aplicada à resolução de incidentes de TI.

Com o desenvolvimento desta dissertação, após a realização da revisão bibliográfica e escrita de artigo, entende-se que as principais contribuições obtidas com a criação da abordagem metodológica proposta são:

a) Representação estruturada dos casos de incidentes para armazenamento e recuperação na base de casos;
b) Apresentação de um esquema com a combinação do Raciocínio Baseado em Casos e Raciocínio Baseado em Regras para apoiar as atividades de tomada de decisão na resolução de incidentes;
c) Definição de uma abordagem metodológica para resolução de incidentes de TI, descrevendo atividades com um conjunto de passos a serem seguidos de forma ordenada pelos tomadores de decisão, respeitando o modelo decisório de Simon.

## 6.3 TRABALHOS FUTUROS

Com a conclusão deste trabalho, percebeu-se que alguns pontos ainda encontram-se abertos, os quais podem ser analisados com o intuito de ampliar o conjunto de contribuições disponibilizadas para a comunidade científica. Desta forma, algumas indicações de trabalhos futuros, são listadas a seguir:

a) **Utilizar sistemas colaborativos:** Estes tipos de sistemas são ferramentas de *software* utilizadas através de redes de computadores que facilitam a interação e a execução de trabalhos em grupos. A troca de informações entre os integrantes da equipe de suporte pode ser enriquecida utilizando sistemas colaborativos de tal forma que dúvidas sobre configurações e funcionamento dos serviços de TI possam ser facilmente compartilhados;

b) **Adicionar sistemas de recuperação da informação:** Esses sistemas realizam procedimentos para localizar, dentro da totalidade de informações disponíveis, aquelas informações realmente relevantes. Seria importante adicionar na abordagem metodológica técnicas de recuperação da informação de tal forma que recuperasse informações em fontes confiáveis na internet sobre problemas de incidentes, possibilitando o apoio nas tomadas de decisões dos integrantes da equipe de suporte, quando as informações das bases de regras e casos não sejam suficientes;

c) **Inclusão do gerenciamento de problemas**: Enquanto o objetivo do gerenciamento de incidente é restaurar a operação normal do serviço de TI o mais rápido possível, o gerenciamento de problemas, de acordo com o ITIL, cuida da resolução definitiva e da prevenção de falhas, fazendo com que haja assim, a possibilidade do desenvolvimento de uma abordagem metodológica para apoiar as decisões no gerenciamento de problemas;

d) **Construir ontologia**: Ontologias são utilizadas para facilitar o compartilhamento e a reutilização de informação, tornou-se popular com o comprometimento de fornecer uma visão compartilhada e comum de um domínio específico. Dessa forma, a construção de uma ontologia para a abordagem proposta poderia ajudar no compartilhamento e na reutilização das informações sobre resolução de incidentes fornecidas por ela.

# REFERÊNCIAS


AAMODT, Agnar; PLAZA, Enric. Case-based reasoning: Foundational issues, methodological variations, and system approaches. **AI communications**, Netherlands, v. 7, n. 1, p. 39-59, 1994.

BELLO-TOMÁS, Juan José; GONZÁLEZ-CALERO, Pedro A.; DÍAZ-AGUDO, Belén. Jcolibri: An object-oriented framework for building cbr systems. In: ADVANCES in case-based reasoning. Springer Berlin Heidelberg, p. 32-46, 2004.

CABRERA, Mariana Maceiras; EDYE, Ernesto Ocampo. Integration of rule based expert systems and case based reasoning in an acute bacterial meningitis clinical decision support system. **International Journal of Computer Science and Information Security**, EUA, v.7, n.2, p. 112-118, mar. 2010.

CARVALHO, Alexey. Raciocínio baseado em casos aplicado ao processo decisório. **Revista de Ciências Gerenciais,** Valinhos (SP), v. 12, n. 16, p. 25-36, 2008.

CHIAVENATO, Idalberto. **Administração nos Novos Tempos**. 2. ed. Rio de Janeiro: Elsevier, 2010.

