

UNIVERSIDADE DE SANTA CRUZ DO SUL  
DEPARTAMENTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Ivan Carlos Hennig

**CBRAIN - FERRAMENTA DE RACIOCÍNIO BASEADO EM CASOS**

Santa Cruz do Sul  
2015

Ivan Carlos Hennig

**CBRAIN - FERRAMENTA DE RACIOCÍNIO BASEADO EM CASOS**

Trabalho de Conclusão II apresentado ao Curso de Ciência da Computação da Universidade de Santa Cruz do Sul para obtenção parcial do título de Bacharel em Ciência da Computação.

Orientadora: Prof<sup>ª</sup>. Dra. Rejane Frozza

Santa Cruz do Sul  
2015

## RESUMO

Este trabalho apresenta o desenvolvimento, características e funcionalidades da CBRAIN - ferramenta para projetar e desenvolver sistemas de Raciocínio Baseado em Casos (RBC). Após estudo das principais ferramentas existentes, observou-se uma carência em termos de usabilidade e o interesse por pesquisadores e profissionais nesta técnica para modelagem de situações reais. A ferramenta foi desenvolvida respeitando conceitos de usabilidade em sua interface gráfica para o usuário, onde é possível projetar e testar os sistemas RBC determinando atributos, índices e regras de similaridade para os casos. Um serviço de processamento foi criado como uma aplicação *web*, onde as requisições para executar as etapas do ciclo RBC são processadas, como a recuperação de casos, aplicando funções de similaridade local e global, adaptação e reuso. O processo de avaliação da ferramenta realizado com dois participantes mostrou que a proposta da ferramenta é promissora de ser utilizada, mesmo necessitando de ajustes e melhorias.

**Palavras chave:** Raciocínio Baseado em Casos, Técnicas de Similaridade, Ferramenta CBRAIN.

## **ABSTRACT**

This paper presents the design, features and functionality of CBRAIN – tool for projecting and developing systems using Case Based Reasoning (CBR). After studying mainly existing tools, it was noticed a shortage in terms of usability and a growing interest by researchers and professionals in this technique for modeling real world situations. The tool was developed respecting usability concepts in its graphical user interface where is possible to design and test the systems determining CBR attributes, indexes and rules of similarity for cases. A processing service was created as a web application, which requests to perform steps of the CBR cycle are processed. Like recovering cases, applying local and global similarity functions and adaptation and reuse.

**Keywords:** Case Based Reasoning, Similarity Techniques, CBRAIN Tool.

## LISTA DE FIGURAS

Figura 1: Ciclo do Raciocínio Baseado em Casos .....	13
Figura 2: Representação de um Caso .....	14
Figura 3: Tabela Comparativa .....	19
Figura 4: Interface FreeCBR.....	21
Figura 5: Interface myCBR.....	22
Figura 6: Interface CBR Shell.....	23
Figura 7: Interface jCOLIBRI – Representação .....	24
Figura 8: Interface jCOLIBRI – Similaridade .....	24
Figura 9: Fluxo Criação de Projeto na Interface Gráfica.....	28
Figura 10: Fluxo do Funcionamento Serviço de Processamento.....	29
Figura 11: Tela Símbolos Ordenados .....	31
Figura 12: Tela Principal .....	31
Figura 13: Tela Configuração .....	32
Figura 14: Tela Criação Projeto.....	33
Figura 15: Tela Criação Caso .....	34
Figura 16: Tela Criação Atributo.....	35
Figura 17: Formulário de Entrada.....	36
Figura 18: Mensagem Validação Formulário de Entrada .....	36
Figura 19: Listagem de Casos Recuperados .....	37
Figura 20: Formulário Adaptação e Reúso .....	38
Figura 21: Aparência da Ferramenta no Gnome.....	39
Figura 22: Aparência da Ferramenta no KDE .....	40
Figura 23: Tela Poedit.....	41

## **LISTA DE TABELAS**

Tabela 1: Representação Atributo-Valor .....	15
Tabela 2: Exemplo Similaridade.....	17
Tabela 3: Comparativo de Recursos da Interface do Usuário.....	25
Tabela 4: Comparativo de Recursos Técnicos.....	25

## LISTA DE EQUAÇÕES

Equação 1: Vizinho mais próximo.....	17
Equação 2: Distância Euclideana.....	17

## SUMÁRIO

RESUMO.....	3
ABSTRACT.....	4
LISTA DE FIGURAS .....	5
LISTA DE TABELAS .....	6
LISTA DE EQUAÇÕES .....	7
1. INTRODUÇÃO .....	10
2. FUNDAMENTAÇÃO TEÓRICA .....	12
2.1. Técnica de Raciocínio Baseado em Casos .....	12
2.2. Ciclo de RBC .....	12
2.3. Representação de Casos.....	13
2.4. Representação Atributo-Valor .....	15
2.5. Indexação .....	15
2.6. Similaridade .....	16
2.6.1. Similaridade Global.....	16
2.6.2. Similaridade Local .....	17
3. FERRAMENTAS.....	21
3.1. CBR-Works ( <a href="http://www.inf.ufsc.br/~patrec/cbr.html">http://www.inf.ufsc.br/~patrec/cbr.html</a> ) .....	21
3.2. FreeCBR ( <a href="http://freecbr.sourceforge.net">http://freecbr.sourceforge.net</a> ) .....	21
3.3. myCBR ( <a href="http://mycbr-project.net">http://mycbr-project.net</a> ).....	21
3.4. CBR Shell ( <a href="http://www.aiai.ed.ac.uk/project/cbr">http://www.aiai.ed.ac.uk/project/cbr</a> ).....	22
3.5. jCOLIBRI ( <a href="http://gaia.fdi.ucm.es/research/colibri/jcolibri">http://gaia.fdi.ucm.es/research/colibri/jcolibri</a> ).....	23
3.6. Análise Comparativa das Ferramentas Estudadas .....	24
3.7. Considerações .....	26
4. FERRAMENTA PARA RACIOCÍNIO BASEADO EM CASOS.....	27
4.1. Introdução.....	27
4.2. Serviço de Processamento .....	29
4.2.1. Implementação das Funções de Similaridade .....	30
4.3. Interface da Ferramenta .....	31
4.4. Configurações .....	32
4.5. Criação Projeto.....	32
4.6. Criação dos Casos.....	33
4.7. Criação de Atributos .....	34
4.8. Formulário de Entrada .....	35



4.9.	Adaptação e Reúso.....	36
4.10.	Usabilidade .....	38
4.10.1.	Disposição dos Botões.....	39
4.10.2.	Fontes e Cores.....	39
4.11.	Ambiente de Desenvolvimento .....	40
4.11.1.	Internacionalização .....	40
4.12.	Processo de Validação .....	41
5.	CONCLUSÃO E TRABALHOS FUTUROS .....	43
6.	REFERÊNCIAS.....	44
	ANEXO A - Avaliação do Sistema CBRAIN.....	46
	ANEXO B - Resposta Avaliação Ederson.....	48
	ANEXO C - Resposta Avaliação Katieli.....	50

## 1. INTRODUÇÃO

Recentemente cresce o interesse em raciocínio baseado em casos (RBC), bem como o número de artigos relacionados ao assunto. Como exemplo cita-se o trabalho desenvolvido em (KIPPER *et al.*, 2014), que apresenta o processo de desenvolvimento de um sistema de gestão do conhecimento, aplicado no setor comercial de uma empresa, que utiliza a técnica de RBC para apoio à tomada de decisão.

De acordo com Kipper *et al.*, (2014), foi possível concluir que as técnicas e fundamentos de RBC podem facilitar e auxiliar nos processos de tomada de decisão, apresentando-se como caminho ao sucesso e efetivando o uso de sistema de gestão do conhecimento organizacional.

Outro trabalho com o uso de RBC é o de gerenciamento de processos de negócio, desenvolvido por (SANTOS *et al.*, 2013). O objetivo foi identificar quando uma determinada execução de processo tende a não render os resultados esperados, com base em execuções passadas similares a este processo, que estão armazenadas na base de casos.

RBC é um paradigma para resolução de problemas que difere das abordagens conhecidas da inteligência artificial, pois não se utiliza de bases de conhecimento genéricas. Ele usa conhecimento específico de problemas vivenciados e extrai soluções concretas destes casos fazendo adaptações, se necessário. Em outras palavras, um problema pode ser resolvido buscando, revisando e ajustando problemas similares ao do passado (AMODT; PLAZA, 1994).

O objetivo principal desse trabalho é desenvolver uma ferramenta computacional para permitir o planejamento, a criação e testes de aplicações de RBC.

Para alcançar o objetivo principal, alguns objetivos específicos tiveram que ser atingidos, tais como:

- Estudar ferramentas de RBC existentes, com o propósito de levantar características de interface e funcionamento.
- Criar uma estrutura para armazenamento das bases de casos, de forma dinâmica, ou seja, sem a necessidade de criação e manutenção dos campos, como é comum em um banco de dados relacional.
- Criar mecanismos para organizar a base de conhecimento, facilitando a criação e utilização de índices que levam em conta as similaridades armazenadas, que são necessários para tornar o processo de recuperação dos casos rápido e eficiente.
- Desenvolver uma interface com o usuário, que permita:

- Criar entradas de dados customizadas, podendo ser texto puro, *boolean*, múltiplas opções, série de valores, para receber novos casos.
- Criar as saídas dos dados em forma de consulta, ordenado e iniciando pelo caso mais similar. Esta deverá apresentar a pontuação dos casos recuperados e um rastreador que explica como cada similaridade foi calculada.
- Criar uma área para edição de pesos e conexões entre valores, que deverá refletir nos índices e cálculos de similaridades.
- Desenvolver uma interface *Web services*, que deverá fornecer acesso aos métodos de inclusão de novos casos e recuperação de casos similares.

A pesquisa em busca de ferramentas RBC, que será vista no capítulo 3, indica que há uma grande carência de softwares modernos, intuitivos e que respeitem normas de usabilidade, e também que possuam uma interface fácil de integração com outras aplicações. Esta tarefa precisa ser fácil e produtiva, para qualquer empresa ou profissional que deseja incluir este ramo da inteligência artificial em suas soluções.

Como justificativa científica para o desenvolvimento deste trabalho, cita-se a possibilidade de oferecer uma ferramenta visual prática e de fácil utilização, que respeita regras de usabilidade, bem como uma interface para interconectividade, para grupos de pesquisadores e profissionais da área que desejam desenvolver suas aplicações com esta técnica.

O texto está organizado em capítulos, sendo o capítulo 2 uma apresentação da fundamentação teórica, ressaltando aspectos de RBC. O capítulo 3 apresenta um estudo das ferramentas existentes de RBC. O capítulo 4 descreve a ferramenta desenvolvida em todos os seus aspectos e o processo de validação realizado. O capítulo 5 mostra a conclusão e a indicação de trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os principais conceitos de RBC, descreve os elementos que compõem a sua estrutura e como cada etapa está relacionada para atingir o resultado esperado, que é a obtenção de soluções para problemas já solucionados anteriormente.

### 2.1. Técnica de Raciocínio Baseado em Casos

RBC é um tema abrangente e possui técnicas variadas para cada etapa do seu ciclo de funcionamento.

Um caso é uma parte de conhecimento representando uma experiência que ensina uma lição fundamental para atingir os objetivos do motor ou processador do raciocínio (KOLODNER, 1993).

A seguir, apresenta-se as vantagens de usar RBC, conforme (KOLODNER, 1993, p.25-27):

- Permite propor soluções para problemas rapidamente.
- Permite propor soluções para problemas em áreas que não são totalmente compreendidas.
- Fornece uma forma de solucionar problemas, que nem um método algorítmico poderia.
- Os casos são úteis para interpretar problemas que não foram definidos de forma adequada.
- Relembrar experiências prévias auxilia no aprendizado e evita a repetição de erros.
- Os casos ressaltam aspectos importantes do problema, permitindo ao tomador de decisão raciocinar sobre estes aspectos.

