

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**RECRUTA-IA: APLICAÇÃO PARA ANÁLISE DE**  
**CURRÍCULOS PARA VAGAS UTILIZANDO RACIOCÍNIO**  
**BASEADO EM CASOS**

**CARLOS ALBERTO CURBANI**

**BLUMENAU**  
**2017**

**CARLOS ALBERTO CURBANI**

**RECRUTA-IA: APLICAÇÃO PARA ANÁLISE DE  
CURRÍCULOS PARA VAGAS UTILIZANDO RACIOCÍNIO  
BASEADO EM CASOS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Francisco Adell Péricas- Orientador

**BLUMENAU  
2017**

**RECRUTA-IA: APLICAÇÃO PARA ANÁLISE DE  
CURRÍCULOS PARA VAGAS UTILIZANDO RACIOCÍNIO  
BASEADO EM CASOS**

Por

**CARLOS ALBERTO CURBANI**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Francisco Adell Péricas, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Cláudio Ratke, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Roberto Heinzle, Doutor – FURB

Blumenau, 11 de dezembro de 2017

Dedico este trabalho à minha família, meus amigos, meus professores e a todos que contribuíram para a realização deste trabalho e de toda minha formação acadêmica.

## **AGRADECIMENTOS**

A Deus por dar a oportunidade e a força para realizar este trabalho.

À minha família por sempre estar presente e me apoiarem durante minha trajetória.

Aos meus amigos que sempre ajudaram quando necessário.

Ao meu orientador Francisco Adell Péricas e a professora Andreza Sartori, que sempre me auxiliaram e estiveram presente na realização deste trabalho.

## RESUMO

Este trabalho apresenta uma aplicação Web para auxiliar o profissional de Recursos Humanos no processo de recrutamento e seleção de novos funcionários. A aplicação possibilita o cadastro de vagas em aberto e de candidatos disponíveis para contratação, informando suas experiências profissionais, formações, endereço, entre outros dados, além da busca de informações do candidato no seu perfil do LinkedIn. Na tomada de decisão para o recrutamento, a aplicação auxilia o profissional de Recursos Humanos executando um cálculo de similaridade entre as vagas e os candidatos disponíveis, utilizando técnicas de inteligência artificial para retornar os candidatos que mais se assemelham aos atributos da vaga em aberto. No desenvolvimento da interface para o usuário, utilizou-se o *framework* Primefaces, junto com HTML5, JavaScript e JSF para criação das páginas, utilizando JQuery para comunicação com a API do LinkedIn. Na parte do servidor foi utilizada a linguagem Java e o servidor TomCat 8, juntamente com o banco de dados MySQL para salvar os dados. O cálculo de similaridade foi desenvolvido com uso de técnicas de Raciocínio Baseado em Casos, utilizando vagas e candidatos já vinculados anteriormente, utilizando o conhecimento utilizado nas soluções anteriores para solucionar uma nova vaga disponível. Como resultado obteve-se uma aplicação Web para agilizar e facilitar no processo de recrutamento e seleção de candidatos executado pelo profissional de Recursos Humanos.

Palavras-chave: Inteligência artificial. Recursos humanos. Recrutamento. Seleção. LinkedIn.

## **ABSTRACT**

This project presents a Web application to assist the Human Resources professional in the process of recruiting and of selecting new employees. The application enables the registration of open positions and candidates available for hiring, informing their professional experiences, training, address, among other data, in addition to the candidate's search of information in their LinkedIn profile. In decision-making for recruitment, the application assists the Human Resources professional by performing a similarity calculation between available positions and candidates using artificial intelligence techniques to return candidates who most closely resemble the job attributes. In the development of the user interface, we used the framework Primefaces, together with HTML5, JavaScript and JSF for page authoring, using JQuery to communicate with the LinkedIn API. On the server side was used the Java language and the TomCat 8 server, along with the MySQL database to save the data. The calculation of similarity was developed using case-based reasoning techniques, using vacancies and candidates previously linked, taking advantage of the knowledge used in previous solutions to solve the current available vacancy. As a result, we obtained a web application to streamline the recruitment and selection of candidates carried out by the Human Resources professional.