ERBACHER, Robert F.; HUTCHINSON, Steve E. **Extending Case-Based Reasoning (CBR) Approaches to Semi-automated Network Alert Reporting**. Adelphi: ARMY RESEARCH LAB ADELPHI MD, 2013.

ESHACH, Haim (Ed.). **Science literacy in primary schools and pre-schools**. Netherlands: Springer Science & Business Media, 2006.

ESHETE, Azeb Bekele. **Integrated Case Based and Rule Based Reasoning for Decision Support**. Trondheim: Norwegian University of Science and Technology, 2009.

FRIEDMAN-HILL, Ernest. **Jess in Action**. Greenwich, CT: Manning, 2003.

GARCÍA, Juan A. **jCOLIBRI:** A multi-level platform for building and generating CBR systems. 2008. 91 f. Tese (Doutorado em Engenharia de Software), Universidad Complutense de Madrid, Madrid, 2008.

GENTNER, Dedre; SMITH, Linsey. Analogical reasoning. In: ENCYCLOPEDIA of human behavior. London: Elsevier, 2012. p. 130-136.

GUESSOUM, Souad et al. Combining Case and Rule Based Reasoning for the Diagnosis and Therapy of Chronic Obstructive Pulmonary Disease. **International Journal of Hybrid Information Technology**, Coreia do Sul, v. 5, n. 3, 2012. p. 145-160.

GARCIA, Juan et al. PETRIDIS, Milton (Ed.). jcolibri 1.0 in a nutshell. A software tool for designing cbr systems. In: UK WORKSHOP ON CASE BASED REASONING, 10., 2005, Greenwich. **Proccedings…** Greenwich: University of Greenwich, 2005. p. 20–28.

KOLODNER, J. **Case-Based Reasoning**. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

LACHTERMACHER, Gerson. **Pesquisa Operacional na Tomada de Decisão**. 4. ed. Rio de Janeiro: Pearson Prentice Hall, 2009.

LEE, Mal Rey. An exception handling of rule-based reasoning using case-based reasoning. **Journal of Intelligent and Robotic Systems**, Denver, v. 35, n. 3, 2002. p. 327-338

LOPES, Expedito C.; SCHIEL, Ulrich; SANTOS JR.; Gilson, P. A Decision Support Methodology for the Control of Alternative Penalties - A Case-Based Reasoning Approach**.** In: EKNOW, 1., 2009, Cancun (Mex.). INTERNATIONAL CONFERENCE ON INFORMATION, PROCESS, AND KNOWLEDGE MANAGEMENT, 2009, Washington (EUA). **Proccedings…** Washington (EUA): IEEE Computer Society, 2009. p. 72-77.

LORIGGIO, Antônio. Diagnóstico: Um modelo e seus fatores críticos de sucesso. **Caderno de Pesquisas em Administração**, São Paulo, vol. 1, n. 2, 1996. p. 1-9.

LUGER, George F. **Inteligência Artificial**. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

MAGALHÃES, Ivan Luizio; PINHEIRO, Walfrido Brito. **Gerenciamento de serviços de TI na prática:** uma prática abordagem com base na Itil. São Paulo: Novatec, 2007.

MÁNTARAS, Ramon et al. Retrieval, reuse, revision and retention in case-based reasoning. **The Knowledge Engineering Review**, Cambridge, v. 20, n. 3, 2005. p. 215-240.

MELLO, Thiago Dias Freire; LOPES, Expedito Carlos. Using case-based reasoning into a decision support methodology for the incident resolution control in IT. Iberian Conference on, 10., 2015, Portugal. **Information Systems and Technologies (CISTI)**: IEEE, 2015. p. 1-6.

MENDONÇA, Andréa. **Programação Orientada ao Problema**: **uma Metodologia para Entendimento de Problemas e Especificação no Contexto de Ensino de Programação para Iniciantes.** 2010. 187 f. Tese (Doutorado em Ciência da Computação), Universidade Federal de Campina Grande, Campina Grande, 2010.