### 2.2. Ciclo de RBC

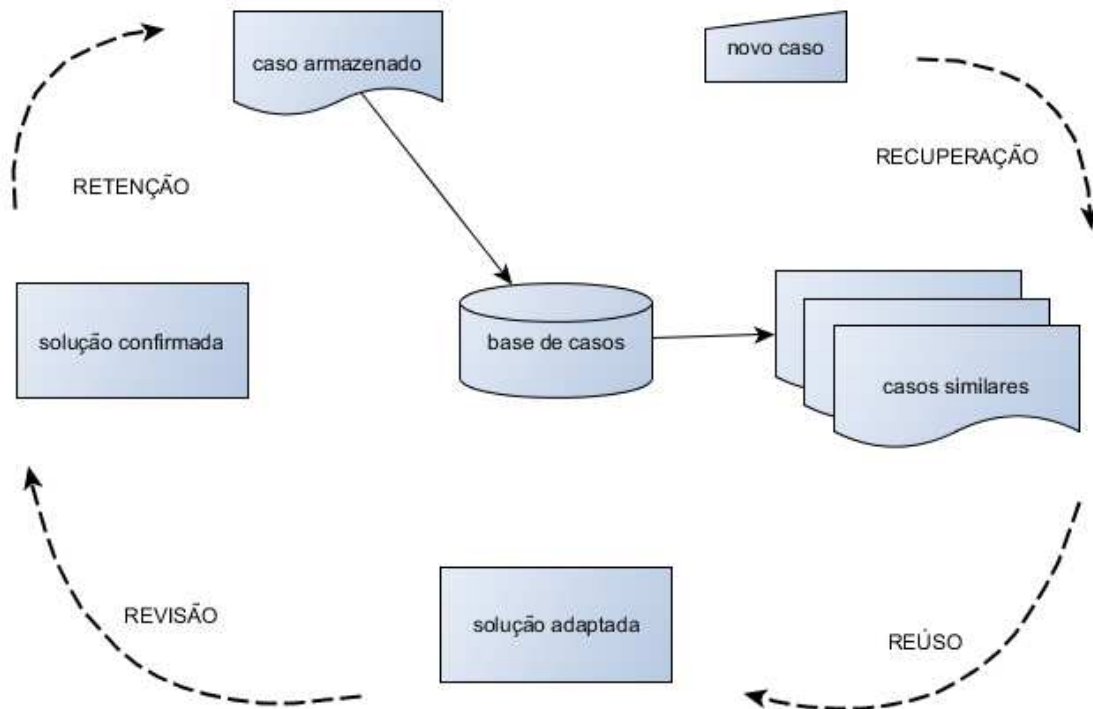
A Figura 1 ilustra o modelo mais aceito para o processo RBC, que engloba um ciclo de raciocínio contínuo composto por quatro tarefas principais (WANGENHEIM; WANGENHEIM, 2003). São elas:

- **Recuperar** os casos mais similares da base de casos.
- **Reutilizar** os casos para resolver o problema.
- **Revisar** a solução proposta.
- **Reter** a experiência ou partes desta representando o caso atual para reutilização futura.

Quando um novo caso ou problema é apresentado, dá-se início à etapa de recuperação do ciclo de RBC. Nesta fase do processo, são usadas técnicas de indexação e similaridade, que serão abordadas nas seções 2.5 e 2.6, para buscar um conjunto de casos similares.

Na etapa de reuso, os casos similares são analisados para eleger a melhor solução para um dado problema. Neste momento, é possível realizar adaptações ou incluir novos atributos recuperados para atender plenamente o propósito do RBC.

Durante a revisão, o analista decide por reter ou não um caso, levando em consideração a confirmação de que a solução resolveu satisfatoriamente o problema proposto.



**Figura 1: Ciclo do Raciocínio Baseado em Casos**

**Fonte: (WANGENHEIM; WANGENHEIM, 2003)**

Na próxima seção, 2.3, serão listadas as técnicas para representação lógica do problema e solução.

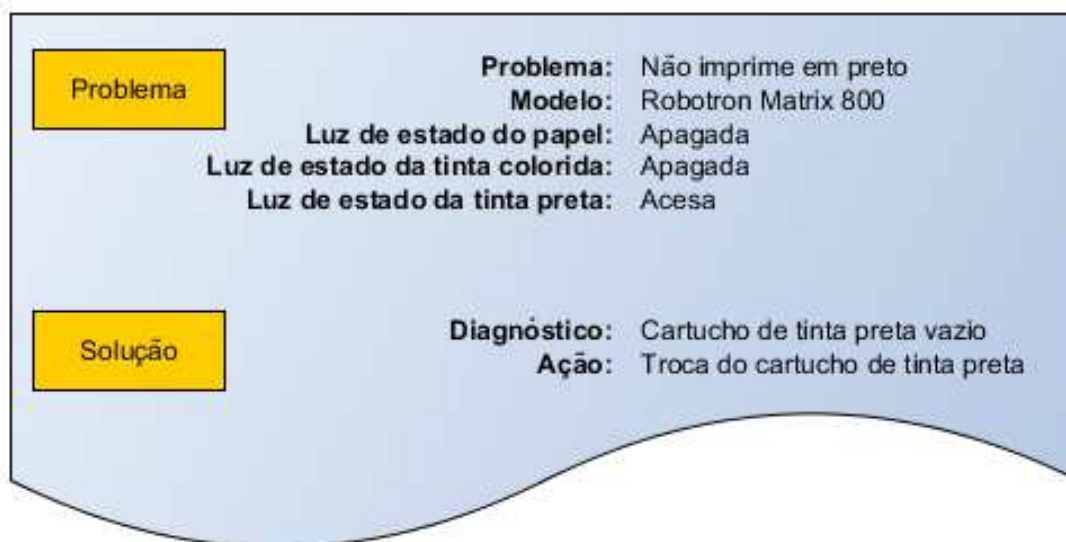
### 2.3. Representação de Casos

Embora o ciclo de RBC seja dividido em quatro tarefas, os casos de RBC são representados por dois componentes principais **Problema** e **Solução**. Na Figura 2 pode-se visualizar um modelo de representação de caso de diagnóstico.

A **descrição do problema** ou da situação codifica o estado do mundo no momento em que o raciocínio se inicia. Pode representar um problema que necessita ser resolvido ou uma situação que necessita ser interpretada, classificada ou compreendida. Um problema pode incluir objetivos, restrições e atributos que podem estar interconectados de forma a influenciar na solução do caso (WANGENHEIM; WANGENHEIM, 2003 p.67).

A **descrição da solução** pode ser representada de diversas formas, seja explicitamente, descrevendo a solução, seja por meio de um ponteiro para um documento textual com instruções ou outro conjunto de informações relevantes (WANGENHEIM; WANGENHEIM, 2003 p.68).

Qualquer representação necessita de uma linguagem definida que permita ao sistema RBC realizar o processamento de forma fácil e eficiente. Mas isto não impede que textos, fórmulas matemáticas, imagens e vídeos possam ser representados diretamente. É, porém, aconselhável definir-se representações adicionais a estes itens. Como exemplo, usa-se palavras-chave para contextualizar imagens e vídeos. (WANGENHEIM; WANGENHEIM, 2003 p.71).



**Figura 2: Representação de um Caso**

**Fonte: (WANGENHEIM; WANGENHEIM, 2003)**

Existem várias abordagens para realizar o armazenamento da estrutura dos casos em uma base de dados, neste trabalho é utilizada a **representação atributo-valor**, por ser uma técnica simples e com uma baixa curva de aprendizado, o que traz um resultado satisfatório em termos de desempenho e praticidade para o RBC. Os dados são representados utilizando pares de atributo-valor, como pode ser visto na Tabela 1, que exemplifica uma ordem de serviço em uma empresa de tecnologia. Na Figura 2, também ocorre o mesmo tipo de representação, inclusive na solução do caso.

Atributo	Valor
Nível de conhecimento	<i>avançado</i>
Tipo de tarefa	<i>analisar</i>
Experiência	<i>iniciante</i>
Disponibilidade	<i>livre</i>
Orçamento	1.000
Descrição	"Alterar formatação campo"

**Tabela 1: Representação Atributo-Valor**

**Fonte: Dos Autores (2015)**

#### 2.4. Representação Atributo-Valor

Praticamente, todo tipo de informação pode ser armazenado nesta estrutura de armazenamento, desde um simples valor do tipo “sim” ou “não” até arquivos binários (imagens, áudio, documentos). Geralmente estes atributos estão associados a um domínio (tipo de dado). Analisando-se a Tabela 1, o atributo “Orçamento” está no domínio dos *números*, “Descrição” nos *textos*, os demais estão em uma categoria mais complexa de domínio denominada *símbolos*, onde é necessário especificar todos os possíveis símbolos para um atributo, estes ainda podem ser subdivididos em *ordenados e não-ordenados* (WANGENHEIM; WANGENHEIM, 2003, p73).

Na seção 2.6.2, serão explicados e apresentados exemplos destes símbolos.

Nem todos os atributos são úteis para a etapa de recuperação do ciclo RBC, mas sim no contexto geral de uma aplicação, por exemplo, atributos que armazenam informações de auditoria de um caso (data, hora e usuário de inclusão) ou documentos externos anexados.

Entre os atributos úteis, deve-se escolher quais contribuirão na obtenção dos melhores resultados. Esta parte do RBC é importante, pois determinam-se os índices da estrutura de dados. A seguir, na seção 2.5, serão explicados mais detalhadamente os índices e as qualidades de um bom índice.

#### 2.5. Indexação

Para que se possam encontrar casos similares na base de casos para um problema dado, é necessário definir quais atributos serão utilizados para realizar a comparação entre um caso e a situação presente. Estes atributos são chamados de **índices**, em outras palavras, são

combinações dos atributos mais importantes, que permitem distinguir casos úteis para um problema apresentado (WANGENHEIM; WANGENHEIM, 2003).

Índices devem ser construídos de forma suficientemente abstrata e concreta para que possam prever e facilmente se adaptar ao futuro e ainda assim sejam úteis com o crescimento da base de casos (KOLODNER, 1993).

## 2.6. Similaridade

Ao processo de distinguir quais casos dentre vários casos armazenados em uma base têm potencial para ser mais útil que outros dá-se o nome de similaridade, podendo ser global ou local. Neste processo podem ser usadas diversas funções para realizar o cálculo de similaridade.

A similaridade pode ser calculada globalmente para um caso, por exemplo, contando os atributos do caso que apresentam valores iguais, ou também localmente, considerando similaridades entre os valores de um atributo (WANGENHEIM; WANGENHEIM, 2003).

### 2.6.1. Similaridade Global

Este cálculo é utilizado para determinar a utilidade de um caso em relação a um novo caso apresentado, ele depende do resultado das funções de similaridade local realizadas em cada um dos atributos, que serão explicadas na seção 2.6.2. Existem diversos enfoques para determinar esta relevância global, a seguir, serão listados os principais com exemplos:

- **Vizinho mais próximo** (*Nearest neighbour*), é usada em todas as ferramentas estudadas neste trabalho, bem como, em (KIPPER *et al.*, 2014), que diz que esta técnica consiste em multiplicar todos os atributos por um fator (peso) e calcular a sua somatória. Esta soma representa a distância entre casos em um plano geométrico e na Equação 1 visualiza-se sua representação matemática, onde  $a_i$  é um dos atributos do caso apresentado,  $b_i$  é um dos atributo do caso que está sendo comparado e  $w_i$  é o peso atribuído pelo especialista. Na Tabela 2, pode ser visualizado um exemplo, com similaridades locais previamente calculadas, onde são comparados dois casos tendo como resultado a similaridade local. Usando este exemplo, pode-se ter o seguinte cálculo para determinar a similaridade global:  $(5 * 0,9 + 1 * 0,8 + 4 * 0,8 + 3 * 1,0) / (5 + 1 + 4 + 3) = 0,88$ , ou seja, o novo caso é 88% similar ao caso armazenado.
- **Distância Euclidiana** (*Euclidean distance*) da mesma forma que na técnica anterior, também usa distância entre coordenadas geométricas para determinar a similaridade. Mas, usando a fórmula de Euclides que retorna a real distância entre os pontos como



resultado da Equação 2, onde  $a_i$  é um dos atributos do caso apresentado,  $b_i$  é um dos atributo do caso que está sendo comparado e  $w_i$  é o peso atribuído pelo especialista.

$$\frac{\sum_{i=0}^n f(a_i, b_i) \cdot w_i}{\sum_{i=0}^n w_i}$$

**Equação 1: Vizinho mais próximo**

**Fonte: Adaptado de (WANGENHEIM; WANGENHEIM, 2003)**

$$\sqrt{\frac{\sum_{i=0}^n f(a_i, b_i)^2 \cdot w_i}{\sum_{i=0}^n w_i}}$$

**Equação 2: Distância Euclideana**

**Fonte: Adaptado de (WANGENHEIM; WANGENHEIM, 2003)**

Atributo	Importância (Peso)	Caso (Novo)	Caso (Armazenado)	Similaridade Local
Nível de conhecimento	máxima(5)	avançado	alto	0.9
Tipo de tarefa	baixa(1)	analisar	desenvolver	0.8
Experiência	alta(4)	iniciante	média	0.8
Disponibilidade	média(3)	livre	livre	1.0

**Tabela 2: Exemplo Similaridade**

**Fonte: Dos Autores (2015)**

### 2.6.2. Similaridade Local

O cálculo de similaridade local é usado para realizar a comparação entre os atributos de um caso com outro caso, o resultado é usado para determinar a utilidade dos casos na etapa de recuperação.