**Key-words:** Artificial intelligence. Human resources. Recruitment. Selection. LinkedIn.

## LISTA DE FIGURAS

Figura 1 - Seleção de pessoal como uma comparação.....	15
Figura 2 - Estágios de evolução da aprendizagem.....	17
Figura 3 - Estágios do processamento do conhecimento .....	17
Figura 4 - Áreas relacionadas com IA .....	18
Figura 5 - Exemplo simplificado de um caso .....	19
Figura 6 - Ciclo do RBC .....	20
Figura 7 - Função de Similaridade.....	21
Figura 8 - Tela de cadastro do candidato .....	23
Figura 9 - Lista de vagas cadastradas .....	24
Figura 10 – Lista de vagas disponíveis .....	24
Figura 11 - Tela de busca de currículos .....	26
Figura 12 - Diagrama de casos de uso .....	28
Figura 13 - Diagrama de classes do candidato .....	29
Figura 14 - Diagrama de classes da vaga .....	30
Figura 15 - Diagrama de classes do RBC .....	31
Figura 16 - Diagrama de atividades .....	32
Figura 17 - Principais tabela para similaridade .....	33
Figura 18 - Modelo Entidade Relacionamento.....	34
Figura 19 - Página de autenticação do aplicativo .....	35
Figura 20 - Resposta ao pedido de parceria .....	37
Figura 21 - Formula para cálculo da similaridade .....	43
Figura 22 - Página de <i>login</i> .....	44
Figura 23 - Página para o candidato cadastrar seu currículo.....	45
Figura 24 - Página de candidatos.....	45
Figura 25 - Página de vagas .....	46
Figura 26 - Página de experiências do candidato .....	46
Figura 27 - Página de atributos para similaridade .....	47
Figura 28 - Página de parâmetros .....	48
Figura 29 - Página de similaridade .....	48
Figura 30 - Detalhes da pontuação .....	49
Figura 31 - Página de soluções .....	49



Figura 32 - Resultado do uso dos casos passados..... 50

## LISTA DE QUADROS

Quadro 1 - Campos acessados com a permissão básica.....	36
Quadro 2 - Método para implementar a etapa recuperar do RBC .....	38
Quadro 3 - Método para popular os valores aceitos .....	39
Quadro 4 - Uso das Anotações em classes .....	40
Quadro 5 – Código para buscar o método referente ao atributo.....	40
Quadro 6 - Método para retornar a pontuação da similaridade entre dois objetos .....	41
Quadro 7 - Função para recuperar dados do perfil .....	43
Quadro 8 - Características dos trabalhos correlatos e do trabalho proposto .....	52
Quadro 9 - Caso de uso: Cadastrar candidatos.....	55
Quadro 10 - Caso de uso: Cadastrar vaga.....	55
Quadro 11 – Caso de uso: Configurar o peso dos atributos.....	55
Quadro 12 – Caso de uso: Executar a similaridade entre vagas e candidatos.....	56
Quadro 13 - Caso de uso: Consultar o relatório da similaridade entre candidato e vaga.....	56
Quadro 14 – Caso de uso: Vincular a vaga com o candidato.....	56
Quadro 15 – Caso de uso: Executar o próprio cadastro.....	57
Quadro 16 – Caso de uso: Buscar informações no LinkedIn para o cadastro.....	57
Quadro 17 - Dicionário da tabela candidato.....	58
Quadro 18 - Dicionário da tabela vaga .....	58
Quadro 19 - Dicionário da tabela solucao .....	58
Quadro 20 - Dicionário da tabela parametro.....	58
Quadro 21 - Dicionário da tabela atributo .....	59

## **LISTA DE ABREVIATURAS E SIGLAS**

API - Application Programming Interface

ARH – Administrador de Recursos Humanos

IA – Inteligência Artificial

JSF - JavaServer Faces

MER – Modelo Entidade Relacionamento

RBC – Raciocínio Baseado em Casos

RF – Requisito Funcional

RH - Recursos Humanos

RNF – Requisito Não Funcional

UC – Use Case

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 OBJETIVOS .....	13
1.2 ESTRUTURA .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>14</b>
2.1 RECRUTAMENTO E SELEÇÃO .....	14
2.2 INTELIGÊNCIA ARTIFICIAL E SUA APLICAÇÃO .....	16
2.3 RACIOCÍNIO BASEADO EM CASOS.....	18
2.4 TRABALHOS CORRELATOS .....	22
2.4.1 Sistema RBC para recrutamento e seleção de profissionais .....	22
2.4.2 Desenvolvimento de sistema online para recrutamento de pessoas .....	23
2.4.3 SIIEE – Sistema de raciocínio baseado em casos para análise de perfil de candidatos a vagas de estágio .....	25
<b>3 DESENVOLVIMENTO</b> .....	<b>27</b>
3.1 REQUISITOS .....	27
3.2 ESPECIFICAÇÃO .....	27
3.2.1 Diagrama de casos de uso .....	28
3.2.2 Diagrama de classes .....	28
3.2.3 Diagrama de atividade .....	31
3.2.4 Modelo Entidade Relacionamento.....	33
3.2.5 Comunicação com LinkedIn.....	35
3.3 IMPLEMENTAÇÃO .....	37
3.3.1 Técnicas e ferramentas utilizadas .....	37
3.3.2 Operacionalidade da implementação .....	44
3.4 ANÁLISE DOS RESULTADOS .....	50
<b>4 CONCLUSÕES</b> .....	<b>53</b>
4.1 EXTENSÕES .....	53
<b>REFERÊNCIAS</b> .....	<b>54</b>
<b>APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO</b> .....	<b>55</b>
<b>APÊNDICE B – DICIONÁRIO DE DADOS</b> .....	<b>58</b>