MENKEN, Ivanka. **ITIL V3 Implementation Quick Guide-The Art of Stress-Free IT Service Management.** Disponível em: <ftp://ftp.psu.ac.th/pub/itil/The%20Art%20of%20Stress%20free%20IT%20service%20mgt%20-%20second%20edition.pdf>. Acesso em: 5 fev. 2015.

MONTANA, Patrick J.; CHARNOV, Bruce H. **Administração**. 3. ed. São Paulo: Saraiva, 2010.

OGC. **ITIL Service Operation**. UK: TSO, 2011a.

OGC. **ITIL Service Strategy**. UK: TSO, 2011b.

OLIVEIRA, Carlos Henrique J. S.; GREGHI, Renan R.; PIMENTEL, Edson P. **Ambiente para Assistência à Aprendizagem de Programação Baseado em Padrões Pedagógicos.** São Caetano do Sul, 2008. Disponível em: <http://www.proativa.virtual.ufc.br/sbie/CD_ROM_COMPLETO/workshops/workshop%202/

Ambiente%20para%20Assist%EAncia%20%E0%20Aprendizagem%20de.pdf>. Acesso em: 5 fev. 2015.

PEREIRA, Breno A. Diniz; LOBLER, Mauri Leodir; SIMONETTO, Eugênio de Oliveira. Análise dos Modelos de Tomada de Decisão sob o Enfoque Cognitivo. **Revista de Administração da UFSM**, Santa Maria (RS), v. 3, n. 2, p. 260-268, 2010.

PIANEZZOLA, Tatiano et al. Representação de Conhecimentos Centrada em Modelos para o Uso em Ambientes Complexos de Aprendizagem. **RENOTE,** Porto Alegre, v. 4, n. 2, p. 1-11, 2006.

PINHEIRO, Flávio R. **Fundamentos no gerenciamento de serviços de TI**. São Paulo: TIEXAMES, 2014. Disponível em: <http://www.tiexames.com.br/ensino/home.php>. Acesso em: 5 fev. 2015.

PRENTZAS, Jim; HATZILYGEROUDIS, Ioannis. Categorizing approaches combining rule based and case-based reasoning. **Expert Systems**, v. 24, n. 2, p. 97-122, 2007.

RICHTER, Michael M.; AAMODT, Agnar. Case-based reasoning foundations.The **Knowledge Engineering Review**, Cambridge, v. 20, n. 03, p. 203-207, 2005.

RICHTER, Michael M.; WEBER, Rosina O. **Case-Based Reasoning**. A Textbook, Springer, 2013.

SCHANK, Roger C.; ABELSON, Robert P.; SCRIPTS, Plans. **Goals and understanding**. Erlbanum: Eksevier Science, 1977

SHIU, Simon; PAL, Sankar K. **Foundations of Soft Case-Based Reasoning**. New Jersey: John Wiley & Sons, 2004.

SILVA, Renato Afonso Cota. Inteligência artificial aplicada a ambientes de engenharia de software: Uma visão geral. **INFOCOMP Journal of Computer Science**, Viçosa, v. 4, n. 4, p. 27-37, 2005.

SIMON, Herbert A. **The new science of management decision**. New York: Harper & Row, 1960.

TEHRANI, Abtin Refahi Farjadi; MOHAMED, Faras Zuheir Mustafa. A CBR-based Approach to ITIL-based Service Desk. **Journal of Emerging Trends in Computing and Information Sciences**, Québec, v. 2, n. 10, p. 476-484, 2011.

TURBAN, Efraim; VOLONINO, Linda. **Tecnologia da informação para gestão:** em busca do melhor desempenho estratégico e operacional. Porto Alegre: Bookman, 2013.

SPIRANDELLI, Lennon P. et al. Sistemas especialistas: um estudo de caso com o expert sinta. **Revista Eletrônica de Sistemas de Informação e de Gestão Tecnológica**, Franca, v. 1, n. 1, p. 1-16, 2011.