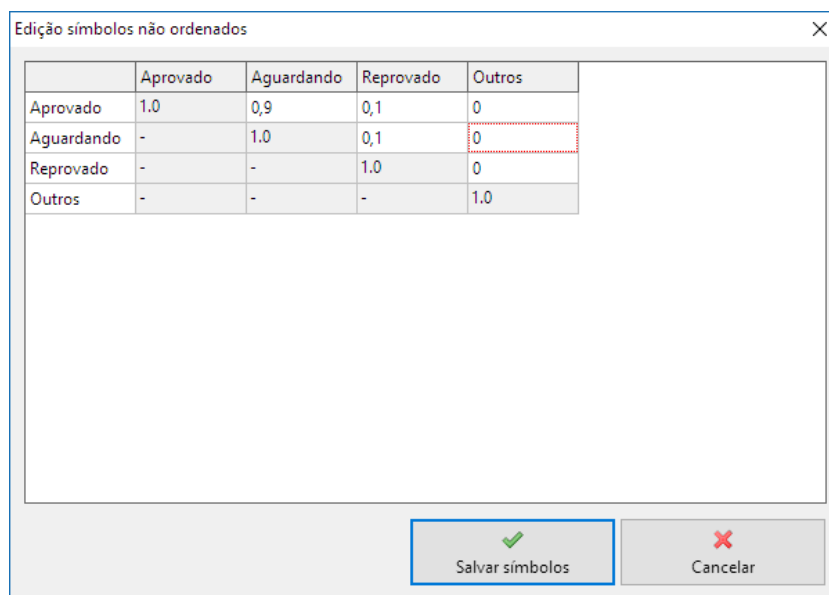
Para cada tipo de dado dos atributos, diferentes abordagens podem ser utilizadas para determinar a similaridade. A seguir serão listados os tipos e as técnicas mais comuns:

- **Números**, não há uma forma de comparar dois números, seja inteiro ou real, e determinar quão similares eles são sem uma função de comparação. Segundo (WANGENHEIM; WANGENHEIM, 2003) é possível utilizar:
  - **Função degrau** (*step function*), calcula-se usando a comparação do módulo da diferença dos valores entre dois casos com um limiar fornecido por especialista. Então, quando este resultado é inferior significa que o caso é totalmente similar,

retornando o valor 1,0, do contrário é considerado dissimilaridade com o valor 0,0.

- **Função linear**, calcula-se usando a divisão do módulo da diferença entre dois casos pela diferença entre extremidade superior e inferior de um limiar fornecido por especialista. Como resultado deste cálculo tem-se uma similaridade progressiva onde quanto menor a distância entre os valores maior é a similaridade. Esta função pode ser vista em (POSSELT, 2011) como rampa esquerda e rampa direita utilizada para cálculos difusos.
- **Textos**, calcular similaridade de textos pode ser uma tarefa complexa e com alto custo computacional, (WANGENHEIM; WANGENHEIM, 2003, p.132) sugere que sempre que possível um texto seja substituído por símbolos e ainda cita alguns enfoques como **correspondência exata** e **correção ortográfica** ao utilizar uma única palavra e **contagem de palavras** ao usar frases. No estudo das ferramentas deste trabalho, do capítulo 3, enfoques mais abrangentes podem ser vistos:
  - Correspondência exata e *case insensitive*, ambas são largamente utilizadas em bancos de dados relacionais e em linguagens de programação. Em resumo, quando o teste for do tipo exato a palavra “impressora” em um atributo de um caso terá similaridade total quando no outro caso também constar a palavra “impressora”, já em *case insensitive*, a palavra “impressora” é considerada totalmente similar à “IMPRESSORA” ou “Impressora”, por exemplo.
  - **Trigrama** consiste em gerar vetores com base no texto do atributo dos casos, estes vetores depois de construídos e normalizados são usados para determinar a similaridade calculando o cosseno do ângulo formado por dois vetores. Este algoritmo pode ser melhorado usando técnicas como *word stemming* e *text normalization* (TARDELLI, 2008). Este método não será usado neste projeto, sugere-se como um trabalho futuro. Em substituição inclui-se o método fonético *metaphone* que será abordado no próximo item.
  - **Metaphone**, é um algoritmo criado por (PHILIPS, 1990), que gera uma codificação específica para cada palavra baseado em sua pronúncia. Originalmente a implementação foi feita para a língua inglesa. As adaptações feitas para a língua espanhola e portuguesa que estão disponíveis em <http://informatica.varzeapaulista.sp.gov.br> e <http://swoodbridge.com/> respectivamente, são utilizadas neste projeto.

- **Frase** é um conjunto de textos, a similaridade é calculada através de um método que elimina textos com tamanho inferior ou igual a três, elimina símbolos de pontuação e elimina repetições do mesmo texto na frase. Após, realiza-se uma contagem da intersecção de textos entre os casos que é dividida pela soma de textos distintos entre os casos, como resultado obtém-se a medida de similaridade. Este projeto inclui como opção, uma variação onde aplica-se o algoritmo *metaphone* nas etapas de preparação da frase.
- **Símbolos ordenados** são valores numéricos finitos e ordenados atribuídos a textos que são compreensíveis, como por exemplo: (1 = Baixo, 2 = Médio, 3 = Alto), em seguida aplicam-se as mesmas funções usadas em tipos numéricos.
- **Símbolos não ordenados** são valores finitos em forma de texto que não possuem ordem definida, portanto, impossível de utilizar funções numéricas. Assim, para determinar a similaridade deve-se estabelecer uma relação entre cada um dos itens da lista. (WANGENHEIM; WANGENHEIM, 2003) usa como exemplo a distância entre cidades, atribuindo peso alto para as cidades próximas e baixo para as cidades distantes. Na Figura 3, pode-se visualizar a edição de uma tabela contendo as relações simétricas. Observa-se que “Aprovado” e “Aguardando” são considerados 90% (0,9) similares, em contraste dos demais que possuem uma similaridade de 10% (0,1).



	Aprovado	Aguardando	Reprovado	Outros
Aprovado	1.0	0,9	0,1	0
Aguardando	-	1.0	0,1	0
Reprovado	-	-	1.0	0
Outros	-	-	-	1.0

At the bottom of the window, there are two buttons: 'Salvar símbolos' (Save symbols) with a green checkmark icon, and 'Cancelar' (Cancel) with a red X icon.

**Figura 3: Tabela Comparativa**

**Fonte: (WANGENHEIM; WANGENHEIM, 2003)**

Neste capítulo, foram vistos todos os elementos que compõem a estrutura do RBC, as etapas do ciclo RBC; o que é um caso e como ele é representado; indexação e como ela pode influenciar no desempenho da recuperação de casos; atributos de casos e como as funções de similaridade são aplicadas aos seus valores.

A seguir, no capítulo 3, é apresentado um estudo sobre ferramentas de RBC.

### 3. FERRAMENTAS

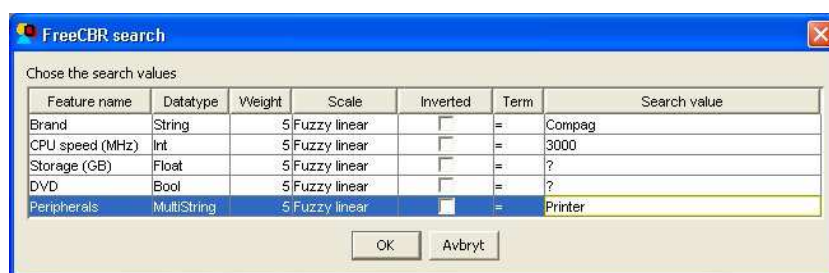
Neste estudo apresenta-se uma comparação entre as ferramentas que permitem planejar, desenhar e criar projetos que utilizam técnicas de RBC. Algumas delas não possuem mais um *site* oficial, mas ainda assim foram analisadas por meio de outras fontes. Como, por exemplo, a ferramenta CBR-Works sugerida em (WANGENHEIM; WANGENHEIM, 2003).

#### 3.1. CBR-Works (<http://www.inf.ufsc.br/~patrec/cbr.html>)

Possui uma interface gráfica de usuário para criação de projetos, interface Web e inserção de casos; a ferramenta é para fins acadêmicos. Nem *site* oficial, documentação, *downloads* ou código fonte foram encontrados, assim não foi possível realizar a instalação da ferramenta para análise. Embora, algumas características podem ser encontradas na internet e em (WANGENHEIM; WANGENHEIM, 2003), como a possibilidade de rodar em sistemas Linux e Windows e acessar qualquer banco de dados através da interface ODBC (*Open Database Connectivity*). Por fim, a ferramenta permitia determinar a similaridade global através das principais técnicas e ainda possuía uma opção onde o analista podia customizar o cálculo.

#### 3.2. FreeCBR (<http://freecbr.sourceforge.net>)

Esta ferramenta possui várias formas para sua utilização: GUI (*Graphical User Interface*), como pode ser visto na Figura 4, linha de comando, Java bean, MS ActiveX e as bibliotecas Java em forma de API (*Application programming interface*), pois o código fonte é aberto e livre para usar e fazer alterações. O cálculo da similaridade é obtido usando: Distância Euclidiana, Algoritmo de Distância Normal e Algoritmo de Distância Logarítmica.



**Figura 4: Interface FreeCBR**

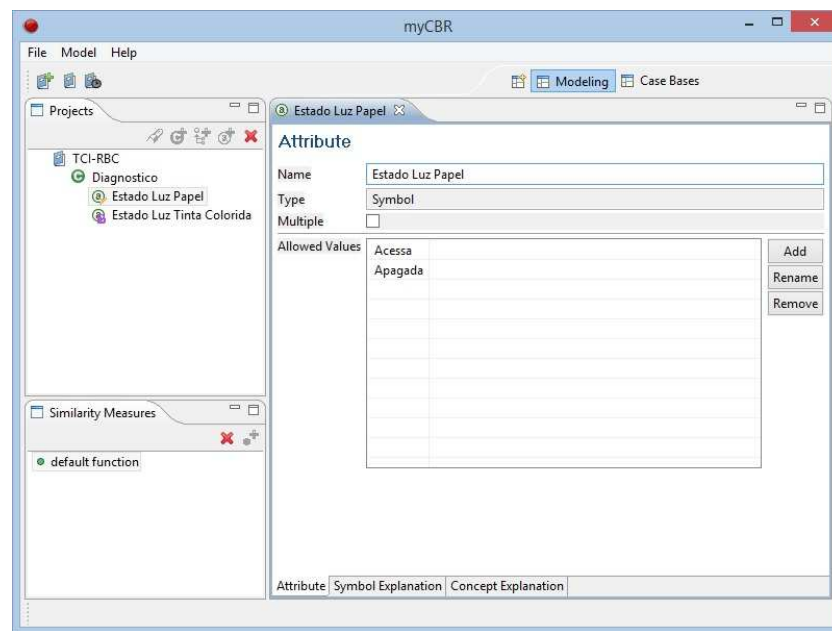
**Fonte:** <http://freecbr.sourceforge.net/screenshots.shtml>, acesso em abril, 2015

#### 3.3. myCBR (<http://mycbr-project.net>)

Esta é uma ferramenta de código aberto, que segundo o *site* do autor possibilita modelar e testar rapidamente aplicações sofisticadas. Possui uma interface gráfica de usuário, SDK para desenvolvimento em forma de *plugin* para Eclipse. Também possui uma completa

documentação de apoio que inclui tutoriais para guiar os desenvolvedores. Um dos diferenciais é a possibilidade de usar arquivos do tipo CSV (*Comma Separated Value*) para importar as definições de casos, onde para cada coluna do arquivo é gerado um atributo.

Os métodos para cálculo de similaridade global desta ferramenta são: Soma ponderada, diferença Euclidiana, máximos e mínimos. A Figura 5 mostra uma tela da interface da ferramenta.



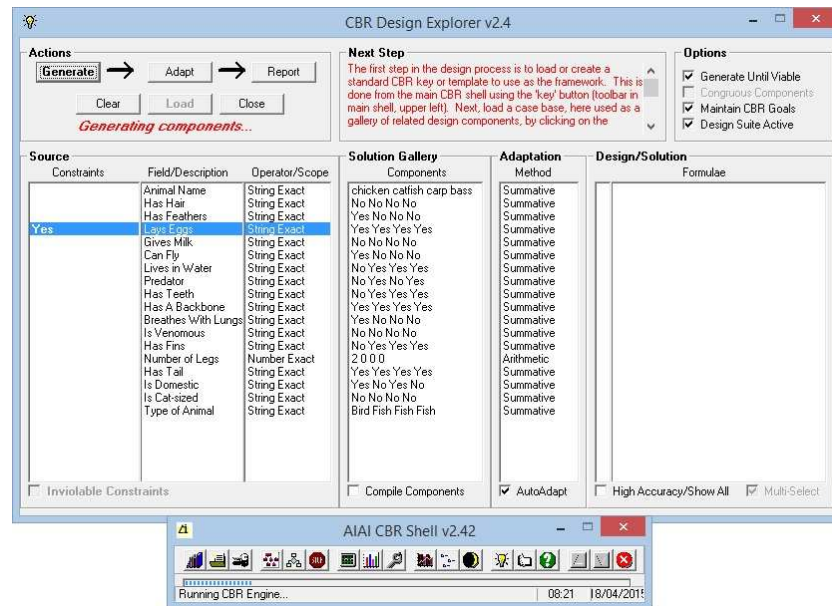
**Figura 5: Interface myCBR**

**Fonte: Captura de Tela da Ferramenta**

### 3.4. CBR Shell (<http://www.aiai.ed.ac.uk/project/cbr>)

Stuart Aitken do AIAI (*Artificial Intelligence Applications Institute*) desenvolveu a ferramenta CBR Shell (Figura 6), que é voltada para uso acadêmico e foi usada nos projetos RBC do instituto. O site não possui documentação de apoio, apenas *links* para *download* das duas versões disponíveis. A versão 1.0 foi desenvolvida em JAVA, podendo ser utilizada em qualquer sistema operacional, já a 2.0 é específica para sistemas Windows.