WANGENHEIM, Christiane Gress Von; WANGENHEIM, Aldo Von. **Raciocínio Baseados em Casos**. Barueri, SP: Manole, 2003.

ZIMMERMANN, Antonio Carlos. et al. Uma proposta de utilização de raciocínio baseado em casos–rbc em reconhecimento faces humanas tri-dimensionais. In: CONGRESSO BRASILEIRO DE COMPUTAÇÃO, 2., 2002, Itajaí (SC). **Anais…** 2002. Disponível em: <http://www.inf.ufsc.br/~l3c/artigos/Zimmermann02a.pdf>. Acesso em: 5 fev. 2015.

WATSON, Ian; MARIR, Farhi. Case-based reasoning: A review. **The knowledge engineering review,** Cambridge v. 9, n. 4, p. 327-354, 1994.

## APÊNDICE A – Questionário Aplicado

Este apêndice destina-se ao questionário aplicado neste trabalho, podendo ser necessário para o entendimento do mesmo.

# *Questionário sobre a Abordagem Metodológica de Apoio à Decisão para resolução de incidentes de TI centrada na combinação RBC/RBR.*

*O objetivo deste questionário é avaliar a compreensão e aceitabilidade da Abordagem Metodológica proposta para a resolução de incidentes em TI.*

### Dados do participante

1. **Tipo de organização que atua**

   ( ) Pública ( ) Privada

2. **Cargo atual**:

3. **Em qual nível do suporte você atua?**

   ( ) 1º nível ( ) 2º nível

4. **Tempo de experiência na área de suporte (em anos):**

### Conhecimentos gerais

5. **Ao se deparar com um problema de incidente, quais mecanismos normalmente utiliza para resolvê-lo (possíveis múltiplas respostas)?**

   ( ) Busca soluções de problemas ocorridos anteriormente na sua organização
   ( ) Busca apoio do especialista com experiência prática na sua organização
   ( ) Procura solução fora da organização
   ( ) Outra resposta. Descreva-a_____

6. **Numere de 0 a 5, qual seu nível de entendimento, teórico ou prático, acerca de**

( ) Raciocínio Baseado em Casos (RBC)

( ) Raciocínio Baseado em Regras (RBR)

( ) Combinação de RBC/RBR

7. **Com base no respondido na questão anterior, você considera que estes raciocínios, isoladamente ou combinados, podem auxiliar na resolução de incidentes de TI em sua organização?**

( ) Sim. Por quê? ( ) Depende. Por quê? ( ) Não. Por quê?

**Questões específicas sobre a abordagem metrológica proposta**

8. **Sobre a realização das atividades para resolução de incidentes em TI propostas, preencha o quadro abaixo com os níveis padronizados a seguir:**

| **Atividade** | **Nível** | **Passos não compreendida nesta atividade /comentários/sugestões** |
|---|---|---|
| Diagnóstico inicial | | |
| Investigação e diagnóstico | | |
| Resolução e recuperação | | |

| | | |
|---|---|---|
| | | |

9. **Em linhas gerais, considera a abordagem apropriada? Caso negativo, o que poderia ser melhorado?**

10. **Considera que a abordagem proposta pode ser aplicada dentro da organização em que trabalha para resolver incidentes de TI?**

( ) Sim, na íntegra.

( ) Sim, parcialmente. Neste caso, qual(is) atividades utilizaria?

( ) Não. Porquê?

# APÊNDICE B – Código JCOLIBRI

Este apêndice destina-se ao código criado no Jcolibri para o desenvolvimento do módulo do raciocínio por analogia do protótipo.

## B.1 Arquivos para situação-problema

A seguir serão descritos as principais partes dos arquivos para recuperação dos casos quando inserida a situação-problema.

**Parte do código CBRAplication.java para situação-problema**

```
/** Configures the similarity */
	@Generated(value = { "CS-Similarity" })
	private NNConfig getSimilarityConfig() {
		NNConfig simConfig = new NNConfig();
		simConfig
				.setDescriptionSimFunction(new
jcolibri.method.retrieve.NNretrieval.similarity.global.Average());
		Attribute attribute0 = new Attribute("sitProblema",
CasoDescricao.class);
		simConfig
				.addMapping(
						attribute0,
						new
jcolibri.method.retrieve.NNretrieval.similarity.local.MaxString());
		simConfig.setWeight(attribute0, 1.00);
		return simConfig;
	}

	//*******************************************************************/
	// Methods
	//*******************************************************************/


	@Generated(value = { "ColibriStudio" })
	@Override
	public CBRCaseBase preCycle() throws ExecutionException {
		casebase.init(connector);
		return casebase;
	}

	@Generated(value = { "ColibriStudio" })
	@Override
	public void cycle(CBRQuery query) throws ExecutionException {
		NNConfig simConfig = new NNConfig();
		simConfig.setDescriptionSimFunction(new
jcolibri.method.retrieve.NNretrieval.similarity.global.Average());
		Attribute attribute0 = new
Attribute("sitProblema",CasoDescricao.class);
		simConfig.addMapping(attribute0,new
jcolibri.method.retrieve.NNretrieval.similarity.local.MaxString());
		simConfig.setWeight(attribute0, 1.00);
```

```
            dados = NNScoringMethod.evaluateSimilarity(casebase.getCases(),
query, simConfig); // Calculando as similaridades dos casos da base em relação ao
caso consultado
        Collection<CBRCase> selectedcases = SelectCases.selectTopK(dados, 4); //
Selecionando os 4(quatro) casos mais similares ao consultado
        //DisplayCasesMethod.displayCasesWithEditOption(selectedcases); //
Exibindo em uma interface gráfica padrão do framework os casos selecionados
        DisplayCasesTableMethod.displayCasesInTableBasic(selectedcases);
      }

      @Generated(value = { "ColibriStudio" })
      @Override
      public void postCycle() throws ExecutionException {
            connector.close();
      }
```

**Parte do código CasoDescricao.java (atributos do caso) para situação-problema**

```
private java.lang.Integer idDesc;


      public java.lang.Integer getIdDesc()
            {
                  return idDesc;
            }
      public void setIdDesc(java.lang.Integer idDesc0)
            {
                  this.idDesc = idDesc0;
            }

      private java.lang.String sitProblema;


      public java.lang.String getSitProblema()
            {
                  return sitProblema;
            }
      public void setSitProblema(java.lang.String sitProblema1)
            {
                  this.sitProblema = sitProblema1;
            }


      @Override
      public Attribute getIdAttribute()
            {
                  return new Attribute("idDesc",this.getClass());
            }

      public String toString()        {
                  return "["+ sitProblema + " , " + idDesc +"]";
            }

}
```

**Parte do código CasoSolucao.java para situação-problema**

```
Public class CasoSolucao implements CaseComponent {

	/* Generated Class. Please Do Not Modify... */
	private java.lang.Integer idSolucao;
	public java.lang.Integer getIdSolucao()
		{
			return idSolucao;
		}
	public void setIdSolucao(java.lang.Integer idSolucao2)
		{
			this.idSolucao = idSolucao2;
		}
	private java.lang.String itemConfig;
	public java.lang.String getItemConfig()
		{
			return itemConfig;
		}
	public void setItemConfig(java.lang.String itemConfig3)
		{
			this.itemConfig = itemConfig3;
		}
	private java.lang.String descProblema;
	public java.lang.String getDescProblema()
		{
			return descProblema;
		}
	public void setDescProblema(java.lang.String descProblema4)
		{
			this.descProblema = descProblema4;
		}

	@Override
	public Attribute getIdAttribute()
		{
			return new Attribute("idSolucao",this.getClass());
		}

	public String toString()	{
			return "["+ idSolucao + " , " + descProblema + " , " +
itemConfig +"]";
		}

}
```

## B.2 Arquivos para problema

A seguir serão descritos as principais partes dos arquivos para recuperação dos casos quando inserido o problema.