A similaridade global é calculada utilizando o algoritmo do vizinho mais próximo e a similaridade local é calculada utilizando algoritmos comuns à lógica difusa. Também utiliza o algoritmo Trigramas para comparação de textos, citado anteriormente na seção 2.6.2.



**Figura 6: Interface CBR Shell**

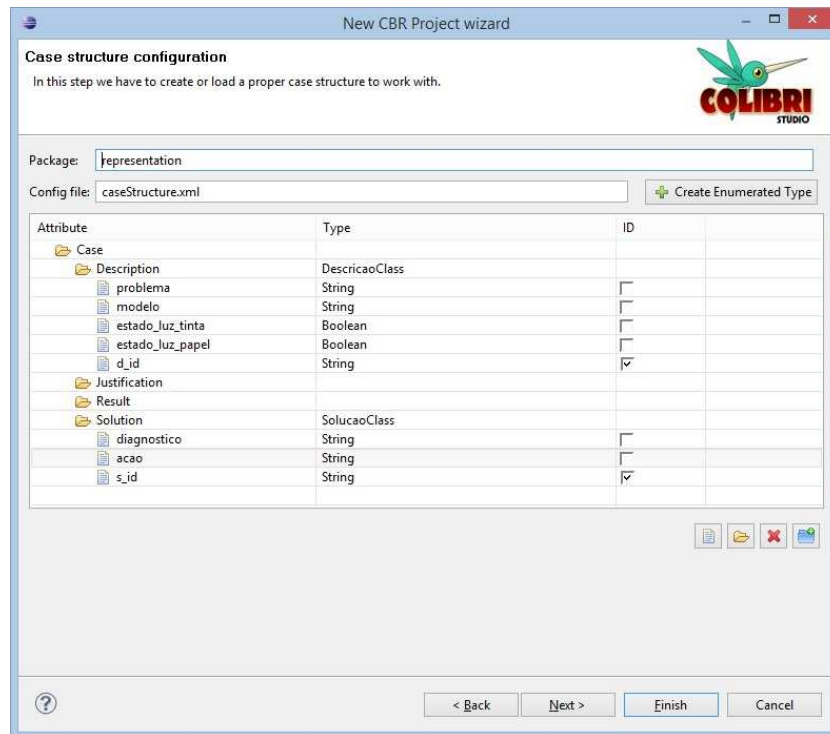
**Fonte: Captura de Tela da Ferramenta**

### 3.5. jCOLIBRI (<http://gaia.fdi.ucm.es/research/colibri/jcolibri>)

De acordo com o *site* do autor, o *framework* jCOLIBRI foi criado para servir de referência no desenvolvimento de aplicações RBC, desenhada para ser extensível e reutilizável. Possui *links* para material de apoio como: documentação, tutoriais e exemplos.

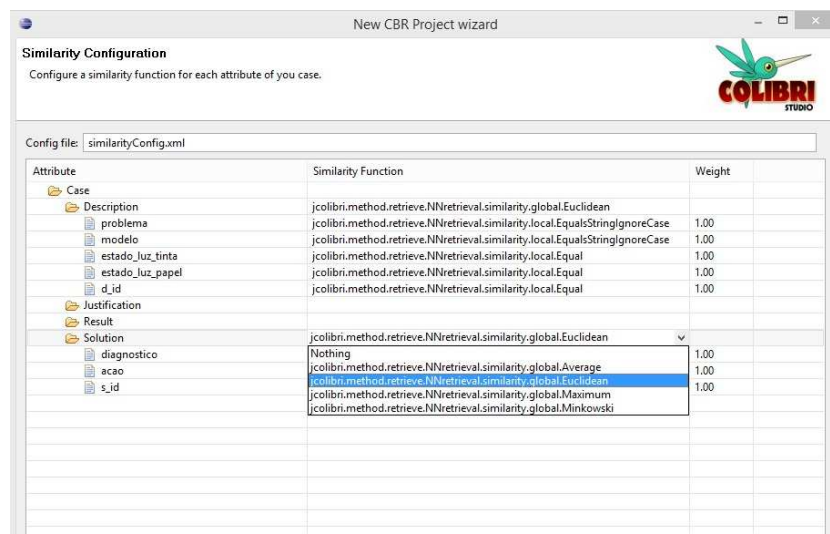
Esta ferramenta foi desenvolvida em JAVA, como *plugin* do Eclipse, sua primeira versão foi disponibilizada em 2005 e o desenvolvimento continua ativo, o que proporciona experiência na área RBC. Dentre as ferramentas analisadas, esta é a mais completa, todas as etapas do Ciclo RBC estão implementadas, a similaridade global pode ser calculada usando: algoritmo de Minkowski, média, máximo e distância euclidiana e trinta algoritmos para cálculo de similaridade local.

Ao criar um novo projeto na ferramenta um *wizard* é apresentado, neste devem ser definidos os atributos do caso e quais algoritmos de similaridade serão usados. A Figura 7 e a Figura 8 apresentam duas telas da ferramenta, respectivamente, em relação à representação dos casos e às funções de similaridade.



**Figura 7: Interface jCOLIBRI – Representação**

Fonte: Captura de Tela da Ferramenta



**Figura 8: Interface jCOLIBRI – Similaridade**

Fonte: Captura de Tela da Ferramenta

### 3.6. Análise Comparativa das Ferramentas Estudadas

Os primeiros passos realizados com cada uma das ferramentas analisadas foi o acesso ao *site*, *download* e instalação. Ficando evidente que todas são voltadas para o meio acadêmico com muita informação técnica e referencial teórico.



myCBR e jCOLIBRI são as mais avançadas e o seu desenvolvimento continua ativo. Ambas são distribuídas juntamente com Eclipse e permitem gerar código JAVA para ser usado diretamente em aplicações em forma de SDK. FreeCBR e CBR Shell em contraste são ferramentas desatualizadas e com uma interface poluída e de difícil compreensão.

O estudo comparativo de ELKAFRAWY; MOHAMED (2014), aplicou a mesma base de casos em todas as ferramentas obtendo bons resultados e também concluíram que myCBR e jCOLIBRI são mais avançadas por implementarem todas as etapas do Ciclo RBC. Já as outras focam apenas na etapa de recuperação de casos similares.

Uma das vantagens de jCOLIBRI sobre os demais é a capacidade de se conectar com bases de dados externas, o que possibilita o uso de *SQL Query's* facilitando a seleção de dados e aproveitando o poder do SGBD (Sistema Gerenciador de Banco de Dados), enquanto os demais utilizam arquivos texto ou XML.

Nas tabelas 3 e 4, é possível analisar o comparativo dos recursos e desempenho das ferramentas, na última linha inclui-se a ferramenta deste projeto.

Ferramenta	Permite Indexação?	Interface Gráfica	Formato Arquivo Casos	Permite Conexão BD
jCOLIBRI	Sim	Sim	CSV e XML	Sim
myCBR	Não	Sim	CSV e XML	Não
FreeCBR	Não	Sim	Texto	Não
CBR Shell	Não	Sim	Texto	Não
<b>CBRain</b>	<b>Sim</b>	<b>Sim</b>	<b>CSV</b>	<b>Sim</b>

**Tabela 3: Comparativo de Recursos da Interface do Usuário**

**Fonte: Dos Autores (2015)**

Ferramenta	Similaridade Global	Recuperação Casos	Revisão Casos
jCOLIBRI	Vizinho Mais Próximo	Árvores	Automática
myCBR	Vizinho Mais Próximo	Sequencial (com filtro)	Manual
FreeCBR	Distância Euclideana	Sequencial	Manual
CBR Shell	Vizinho Mais Próximo	Árvores	Manual
<b>CBRain</b>	<b>Vizinho Mais Próximo e Distância Euclidiana</b>	<b>Sequencial (com filtro)</b>	<b>Manual</b>

**Tabela 4: Comparativo de Recursos Técnicos**

**Fonte: Adaptado de (ELKAFRAWY; MOHAMED, 2014)**

### 3.7. Considerações

O tema RBC é abrangente e existem estudos variados nesta área que está em constante evolução.

Com a pesquisa exploratória e bibliográfica realizada, obteve-se o conhecimento necessário sobre RBC para estudar as ferramentas existentes e encontrar aspectos que poderiam ser melhorados.

Em cálculos de similaridades observa-se uma grande possibilidade de expansão do processo de RBC, pois é possível usar funções e algoritmos de outras áreas de inteligência artificial. Funções citadas nas seções 2.6.1 e 2.6.2 são apenas as mais básicas e não representam tudo que o RBC pode fazer. A ferramenta CBR-Works, que não está mais disponível, possuía um recurso interessante, permitia a inserção de uma função definida pelo usuário ao invés das predefinidas para o cálculo de similaridade global. Este recurso poderá ser utilizado na ferramenta proposta e estendê-lo para similaridade local.

Com base no estudo das ferramentas deste capítulo, os seguintes aspectos foram desenvolvidos na ferramenta CBRAIN:

- Interface gráfica com suporte à internacionalização e adaptação ao tema do sistema operacional.
- Multiplataforma: Foi compilado e testado em Windows e Linux, compatível com Mac.
- Armazenamento de casos revisados em banco de dados não relacional.
- Acesso direto a bases com casos, aproveitando-se do desempenho dos sistemas gerenciadores de banco de dados, como por exemplo os índices.
- Apresentação de novos casos utilizando-se do protocolo HTTP (*Hypertext Transfer Protocol*) com as tecnologias SOAP (*Simple Object Access Protocol*) e JSON-RPC (*JavaScript Object Notation Remote Procedure Call*)

## 4. FERRAMENTA PARA RACIOCÍNIO BASEADO EM CASOS

O trabalho é caracterizado por ser uma pesquisa exploratória, qualitativa e bibliográfica, devido às pesquisas que foram feitas de algumas ferramentas RBC disponíveis e em documentos, livros e artigos publicados sobre os assuntos envolvidos na pesquisa, bem como em trabalhos relacionados já desenvolvidos. A sua natureza qualitativa refere-se à validação realizada, a fim de buscar aspectos de usabilidade e funcionamento da ferramenta junto à percepção de usuários.

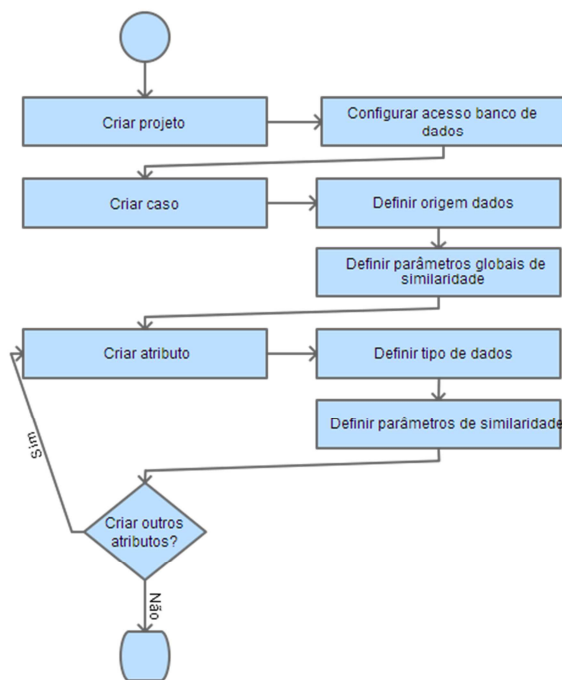
Nas próximas seções descreve-se como a ferramenta CBRAIN foi desenvolvida e como pode ser utilizada para a modelagem de aplicações RBC.

### 4.1. Introdução

CBRAIN é uma ferramenta completa para desenvolvimento de projetos RBC, que utiliza tecnologias e conceitos atuais, visando uma experiência agradável ao usuário. Disponibiliza suporte a múltiplos idiomas e a múltiplas plataformas. Pode ser usada como ferramenta de criação e testes.

Composto por dois módulos principais: interface gráfica do usuário e o serviço de processamento. Este serviço é subdividido em três partes, o conjunto de funções para os cálculos de similaridade, um banco de dados não relacional para armazenar casos revisados e ou adaptados, e um gerador de arquivos WSDL (*Web Services Description Language*) para a integração com sistemas existentes. Esta separação em módulos otimiza o desenvolvimento e facilita a escalabilidade do sistema.

A Figura 9 ilustra o fluxo de criação de um novo projeto RBC utilizando a interface gráfica. Ela demonstra todos os passos até a finalização do projeto, até o ponto em que é possível realizar testes de recuperação e reúso de casos. Os passos são iniciados em “Criar projeto”, onde se faz necessária a configuração do acesso à base de dados, após, em “Criar caso”, e passando por “Definir origem dos dados” e “Definir parâmetros de similaridade global”, uma nova configuração para um caso é criada. Por fim, definir todos os atributos em “Criar atributo”, passando por “Definir tipo de dados” e “Definir parâmetros de similaridade local”, até que não se deseja mais adicioná-los.