**Parte do código CBRAplication.java para problema**

```
/** Configures the similarity */
	@Generated(value = { "CS-Similarity" })
```

```
	@Override
	public CBRCaseBase preCycle() throws ExecutionException {
		casebase.init(connector);
		return casebase;
	}

	@Generated(value = { "ColibriStudio" })
	@Override
	public void cycle(CBRQuery query) throws ExecutionException {
		//System.out.println("Query" + query);
		NNConfig simConfig = new NNConfig();
		simConfig.setDescriptionSimFunction(new
jcolibri.method.retrieve.NNretrieval.similarity.global.Average());
		Attribute attribute0 = new Attribute("DescricaoProblema",
Descricao.class);
		//System.out.println("atributo: " + attribute0);
		simConfig.addMapping(attribute0,new
jcolibri.method.retrieve.NNretrieval.similarity.local.MaxString());
		simConfig.setWeight(attribute0, 1.00);

		//System.out.println("SIMCONFIG: " + simConfig);

		dados = NNScoringMethod.evaluateSimilarity(casebase.getCases(),
query, simConfig); // Calculando as similaridades dos casos da base em relação ao
caso consultado

		String dados0 = dados.toArray()[0].toString();
		String Ddados0 = dados0.substring(dados0.indexOf(">")+2);
		System.out.println("CASE 0: " + Ddados0);

		String dados1 = dados.toArray()[1].toString();
		String Ddados1 = dados1.substring(dados1.indexOf(">")+2);
		System.out.println("CASE 1: " + Ddados1);

		String dados2 = dados.toArray()[2].toString();
		String Ddados2 = dados2.substring(dados2.indexOf(">")+2);
		System.out.println("CASE 2: " + Ddados2);

		String dados3 = dados.toArray()[3].toString();
		String Ddados3 = dados3.substring(dados3.indexOf(">")+2);
		System.out.println("CASE 3: " + Ddados3);

		Collection<CBRCase> selectedcases = SelectCases.selectTopK(dados, 4);
// Selecionando os 3(três) casos mais similares ao consultado
		//System.out.println("CASO SELECIONADO: " + selectedcases);

	//DisplayCasesMethod.displayCasesWithEditOption(selectedcases); //
Exibindo em uma interface gráfica padrão do framework os casos selecionados
	DisplayCasesTableMethod.displayCasesInTableBasic(selectedcases);
	}

	@Generated(value = { "ColibriStudio" })
	@Override
	public void postCycle() throws ExecutionException {
		connector.close();
	}
* Main method
	*/
```

```
        @Generated(value = { "ColibriStudio" }) // interface para exibição
        public static void main(String[] args) {
                final CBRApplication cbrApp = new CBRApplication();
                try {
                        cbrApp.configure();
                        cbrApp.preCycle();

                        final CBRQuery query = new CBRQuery();
                        final Descricao novaDesc = new Descricao();

                        final JDialog dialog = new JDialog();
                        dialog.setModal(true);
                  JPanel p = new JPanel(new SpringLayout());

                  JLabel labelDI = new JLabel("Descrição do Problema:",
JLabel.TRAILING);
                  p.add(labelDI);
                  final JTextField txtDI = new JTextField();
                  p.add(txtDI);

                  // organizando a exibição
                  SpringUtilities.makeCompactGrid(p,
                              1, 2, //rows, cols
                       6, 6,        //initX, initY
                       6, 6);       //xPad, yPad


                        dialog.getContentPane().setLayout(new BorderLayout());
                        dialog.getContentPane().add(p,BorderLayout.CENTER);

                        JPanel okPanel = new JPanel();
                        JButton okButton = new JButton("Ok");
                        okButton.addActionListener(new ActionListener(){
                            public void actionPerformed(ActionEvent arg0)
                            {
                              dialog.setVisible(false);
                              novaDesc.setDescricaoProblema((txtDI.getText()));
                              query.setDescription(novaDesc); // enviar o desc. digita
para consulta
                              try {
                                          cbrApp.cycle(query);
                                    } catch (ExecutionException e) {
                                          // TODO Auto-generated catch block
                                          e.printStackTrace();
                                    }
                                    try {
                                          cbrApp.postCycle();
                                    } catch (ExecutionException e) {
                                          // TODO Auto-generated catch block
                                          e.printStackTrace();
                                    }
                                    System.exit(0);
                            }
                        });
                        okPanel.add(okButton);
                        dialog.getContentPane().add(okPanel,BorderLayout.SOUTH);

                        dialog.pack();
                        dialog.setSize(400, 110);
```

```
                WindowUtils.centerWindow(dialog);
                dialog.setTitle("Atendimento de Incidente");
                dialog.setVisible(true);

                //System.exit(0);
                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);

            } catch (ExecutionException e) {
                e.printStackTrace();
            }

        }

}
```