**Figura 9: Fluxo Criação de Projeto na Interface Gráfica**

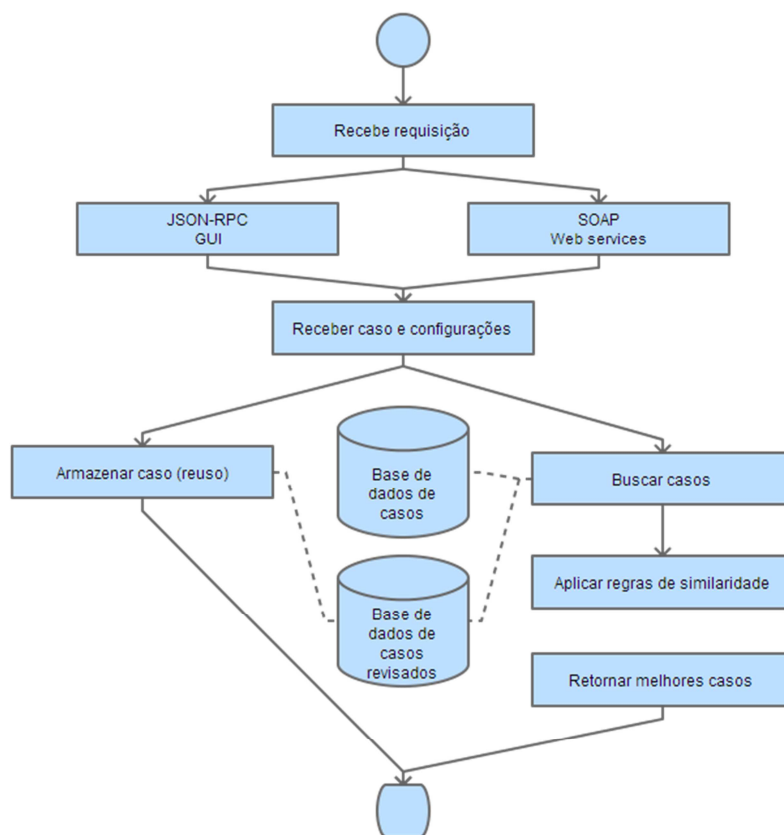
**Fonte: Dos Autores (2015)**

A Figura 10 ilustra o funcionamento do serviço de processamento. Tal serviço foi construído com dois tipos de protocolo de comunicação - JSON-RPC e SOAP, o primeiro é usado como padrão para a integração com a interface gráfica, o segundo é disponibilizado como padrão para integração com outros sistemas. SOAP é um protocolo consolidado e disponível em praticamente todos os ambientes de desenvolvimento atuais.

No primeiro item, “Recebe requisição”, determina-se o protocolo de comunicação direcionado ao interpretador correspondente. Após a validação da requisição em “Receber caso e configurações”, extrai-se as configurações, os dados do caso apresentado e o método que deve ser executado.

Caso o método solicitado seja uma requisição para recuperação de casos, o fluxo é direcionado para “Buscar casos”, fazendo-se a recuperação da base de casos e também da base de casos revisados. No passo “Aplicar regras de similaridade”, aplica-se as funções de similaridade local em cada um dos atributos e a função de similaridade global sobre o resultado destas funções. Por fim, o passo “Retornar melhores casos” constrói um vetor com os melhores resultados adicionando dois elementos fixos: pontuação, que é o resultado da função de similaridade global; e origem, que indica de qual base de dados o caso foi recuperado.

Caso o método solicitado seja uma requisição para armazenar um caso, o fluxo é direcionado para “Armazenar caso”, fazendo a retenção na base de casos revisados, correspondendo à etapa de reuso, revisão e retenção do ciclo RBC.



**Figura 10: Fluxo do Funcionamento Serviço de Processamento**

**Fonte: Dos Autores (2015)**

Nas próximas seções serão explicados individualmente estes módulos.

#### 4.2. Serviço de Processamento

Executado em segundo plano, realiza todas as tarefas referentes ao ciclo de RBC, conforme visto na seção 2.2 do capítulo 2.

Foi utilizada a linguagem PHP, que permite desenvolvimento rápido de aplicações, possui tecnologias avançadas e amplo acesso a gerenciadores de bancos de dados. A partir da versão 5.5 é possível executar de um servidor web integrado, dispensando a instalação de aplicações como Apache (<http://www.apache.org/>) e Nginx (<https://www.nginx.org/>).

Dois arquivos do projeto são responsáveis pelo ponto de entrada para as requisições de uso do RBC, ambos instanciam a classe contida em **class.Main.php** e publicam seus métodos:

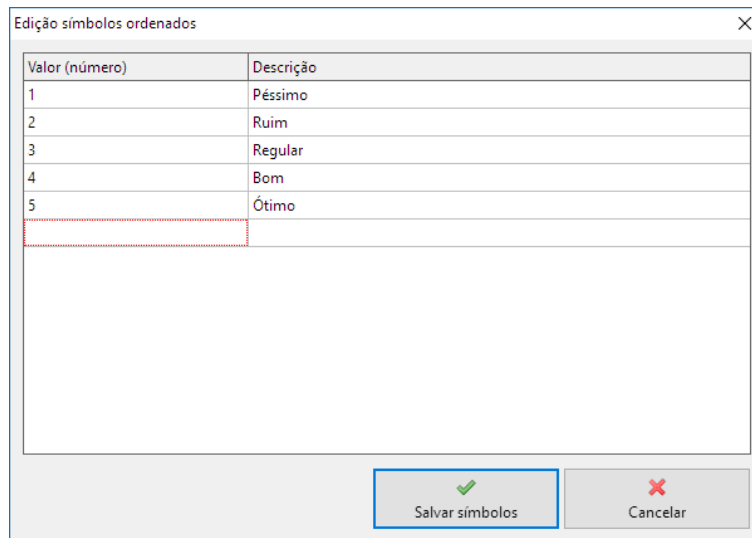
- **rpc.php**, que implementa o protocolo JSON-RPC, usado por padrão pela ferramenta gráfica do projeto. Pode ser usado por outras aplicações para integração.
- **ws.php**, que implementa o protocolo SOAP, amplamente conhecido e usado para integrações entre aplicações.

O método **exec** da classe **Main** é o responsável pelo processamento RBC, ele recebe todas as definições de um caso feitas pela interface gráfica e o caso que está sendo apresentado para processamento. A busca por casos armazenados é iniciada e em seguida a iteração para aplicação das funções de similaridade. A seguir na subseção 4.2.1 descreve-se as funções implementadas para o projeto.

#### 4.2.1. Implementação das Funções de Similaridade

Após o estudo realizado de funções de similaridade no capítulo 2, implementou-se a lista a seguir:

- **Número**
  - **Igualdade simples**, similaridade quando o resultado é exatamente igual a outro.
  - **Degrau**, similaridade quando o resultado é inferior ao limiar fornecido.
  - **Linear**, o resultado da similaridade é progressivo, utilizando-se dos limiares fornecidos.
- **Texto**
  - **Case sensitive**, similaridade quando a comparação entre textos é exata.
  - **Case insensitive**, similaridade quando a comparação entre textos é exata, desconsiderando-se as diferenças de maiúsculas e minúsculas.
  - **Metaphone**, similaridade baseada na pronúncia entre textos.
- **Frase**
  - **Intersecção**, busca baseada em contagem de palavras distintas.
  - **Intersecção e metaphone**, busca baseada em contagem de pronúncias distintas
- **Símbolos ordenados**, utilizam-se das mesmas funções aplicadas aos números. A edição dos símbolos visualiza-se na Figura 11, onde a cada valor numérico atribui-se uma descrição.
- **Símbolos não ordenados**, utilizam-se de uma tabela comparativa para atribuição de similaridades entre símbolos, a edição desta tabela visualiza-se na Figura 3, apresentada na seção 2.6.2, do capítulo 2.



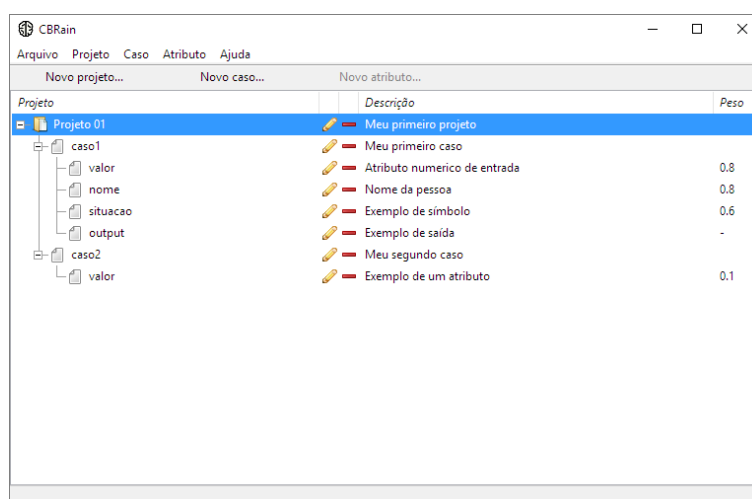
**Figura 11: Tela Símbolos Ordenados**

**Fonte: Dos Autores (2015)**

### 4.3. Interface da Ferramenta

Na tela principal exibe-se um menu no topo com acesso a todos os recursos e, a seguir, uma barra de botões de acesso rápido aos principais recursos.

Uma visão em formato de árvore na parte central exibe os projetos carregados com seus casos e atributos. Todos os itens possuem dois botões de ação: editar e excluir. Os campos “Descrição” e “Peso”, estão posicionados à direita e complementam a visualização, a edição de ambos é feita ao clicar no botão editar, conforme visualiza-se na Figura 12. Nas próximas seções serão apresentadas as telas acessíveis pelo menu principal em uma sequência lógica para criação de um projeto RBC, que contemplará configurações, cadastros, testes e reúso.



**Figura 12: Tela Principal**

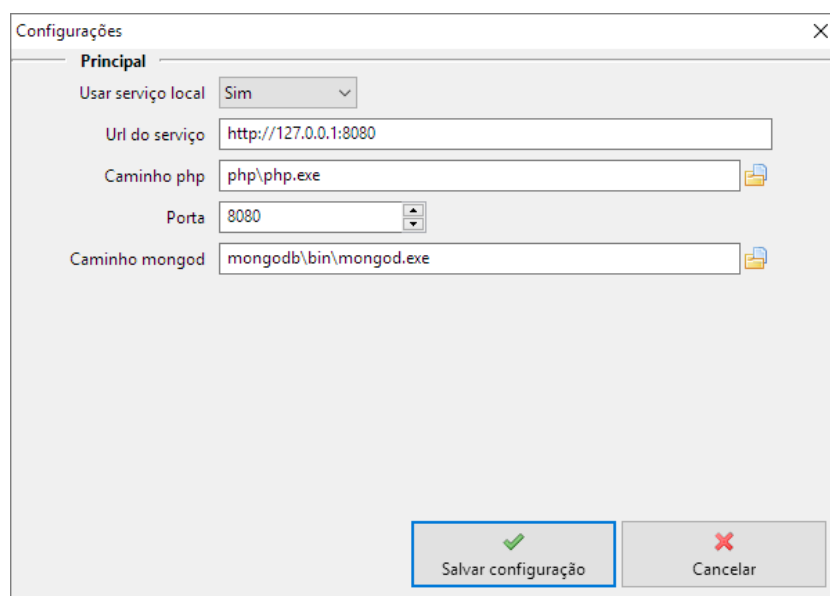
**Fonte: Dos Autores (2015)**

#### 4.4. Configurações

Através do menu Arquivo item Configurações exibe-se a tela com as configurações básicas da ferramenta gráfica. Todas possuem o valor padrão como visualiza-se na Figura 13.

Este projeto permite que o processamento do RBC seja executado em outro equipamento, neste caso é necessário alterar o valor do campo “Usar serviço local” para “Não” e indicar o endereço desejado no campo “Url do serviço”. Estas configurações serão persistidas em um arquivo na pasta do usuário com o formato JSON.

O comportamento padrão é iniciar uma instância do servidor web e uma instância do servidor de banco de dados durante a inicialização da interface gráfica, ambos são distribuídos junto com o projeto.



**Figura 13: Tela Configuração**

**Fonte: Dos Autores (2015)**

#### 4.5. Criação Projeto

Através do menu “Projeto” item “Novo projeto” ou clicando-se no botão de mesmo nome na barra de acesso rápido, exibe-se a tela com as definições para criação de um projeto, como visualiza-se na Figura 14.

As duas primeiras informações identificam e descrevem o projeto, no segundo bloco da tela pode-se configurar o acesso ao gerenciador de banco de dados, onde se encontram casos armazenados para a etapa de recuperação. Esta configuração considera-se opcional quando deseja-se usar arquivos CSV para recuperação de casos.



The image shows a software dialog box titled "Projeto". It is divided into two sections: "Principal" and "Configurações de conexão externa".

- Principal:**
  - Nome:
  - Descrição:
- Configurações de conexão externa:**
  - Driver:
  - Usuário:
  - Senha:
  - Conexão:
  - Porta:

At the bottom, there are two buttons: "Salvar projeto" (with a green checkmark icon) and "Cancelar" (with a red X icon).

**Figura 14: Tela Criação Projeto**

**Fonte: Dos Autores (2015)**

#### 4.6. Criação dos Casos

Para criar um caso e atribuí-lo a um projeto deve-se selecionar o projeto na visualização da tela principal. Através do menu “Caso” item “Novo caso” ou clicando-se no botão de mesmo nome na barra de acesso rápido, exibe-se a tela com as definições para criação de um caso, como visualiza-se na Figura 15.

No bloco “Opções de dados” deve-se escolher o caminho do CSV e suas opções. É possível especificar uma tabela ou construir uma consulta SQL, caso o projeto tenha sido configurado para conexão com banco de dados, como visto na seção 4.5 deste capítulo.

No bloco “Opções de raciocínio” existem duas configurações que são utilizadas no momento que o processamento RBC determina se um caso é útil ou não.

Em “Método de similaridade” o usuário decide qual método de similaridade global será aplicado em cada um dos casos recuperados, as opções disponíveis são: Vizinho mais próximo e Distância Euclideana. O funcionamento destes algoritmos foi explicado na subseção 2.6.1 do capítulo 2.

Para cada um dos casos processados obtém-se uma pontuação, se esta pontuação é superior ao informado em “Pontuação relevante”, ele pode ser considerado útil, embora só as maiores pontuações serão consideradas no momento da recuperação dos casos por similaridade.

**Figura 15: Tela Criação Caso**

**Fonte: Dos Autores (2015)**

#### 4.7. Criação de Atributos

Para criar um atributo e atribuí-lo a um caso deve-se selecionar o caso na visualização da tela principal. Através do menu “Atributo” item “Novo atributo” ou clicando-se no botão de mesmo nome na barra de acesso rápido, exibe-se a tela com as definições para criação de um atributo, como visualiza-se na Figura 16.

Um caso pode conter vários atributos, considera-se a configuração deles a parte mais importante do ciclo RBC.

No bloco “Opções de dados” deve-se indicar como será o comportamento do atributo para a etapa de recuperação.

Para o tipo do atributo existem três opções, listadas a seguir:

- Atributo de entrada será enviado para o serviço de processamento como parâmetro para a pesquisa por casos similares.
- Atributo de saída será retornado como resultado da pesquisa de casos similares
- Entrada e saída, é a união dos anteriores, permite que o atributo usado como entrada, também seja retornado como resultado da recuperação de casos.

Um atributo que é usado como entrada e como saída torna-se potencialmente útil, considerando-se que o usuário tenha inserido o nome César em um atributo com o método de similaridade “Metaphone”, poderá retornar o valor Cesar.

Para o tipo de dados pode-se escolher entre texto, frase, número e símbolos. Estão diretamente conectados ao bloco “Opções de raciocínio”, pois para cada um deles, há um conjunto específico de métodos de similaridade com suas opções. As combinações disponíveis foram detalhadas na subseção 2.6.2 do capítulo 2.

No bloco “Opções de raciocínio”, no campo “Indexação” quando o valor é “Sim”, indica ao serviço de processamento que este campo deve ser usado como uma condição de busca na recuperação de casos em um banco de dados, isto auxilia no desempenho, pois, uma quantidade menor de casos serão submetidos aos cálculos de similaridade. No mesmo bloco, o campo “Peso” representa percentualmente quão relevante é este atributo com relação aos demais atributos do caso. Deve ser informado um número real entre 0 e 1, considera-se o número 1 como 100% de relevância.

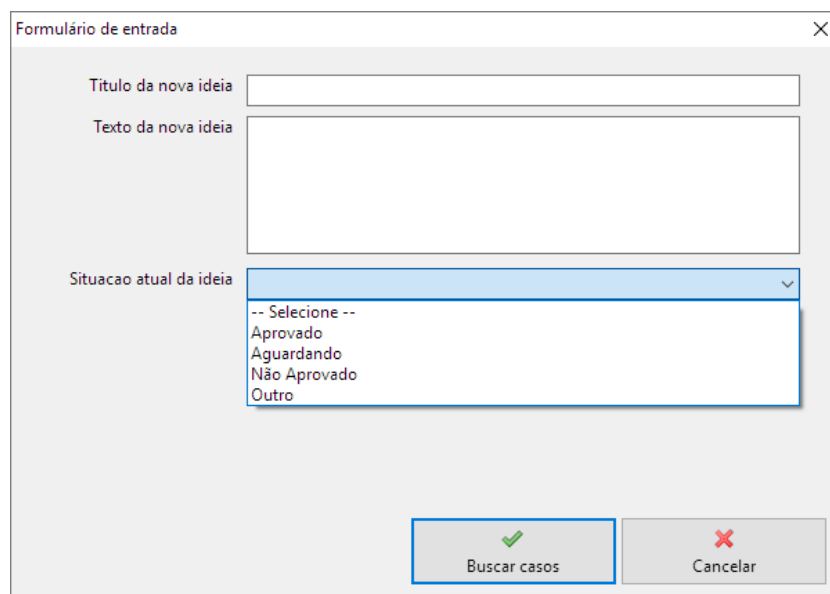
**Figura 16: Tela Criação Atributo**

**Fonte: Dos Autores (2015)**

#### **4.8. Formulário de Entrada**

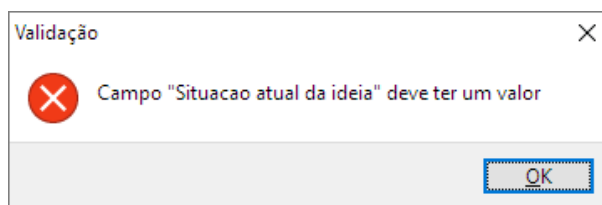
Através do menu “Caso” item “Formulário de entrada” exibe-se um formulário onde todos os atributos de entrada e também os de entrada e saída do caso selecionado, são apresentados ao usuário possibilitando-se a realização do teste de recuperação de casos similares. Os componentes visuais que representam cada um dos atributos, são criados dinamicamente levando em conta o tipo de dado que foi previamente configurado. Por exemplo, para atributos que usam símbolos apresenta-se o componente visual do tipo seleção

e atributos do tipo frase utiliza-se um componente de texto que permite inserção de múltiplas linhas. Todos os campos são considerados obrigatórios, uma notificação é apresentada, caso algo não seja informado, então o foco da edição é reposicionado no campo criticado. Uma amostra deste formulário e da notificação visualiza-se nas Figura 17 e Figura 18 respectivamente.

A screenshot of a web form titled "Formulário de entrada". It contains three input fields: "Titulo da nova ideia" (a single-line text box), "Texto da nova ideia" (a multi-line text area), and "Situacao atual da ideia" (a dropdown menu). The dropdown menu is open, showing options: "-- Seleccione --", "Aprovado", "Aguardando", "Não Aprovado", and "Outro". At the bottom of the form, there are two buttons: "Buscar casos" with a green checkmark icon and "Cancelar" with a red X icon.

**Figura 17: Formulário de Entrada**

**Fonte: Dos Autores (2015)**

A screenshot of a validation message dialog box titled "Validação". It features a red X icon on the left and the text "Campo 'Situacao atual da ideia' deve ter um valor" in the center. An "OK" button is located at the bottom right of the dialog.

**Figura 18: Mensagem Validação Formulário de Entrada**

**Fonte: Dos Autores (2015)**

#### **4.9. Adaptação e Reúso**

Utilizando-se o botão “Buscar casos” do formulário de entrada, exibido na seção anterior, os valores que foram inseridos nos campos são serializados e enviados ao serviço de processamento. Caso existam resultados, eles serão exibidos em uma listagem que pode ser visualizada na Figura 19 e explicada a seguir.

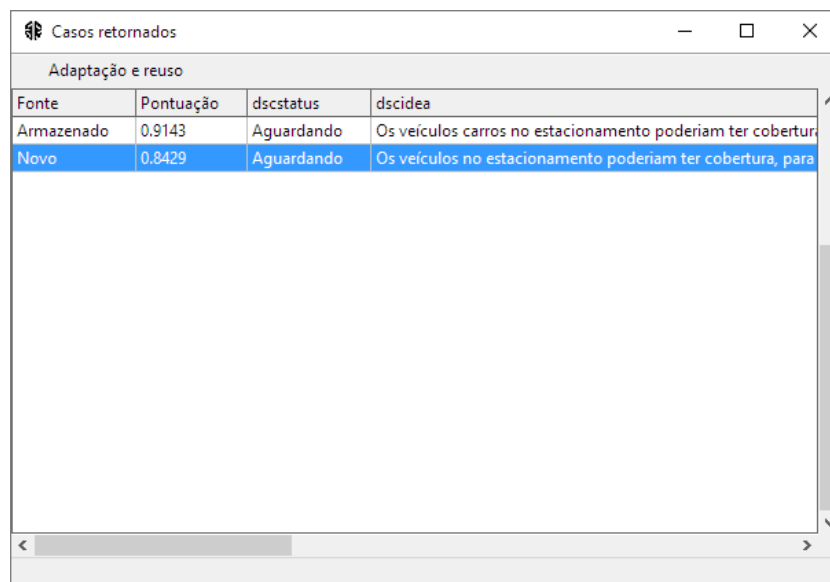
A coluna “Fonte”, apresenta o valor “Armazenado”, se o caso foi recuperado da base de conhecimento, ou seja, casos que foram previamente adaptados e reinseridos para reúso, ou o valor “Novo” para casos recuperados da base de casos configurada no projeto.

A coluna “Pontuação” apresenta a pontuação atingida por estes casos em comparação ao apresentado no formulário de entrada.

As demais colunas são criadas dinamicamente correspondendo aos atributos do caso apresentado, cujo tipo foi indicado como “Saída” ou “Entrada e Saída”. Como exemplo visualiza-se “dscstatus” e “dscidea” na Figura 19.

Os casos apresentados na listagem podem ser revisados, clicando-se duas vezes na linha correspondente ao item, ou selecionando-o e clicando-se no botão “Adaptação e reúso” da barra de acesso rápido. A tela que é exibida é similar ao formulário de entrada, agora com todos os atributos do caso e com os valores do caso recuperado, conforme visualiza-se na Figura 20. Após, podem ser feitas adaptações, ao final clicar no botão “Reusar caso”.

Como exemplo desta adaptação pode-se substituir uma palavra no meio de uma frase para facilitar a recuperação futura, como na Figura 20 onde palavra “veículos”, foi substituída por “veículos, carros e motos” e “cobertura” foi substituída por “cobertura ou telhado”.



Fonte	Pontuação	dscstatus	dscidea
Armazenado	0.9143	Aguardando	Os veículos carros no estacionamento poderiam ter cobertura
Novo	0.8429	Aguardando	Os veículos no estacionamento poderiam ter cobertura, para

**Figura 19: Listagem de Casos Recuperados**

**Fonte: Dos Autores (2015)**

**Figura 20: Formulário Adaptação e Reúso**

**Fonte: Dos Autores (2015)**

#### 4.10. Usabilidade

O desenvolvimento da interface considera os cinco principais conceitos de usabilidade, descritos por (NIELSEN, 1993), que são listados a seguir:

- **Facilidade de aprendizado:** O sistema deve ser intuitivo no seu uso, fazendo com que o usuário consiga utilizar a ferramenta rapidamente e também aprenda com a utilização.
- **Eficiência:** O sistema deve ser produtivo mesmo para usuários avançados.
- **Facilidade de memorização:** O sistema deve ser fácil de memorizar, para que depois de um período sem utilização o processo retomado seja natural.
- **Segurança:** O sistema deve evitar que erros aconteçam direcionando o usuário para uso correto.
- **Satisfação:** O sistema deve ser agradável e causar satisfação aos usuários.

A ferramenta CBRAIN usa textos simples, imagens autoexplicativas, toda a informação útil encontra-se visível e o acesso às telas está disponível em qualquer momento. Atendendo assim, os fatores de usabilidade que tornam o fácil aprendizado e a memorização.

Tratadores globais de exceção foram criados para que eventuais erros sejam capturados, tratando-os de forma amigável e sem prejuízos ao usuário, atendendo ao conceito de segurança.

As operações realizadas nos formulários são rápidas e com prevenção contra cliques consecutivos no momento de confirmar alterações. Estas ações tornam a ferramenta eficiente e juntamente com os demais conceitos proporcionam satisfação.