**Parte do código CasoDescricao.java para problema**

```
public class Descricao implements CaseComponent {
        /* Generated Class. Please Do Not Modify... */
        private java.lang.Integer idDesc;

        public java.lang.Integer getIdDesc()
            {
                return idDesc;
            }
        public void setIdDesc(java.lang.Integer idDesc0)
            {
                this.idDesc = idDesc0;
            }
        private java.lang.String DescricaoProblema;

        public java.lang.String getDescricaoProblema()
            {
                return DescricaoProblema;
            }
        public void setDescricaoProblema(java.lang.String DescricaoProblema1)
            {
                this.DescricaoProblema = DescricaoProblema1;
            }

        @Override
        public Attribute getIdAttribute()
            {
                return new Attribute("idDesc",this.getClass());
            }

        public String toString()        {
                return "["+ DescricaoProblema + " , " + idDesc +"]";
            }

}
```

**Parte do código CasoSolucao.java para problema**

```
public class CasoSolucao implements CaseComponent {
        /* Generated Class. Please Do Not Modify... */
```

```
        private java.lang.Integer idSol;

        public java.lang.Integer getIdSol()
                {
                        return idSol;
                }
        public void setIdSol(java.lang.Integer idSol2)
                {
                        this.idSol = idSol2;
                }
        private java.lang.String Solucao;


        public java.lang.String getSolucao()
                {
                        return Solucao;
                }
        public void setSolucao(java.lang.String Solucao3)
                {
                        this.Solucao = Solucao3;
                }

        private java.lang.String Passos;


        public java.lang.String getPassos()
                {
                        return Passos;
                }
        public void setPassos(java.lang.String Passos4)
                {
                        this.Passos = Passos4;
                }

        private java.lang.String Resultado;


        public java.lang.String getResultado()
                {
                        return Resultado;
                }
        public void setResultado(java.lang.String Resultado5)
                {
                        this.Resultado = Resultado5;
                }


        @Override
        public Attribute getIdAttribute()
                {
                        return new Attribute("idSol",this.getClass());
                }

        public String toString()        {
                        return "["+ Solucao + " , " + Passos + " , " + idSol + " , " +
Resultado +"]";
                }

}
```

# APÊNDICE C – Regras do Expertsinta

Este apêndice destina-se a exibição da construção das principais regras utilizadas para o desenvolvimento do módulo do raciocínio por regra do protótipo.