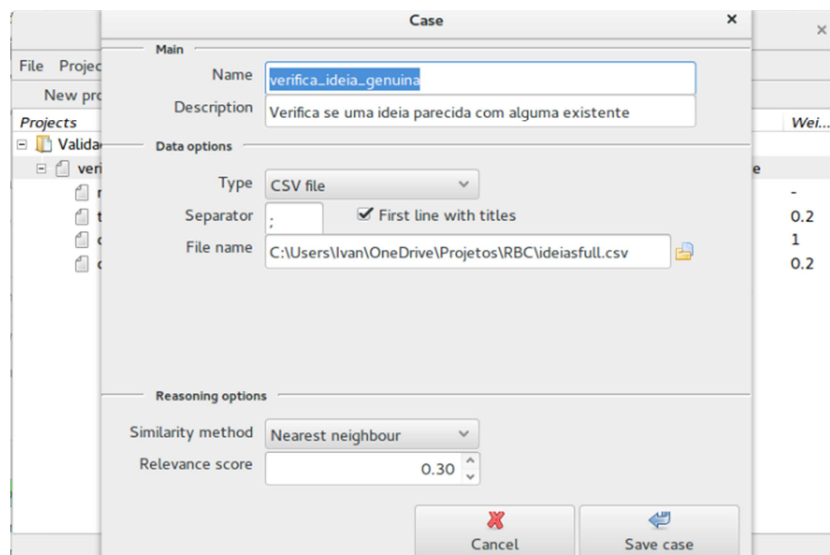
#### 4.10.1. Disposição dos Botões

A disposição dos botões de confirmação e cancelamento, seguem o padrão do sistema operacional onde são executados. De acordo com NIELSEN (2008), tanto “Salvar” e “Cancelar”, como “Cancelar” e “Salvar”, são escolhas plausíveis e uma questão de preferência pessoal. Sugere-se, seguir as regras de interface do sistema operacional. Como exemplo disto, cita-se os sistemas Windows que utilizam o botão OK como primeira opção, já os sistemas Apple colocam a opção de cancelamento como primeira opção. Além disto, recomenda-se colocar um texto que indique a ação que irá ocorrer, ao invés de um simples “OK”, o usuário terá menos dúvidas da ação que quer executar.

#### 4.10.2. Fontes e Cores

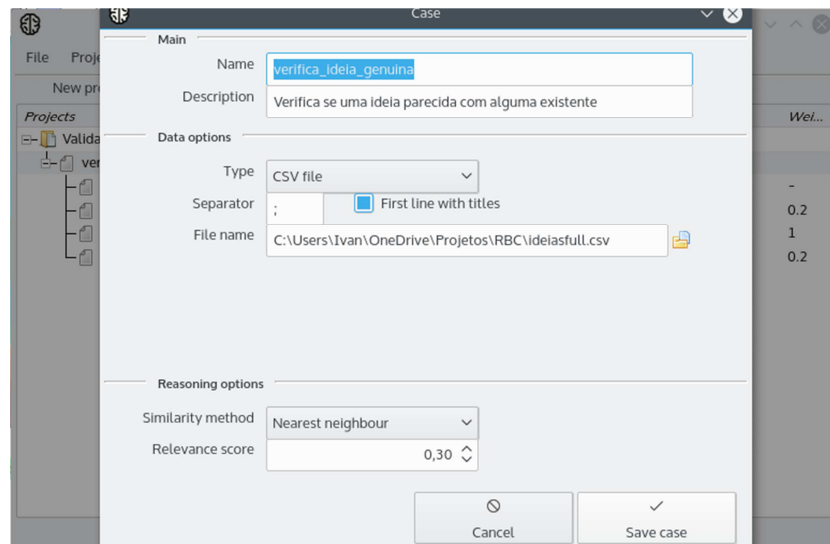
Assim como a disposição dos botões, este é um assunto que pode gerar longas discussões sem uma conclusão definitiva. Portanto, usa-se as mesmas configurações que o sistema operacional na qual o projeto está sendo executado.

Todas as figuras apresentadas neste trabalho foram feitas com a ferramenta em execução no Windows, com as fontes e cores que são o padrão deste sistema operacional. A seguir na Figura 21 e Figura 22 visualiza-se a aparência da ferramenta em execução no Linux com dois ambientes gráficos distintos, Gnome e KDE.



**Figura 21: Aparência da Ferramenta no Gnome**

**Fonte: Dos Autores (2015)**



**Figura 22: Aparência da Ferramenta no KDE**

**Fonte: Dos Autores (2015)**

#### 4.11. Ambiente de Desenvolvimento

O ambiente de desenvolvimento utilizado para a construção da ferramenta gráfica foi o LAZARUS, uma ferramenta para criação de aplicações visuais que usa a linguagem FREEPASCAL. Ela é orientada a objetos e compatível com DELPHI e permite compilar o mesmo código para diferentes plataformas e arquiteturas, como, Windows 32 ou 64 e Linux com bibliotecas QT que são a base do KDE ou Linux utilizando bibliotecas GTK que são a base do GNOME. Outro recurso útil é o suporte à internacionalização que será explicado na próxima subseção.

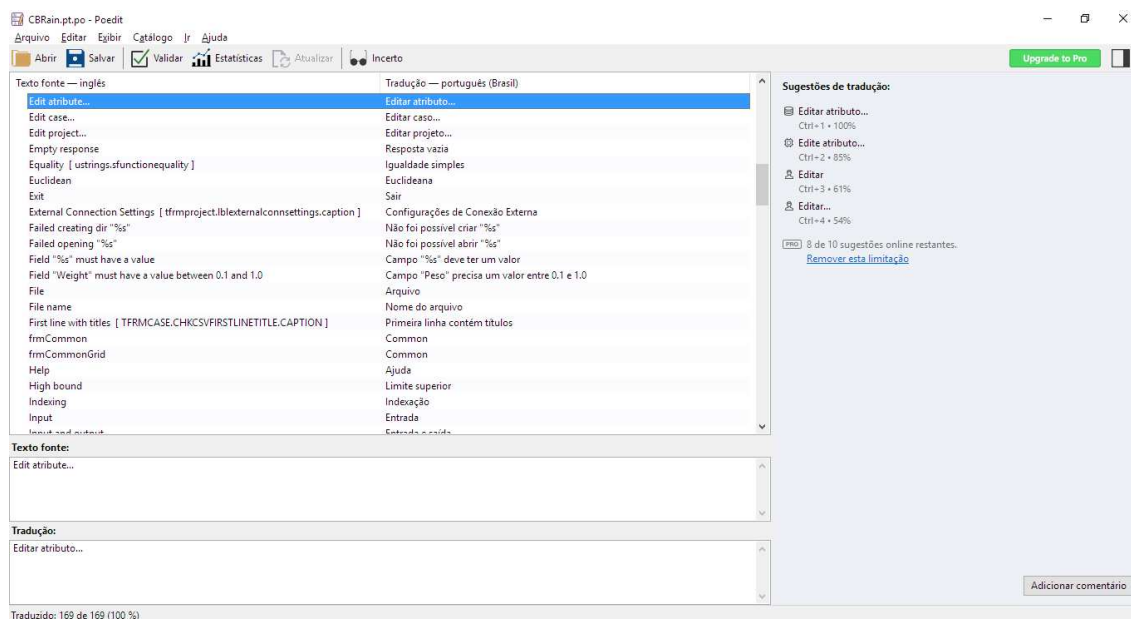
##### 4.11.1. Internacionalização

Após habilitar e configurar o suporte a internacionalização, todos os textos que foram usados durante o desenvolvimento são exportados, a cada compilação, para um arquivo chamado CBRain.po na pasta “languages”, um padrão internacionalmente conhecido para traduções em *software's*.

Os textos utilizados estão no idioma inglês, para traduzir para o idioma português utilizou-se a ferramenta POEDIT (<http://poedit.net>), que facilita e agiliza o processo de tradução, possui validador do arquivo e sugestões baseadas em uma base de conhecimento que ela vai construindo com a utilização. No final salvou-se a tradução com o nome CBRain.pt.po, assim ao executar será determinado o idioma da interface de acordo com o sistema operacional instalado.



A Figura 23 mostra a tela principal da POEDIT, com uma lista de todos os textos do projeto, abaixo dela, os campos de edição do texto selecionado e na direita o painel de sugestões.



**Figura 23: Tela Poedit**  
**Fonte: Dos Autores (2015)**

#### 4.12. Processo de Validação

A empresa Imply Tecnologia Eletrônica (Santa Cruz do Sul – RS) desenvolveu e utiliza um produto chamado 3i, um gerenciador de ideias que funciona da seguinte maneira: Um funcionário realiza cadastro; entra no sistema usando usuário e senha; clica no botão nova ideia e marca as categorias nas quais sua ideia se aplica, preenche título e descreve a ideia textualmente.

Em um segundo momento, um moderador analisa as novas ideias e libera para visualização dos demais. Depois disso, outras pessoas podem ler, apoiar, comentar e denunciar em caso de um abuso ou indicar se existe outra ideia similar. No final do mês, a melhor ideia é premiada pela empresa.

Para o processo de validação da ferramenta foram escolhidos dois participantes com conhecimentos na área de análise e desenvolvimento de sistema. A eles foi exibida uma apresentação com slides sobre a ferramenta e sobre RBC. Em seguida, foi entregue uma versão funcional da CBRAIN, o formulário de avaliação desenvolvido para este projeto que pode ser visualizado no ANEXO A, e um arquivo CSV com os dados exportados do gerenciador de ideias, previamente mencionado, e de um sistema de chamados internos da área de TI.

Foi solicitado que construíssem um novo projeto utilizando os dados CSV, deixando livres as decisões de quais parâmetros deveriam ser usados na configuração de casos e seus atributos. Ao final, deveriam responder as questões da avaliação.

Na primeira questão da avaliação, o primeiro participante avaliou bem a ferramenta, excetua-se apenas o item: mensagens e tratamentos de erros. Isto atribui-se ao fato dele ter encontrado inconsistências no tratamento de erros, que causaram eventuais interrupções da execução do sistema. Cita-se novamente este item na questão sobre dificuldades de uso, onde sugeriu-se a inclusão de um depurador no qual o usuário poderia visualizar como a ferramenta atingiu a pontuação apresentada.

Sugeriu a inclusão de mais funções de similaridade que sejam especializadas em textos, como por exemplo, utilização de sinônimos. Por fim, o participante definiu como uma ferramenta de fácil utilização e que atinge o objetivo principal de uma ferramenta RBC.

Com relação ao segundo participante percebe-se que os fatores de usabilidade aplicados à ferramenta foram eficientes e em um contexto geral a interface gráfica atende o objetivo proposto.

Documentação sobre o conhecimento específico em RBC deverá ser elaborada e agregada em forma de dicas durante a utilização da ferramenta. A ausência de informação sobre as funções de similaridade e como seus pesos influenciam os resultados, foi uma dificuldade identificada entre os participantes e poderá ocorrer com os futuros utilizadores da ferramenta CBRAIN.

Ambos os participantes utilizaram a ferramenta por aproximadamente um dia e as respostas de suas avaliações, apresentam-se no ANEXO B e no ANEXO C, mediante autorização deles.

## 5. CONCLUSÃO E TRABALHOS FUTUROS

Reafirmando as considerações da seção 3.7 do capítulo 3, o tema RBC é abrangente e existem estudos variados nesta área que está em constante evolução.

Este trabalho propôs uma ferramenta que utiliza tecnologias atuais para desenvolvimento de *software*; respeita as cinco premissas de usabilidade; possibilita integração com aplicações existentes de forma padronizada, portanto, o resultado esperado é que pesquisadores e profissionais possam usar esta ferramenta em seus estudos e aplicações que utilizam RBC. Mesmo que o foco deste trabalho tenha sido RBC, a usabilidade foi considerada como fator importante na construção da interface gráfica, em contraste das ferramentas deste estudo, que priorizaram apenas aspectos técnicos de RBC.

Esta ferramenta não implementa na íntegra todas as etapas do ciclo RBC nem todas as formas possíveis de cálculos de similaridade, pois como observou-se no estudo das ferramentas e material bibliográfico, não existe a forma única de executar o processo RBC e não existe uma lista fixa com as funções e cálculos para similaridade.

Como sugestão para trabalhos futuros cita-se: Implementação de novas funções de cálculo de similaridade; novos tipos de dados como data e hora; implementação de mais métodos de similaridade em textos longos como normalização, trigramas e sinônimos como citado por um dos participantes da avaliação; criação de um manual de utilização integrado com a interface gráfica, focado em auxiliar os usuários com as técnicas RBC e suas funções de similaridade. Sugere-se também a construção de projetos para as mais variadas aplicações, com novos recursos que poderão ser implementados, contribuindo para o aprimoramento da CBRAIN. Algumas aplicações que podem fazer uso de RBC são: Sistema de sugestões, suporte técnico, dúvidas frequentes, análise de risco e suporte a decisões.

Esta ferramenta pretende contribuir com a área de inteligência artificial, auxiliando profissionais e pesquisadores a incluir RBC em seus projetos de forma fácil e intuitiva, pois em contraste às ferramentas estudadas no capítulo 3, considerou usabilidade como fator fundamental na construção da interface gráfica. Disponibiliza todos os recursos encontrados nas ferramentas estudadas de forma melhorada com tecnologias atuais e adiciona um novo recurso: integração através da interface web, que possibilita ao usuário utilizar esta ferramenta com qualquer tipo de plataforma ou linguagem de programação.

## 6. REFERÊNCIAS

AMODT, A; PLAZA, C. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press, v. 7: n. 1, p. 39-59. 1994.

CBR Shell, disponível em: <http://www.aiai.ed.ac.uk/project/cbr>. Acessado em: maio de 2015.

CBR-Works, disponível em: <http://www.inf.ufsc.br/~patrec/cbr.html>. Acessado em: maio de 2015.

DELPHI. Develop Apps Windows, Mac OS X, iOS and Android, disponível em <https://www.embarcadero.com/br/products/delphi/>. Acessado em novembro de 2015.

Double Metaphone, disponível em: <http://swoodbridge.com/DoubleMetaPhone/>. Acessado em outubro de 2015.

ELKAFRAWY, P; MOHAMED, R. A. COMPARATIVE STUDY OF CASE BASED REASONING SOFTWARE, International Journal of Scientific Research and Management Studies (IJSRMS), v. 1: n. 6, p. 224-233, 2014.

FreeCBR, disponível em: <http://freecbr.sourceforge.net>. Acessado em: maio de 2015.

FREEPASCAL. Advanced open source Pascal compiler for Pascal and Object Pascal, disponível em <http://www.freepascal.org/>. Acessado em novembro de 2015.

jCOLIBRI, disponível em: <http://gaia.fdi.ucm.es/research/colibri/jcolibri>. Acessado em: maio de 2015.

KIPPER, L.; FROZZA, R.; URNAU, E. Modelagem de um Sistema para apoio à Tomada de Decisão com uso de Técnicas de Raciocínio baseado em Casos. Revista Tecno-Lógica, Santa Cruz do Sul, 2014

KOLODNER, Janet L. Case-Based Reasoning, California: Morgan Kaufmann Publishers, 1993

LAZARUS. The professional Free Pascal RAD IDE, disponível em: <http://www.lazarus-ide.org>, com documentação disponível em: [http://wiki.freepascal.org/Lazarus\\_Documentation](http://wiki.freepascal.org/Lazarus_Documentation). Acessado em: junho de 2015.

Metaphone para a língua portuguesa, disponível em: <http://informatica.varzeapaulista.sp.gov.br/metaphone>. Acessado em: outubro de 2015.

myCBR, disponível em: <http://mycbr-project.net>. Acessado em: maio de 2015.

NIELSEN, J. OK-Cancel or Cancel-OK?, Nielsen Norman Group: 2008.

NIELSEN, J. Usability Engineering, Morgan Kaufmann: San Francisco. 1993.

PHILIPS, L. Hanging on the Metaphone, Computer Language, Vol. 7, No. 12 (December), 1990.

PHP. Hypertext Preprocessor, disponível em: <http://php.net/>, com documentação disponível em: <http://php.net/docs.php>. Acessado em: junho de 2015.

POEDIT. Gettext Translations Editor, disponível em: <http://poedit.net>. Acessado em: junho de 2015.

POSSELT, E. L. INFUZZY – Ferramenta para Desenvolvimento de Aplicações de Sistemas Difusos. UNISC: Santa Cruz do Sul, 2011. (Dissertação de Mestrado – PPGSPI).

SANTOS, C. F. H dos; FROZZA, R.; MOLZ, K. W. Técnica de Raciocínio Baseado em Casos Aplicado ao Gerenciamento de Processos de Negócio, UNISC: Santa Cruz do Sul, 2013.

TARDELLI, A. O. Identificação de Artigos relacionados e citações na Coleção Scielo de Revistas Eletrônicas através de Algoritmo de Similaridade de Textos por Trigramas. Universidade Federal de São Paulo: São Paulo, 2008.

WANGENHEIM, C. G.; WANGENHEIM, A. Raciocínio Baseado em Casos. 1. ed. Barueri, SP: Manole, 2003.

## ANEXO A - Avaliação do Sistema CBRAIN

Prezado participante.

Esta avaliação tem como objetivo validar CBRAIN - Ferramenta de Raciocínio Baseado em Casos, coletando informações sobre a sua opinião com relação à usabilidade. As informações fornecidas auxiliarão no processo de aprimoramento da ferramenta. Deseja-se que você possua conhecimentos na área de desenvolvimento de sistemas e noções de inteligência artificial.

Propõe-se que sejam realizados os passos a seguir:

- Acessar o endereço <http://hennig.net.br/gii/cbrain>; na área de *downloads* escolher a opção que seja compatível com o sistema operacional onde a ferramenta será instalada.
- Executar o instalador, seguindo os passos indicados.
- Executar a ferramenta CBRAIN.
- Criar um novo projeto e configurar para não utilizar banco de dados remoto. Um arquivo CSV com dados será fornecido diretamente ao participante.
- No projeto, criar um novo caso e configurar para utilizar o CSV e parâmetros globais da similaridade.
- No caso criado, criar os atributos que se relacionam com cada uma das colunas do CSV, indicar quais atributos serão usados para pesquisa (atributos de entrada), e quais serão usados como resultado (atributos de saída). Ainda em atributos, escolher diferentes configurações de similaridade, pois cada tipo de dados contém diferentes formas de parametrização.
- Abrir o formulário de entrada do caso e testar os resultados obtidos com as configurações. Ao encontrar resultados similares, estes serão exibidos em uma listagem com a pontuação obtida.

Após instalação e utilização do sistema, responda o questionário a seguir:

1. Marque a opção correspondente a cada critério:

Facilidade de utilização	( ) Ruim ( ) Regular ( ) Bom
Organização e disposição dos elementos	( ) Ruim ( ) Regular ( ) Bom
Visual dos formulários	( ) Ruim ( ) Regular ( ) Bom
Nomenclatura dos elementos	( ) Ruim ( ) Regular ( ) Bom
Mensagens e tratamento de erros	( ) Ruim ( ) Regular ( ) Bom
Assimilação do uso da ferramenta	( ) Ruim ( ) Regular ( ) Bom
Classifique de um modo geral	( ) Ruim ( ) Regular ( ) Bom

2. Descreva as facilidades observadas ao utilizar a ferramenta:

---

---

---

---

---

---

3. Descreva as dificuldades observadas ao utilizar a ferramenta:

---

---

---

---

---

---

4. Você acredita que a ferramenta atingiu o objetivo principal:

---

---

---

---

---

---

5. Espaço livre para expor sua opinião/sugestões/críticas sobre a ferramenta:

---

---

---

---

---

---

### ANEXO B - Resposta Avaliação Ederson

Prezado participante.

Esta avaliação tem como objetivo validar CBRAIN - Ferramenta de Raciocínio Baseado em Casos, coletando informações sobre a sua opinião com relação à usabilidade. As informações fornecidas auxiliarão no processo de aprimoramento da ferramenta. Deseja-se que você possua conhecimentos na área de desenvolvimento de sistemas e noções de inteligência artificial.

Propõe-se que sejam realizados os passos a seguir:

- Acessar o endereço <http://hennig.net.br/gii/cbrain>; na área de *downloads* escolher a opção que seja compatível com o sistema operacional onde a ferramenta será instalada.
- Executar o instalador, seguindo os passos indicados.
- Executar a ferramenta CBRAIN.
- Criar um novo projeto e configurar para não utilizar banco de dados remoto. Um arquivo CSV com dados será fornecido diretamente ao participante.
- No projeto, criar um novo caso e configurar para utilizar o CSV e parâmetros globais da similaridade.
- No caso criado, criar os atributos que se relacionam com cada uma das colunas do CSV, indicar quais atributos serão usados para pesquisa (atributos de entrada), e quais serão usados como resultado (atributos de saída). Ainda em atributos, escolher diferentes configurações de similaridade, pois cada tipo de dados contém diferentes formas de parametrização.
- Abrir o formulário de entrada do caso e testar os resultados obtidos com as configurações. Ao encontrar resultados similares, estes serão exibidos em uma listagem com a pontuação obtida.

Após instalação e utilização do sistema, responda o questionário a seguir:

1. Marque a opção correspondente a cada critério:

Facilidade de utilização	<input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input type="checkbox"/> Bom <input checked="" type="checkbox"/>
Organização e disposição dos elementos	<input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input checked="" type="checkbox"/> Bom
Visual dos formulários	<input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input checked="" type="checkbox"/> Bom
Nomenclatura dos elementos	<input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input checked="" type="checkbox"/> Bom
Mensagens e tratamento de erros	<input type="checkbox"/> Ruim <input checked="" type="checkbox"/> Regular <input type="checkbox"/> Bom
Assimilação do uso da ferramenta	<input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input type="checkbox"/> Bom <input checked="" type="checkbox"/>
Classifique de um modo geral	<input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input checked="" type="checkbox"/> Bom



2. Descreva as facilidades observadas ao utilizar a ferramenta:

Apresenta boa sequencialidade nas etapas até atingir as pesquisas.

3. Descreva as dificuldades observadas ao utilizar a ferramenta:

Não apresenta uma depuração para explicar o resultado apresentado ao final da pesquisa. Deveria existir um elemento que mostrasse todo o processo envolvido até a solução. Poderia ser um simples campo TEXT listando o debug, a medida em que as funções são chamadas, as informações de debug vão se persistindo.

A visualização deveria ocorrer através de acionamento do usuário caso seja necessário, um clique na linha de resultado.

4. Você acredita que a ferramenta atingiu o objetivo principal:

A ferramenta atingiu o objeto principal, e isso pode ser expressado através do fato de ser possível construir projetos RBC de forma dinâmica com diferentes bases de conhecimento, através de um mapeamento de atributos de entrada e saída.

5. Espaço livre para expor sua opinião/sugestões/críticas sobre a ferramenta:

Provavelmente deverão ocorrer trabalhos futuros nos quais devam ser agregadas novas funcionalidades sobre a dinâmica de processamento de texto. Disponibilizar novos algoritmos para essa finalidade.

### ANEXO C - Resposta Avaliação Katieli

Prezado participante.

Esta avaliação tem como objetivo validar CBRAIN - Ferramenta de Raciocínio Baseado em Casos, coletando informações sobre a sua opinião com relação à usabilidade. As informações fornecidas auxiliarão no processo de aprimoramento da ferramenta. Deseja-se que você possua conhecimentos na área de desenvolvimento de sistemas e noções de inteligência artificial.

Propõe-se que sejam realizados os passos a seguir:

- Acessar o endereço <http://hennig.net.br/gii/cbrain>; na área de *downloads* escolher a opção que seja compatível com o sistema operacional onde a ferramenta será instalada.
- Executar o instalador, seguindo os passos indicados.
- Executar a ferramenta CBRAIN.
- Criar um novo projeto e configurar para não utilizar banco de dados remoto. Um arquivo CSV com dados será fornecido diretamente ao participante.
- No projeto, criar um novo caso e configurar para utilizar o CSV e parâmetros globais da similaridade.
- No caso criado, criar os atributos que se relacionam com cada uma das colunas do CSV, indicar quais atributos serão usados para pesquisa (atributos de entrada), e quais serão usados como resultado (atributos de saída). Ainda em atributos, escolher diferentes configurações de similaridade, pois cada tipo de dados contém diferentes formas de parametrização.
- Abrir o formulário de entrada do caso e testar os resultados obtidos com as configurações. Ao encontrar resultados similares, estes serão exibidos em uma listagem com a pontuação obtida.

Após instalação e utilização do sistema, responda o questionário a seguir:

1. Marque a opção correspondente a cada critério:

Facilidade de utilização	( ) Ruim ( ) Regular ( x ) Bom
Organização e disposição dos elementos	( ) Ruim ( ) Regular ( x ) Bom
Visual dos formulários	( ) Ruim ( ) Regular ( x ) Bom
Nomenclatura dos elementos	( ) Ruim ( ) Regular ( x ) Bom
Mensagens e tratamento de erros	( ) Ruim ( ) Regular ( x ) Bom
Assimilação do uso da ferramenta	( ) Ruim ( ) Regular ( x ) Bom
Classifique de um modo geral	( ) Ruim ( ) Regular ( x ) Bom

2. Descreva as facilidades observadas ao utilizar a ferramenta:

As interfaces do sistema são limpas, não possuem elementos que possam desviar a atenção do usuário, e possibilitam um fácil entendimento do uso esperado assim como das respostas geradas. Todas as opções estão acessíveis no menu principal como na barra acessos rápidos (onde são liberadas conforme as opções são usadas) o que facilita e torna o uso ainda mais ágil.

3. Descreva as dificuldades observadas ao utilizar a ferramenta:

Dificuldade com relação a identificação dos valores esperados nos campos do tipo numéricos, como: peso do atributo e na inclusão da tabela de similaridade no caso de atributo do tipo Símbolo não ordenado, mas a mesma foi sanada após alguns testes.

4. Você acredita que a ferramenta atingiu o objetivo principal:

Acredito que sim, a ferramenta atingiu o objetivo, tendo em vista que nos diversos testes que realizei a ferramenta apresentou os resultados esperados de forma entendível, além de ter retornado dados relevantes ou dados vazios quando deveria.

5. Espaço livre para expor sua opinião/sugestões/críticas sobre a ferramenta:

No meu ponto de vista a ferramenta ficou exemplar e possui muita aplicabilidade e um potencial de uso incríveis. Vindo a facilitar o trabalho, por exemplo de equipes de TI, se usada corretamente.