## C.1 Regras para sem acesso aos sistemas

**Construção das regras para sem acesso aos sistemas.**

```
--------------------------------------------------------------------------------

VARIÁVEIS

  Cabo
    Valores:
      Sim
      Não
    Tipo:
      multivalorada
  Diagnostico Inicial
    Valores:
      Configurar IP automatico nas propiedades do TCP/IP
      Reiniciar Gateway e Entrar em contato com a Operadora do LINK
      Conecte o cabo de rede
      Configurar DB2 cliente para srvdb2 para porta 50000
      Sem atribuição de DHCP
    Tipo:
      multivalorada
  EndIP
    Valores:
      Dentro da faixa 190.x.x.x\16
      127.0.0.1
      169.254.x.x
    Tipo:
      multivalorada
  IPFixo
    Valores:
      Não
      Sim
    Tipo:
      multivalorada
  LinkDados
    Valores:
      Sim
      Não
    Tipo:
      multivalorada
  PC
    Valores:
      Sim
      Não
    Tipo:
      multivalorada

OBJETIVOS
```

```
  Diagnostico Inicial

REGRAS

  Regra 1
    SE Cabo = Não
    ENTÃO Diagnostico Inicial = Conecte o cabo de rede CNF 100%

  Regra 2
    SE IPFixo = Sim
    ENTÃO Diagnostico Inicial = Configurar IP automatico nas propiedades do TCP/IP
CNF 100%

  Regra 3
    SE EndIP = 169.254.x.x
    OU EndIP = 127.0.0.1
    ENTÃO Diagnostico Inicial = Sem atribuição de DHCP CNF 100%

  Regra 4
    SE LinkDados = Não
    ENTÃO Diagnostico Inicial = Reiniciar Gateway e Entrar em contato com a
Operadora do LINK CNF 100%

  Regra 5
    SE PC = Sim
    ENTÃO Diagnostico Inicial = Configurar DB2 cliente para srvdb2 para porta
50000 CNF 100%

PERGUNTAS

  Variável:Cabo
     Pergunta:"O cabo de rede esta conectado ao computador?"
     Motivo:"Fundamental o Cabo conectado para serviços de TI
  Variável:EndIP
     Pergunta:"Como aparece o endereço IP?"
  Variável:IPFixo
     Pergunta:"As configurações TCP/IP do computador cliente encontram-se
configuradas de forma manual?"
  Variável:LinkDados
     Pergunta:"Ao executar um ping para o Gateway da unidade a resposta é
positiva?"
  Variável:PC
     Pergunta:"Ao clicar para abrir o sistema aparece a MSG: "Parâmetro de
Configuração Inválido""

--------------------------------------------------------------------------------
Gerado com o Expert SINTA versão 1.1b
(c) 1997 - Universidade Federal do Ceará
           Laboratório de Inteligência Artificial
           LIA/UFC
--------------------------------------------------------------------------------
```

## C.2 Regras para bloqueio de porta de rede web

### Construção das regras para bloqueio de porta de rede web

```
--------------------------------------------------------------------------------

VARIÁVEIS

  gc
    Valores:
      Apache
      IIS
      JBOSS
    Tipo:
      multivalorada
  Problema
    Valores:
      Desbloqueio no Firewall IPTABLES da porta 8080
      Desbloqueio no Firewall do Windows da porta 8080
      Desbloqueio no Firewall IPTABLES da porta 8192
      Desbloqueio no Firewall do Windows da porta 80
    Tipo:
      multivalorada
  so
    Valores:
      Wndows
      Linux
    Tipo:
      multivalorada

OBJETIVOS

  Problema

REGRAS

  Regra 1
    SE so = Linux
    E  gc = Apache
    ENTÃO Problema = Desbloqueio no Firewall IPTABLES da porta 8080 CNF 100%

  Regra 2
    SE so = Wndows
    E  gc = IIS
    ENTÃO Problema = Desbloqueio no Firewall do Windows da porta 80 CNF 100%

  Regra 3
    SE so = Wndows
    E  gc = Apache
    ENTÃO Problema = Desbloqueio no Firewall do Windows da porta 8080 CNF 100%

  Regra 4
    SE so = Linux
    E  gc = JBOSS
    ENTÃO Problema = Desbloqueio no Firewall IPTABLES da porta 8192 CNF 100%

PERGUNTAS
```

```
  Variável:gc
     Pergunta:"Qual o gerenciador de conteúdo web?"
  Variável:so
     Pergunta:"O gerenciador de conteúdo web encontra-se instalado em um Sistema
operacional Windows ou Linux?"

--------------------------------------------------------------------------------
Gerado com o Expert SINTA versão 1.1b
(c) 1997 - Universidade Federal do Ceará
           Laboratório de Inteligência Artificial
           LIA/UFC
--------------------------------------------------------------------------------
```