## 1 INTRODUÇÃO

Atualmente a concorrência por vagas de emprego vem aumentando consideravelmente. Segundo notícia publicada no portal G1, em 2016 a concorrência entre os candidatos para uma vaga foi a maior dos últimos sete anos (G1, 2016). Este crescimento pela procura por emprego intensifica a importância do trabalho de Recursos Humanos (RH) de empresas. Segundo Ribeiro (2005), a área de Recursos Humanos tem como objetivo principal administrar as relações da organização com as pessoas.

Segundo Pontelo e Cruz (2012, p. 64), “o recrutamento de pessoal pode ser definido como o método que uma organização utiliza quando tem necessidade de preencher um cargo, ou seja, a forma pela qual atrai e cadastra candidatos que possam preencher os requisitos da vaga”. Segundo Ribeiro (2005), fica a cargo do profissional de Recursos Humanos prover a organização de meios para avaliar as pessoas em todos os sentidos, desde a sua formação profissional até a coleta de informações que possam lhe dar a indicação sobre seu caráter.

Contudo, essa tarefa pode se tornar demorada dependendo da quantidade de currículos recebidos pela empresa, fazendo com que muitas vezes, seja necessário o uso de algum sistema ou tecnologia que auxilie nesse processo. Segundo Chiavenato (2002), a internet proporciona velocidade de informação e facilitou trabalhar com grandes volumes de dados. Para o cadastro dos currículos, quando a empresa possui um site, os candidatos que acessam podem fazer o próprio cadastro digitando os dados solicitados. Nesses casos o administrador de RH se ocupa somente da análise e decisão sobre o candidato que atende os requisitos da vaga disponível, o que pode ser auxiliado com o uso de Inteligência Artificial (IA).

Segundo Wangenheim e Wangenheim (2003), a IA auxilia na tomada de decisão. Em especial o Raciocínio Baseado em Casos (RBC), uma das técnicas mais populares para Sistemas Baseados em Conhecimento, usa uma abordagem para solução de problemas e para aprendizado com base em experiência passada. O RBC utiliza uma base de conhecimento, que contém os casos anteriores já cadastrados, para serem usados em comparação ao caso atual, para o qual se procura uma solução com base nas experiências anteriores. Para chegar ao resultado é aplicada uma técnica de similaridade entre os casos, podendo ser utilizado para auxiliar em tarefas de comparação de características (WANGENHEIM; WANGENHEIM, 2003).

Diante desse contexto, este trabalho propõe desenvolver uma aplicação web para auxiliar no recrutamento de pessoas, através do uso de técnicas de RBC para análise de

candidatos cadastrados. Assim, busca-se verificar a similaridade com as vagas disponíveis na empresa, para indicar os candidatos mais adequados para os requisitos da vaga.

## 1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um ambiente para auxiliar na análise dos currículos recebidos por uma empresa para o preenchimento de vagas utilizando técnicas de Raciocínio Baseado em Casos (RBC).

Os objetivos específicos são:

- a) disponibilizar uma interface Web para cadastro das vagas e de currículos recebidos;
- b) desenvolver uma integração com o LinkedIn para buscar as informações do candidato cadastrado;
- c) possibilitar definir o valor do peso dos atributos utilizados no cálculo da similaridade entre a vaga e os currículos cadastrados.

## 1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta a introdução do trabalho e os objetivos. O segundo capítulo apresenta a fundamentação teórica sobre recrutamento e seleção, inteligência artificial e sua aplicação, raciocínio baseado em casos e os trabalhos correlatos. No terceiro capítulo é demonstrado o desenvolvimento do trabalho com requisitos, especificação, diagramas, implementação, resultados e operacionalidade da aplicação. Por fim, no quarto capítulo são relatadas as conclusões e também as possíveis extensões.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em três seções. A Seção 2.1 aborda o processo de recrutamento e seleção aplicado pelo RH para contratação de novos funcionários. A Seção 2.2 apresenta a definição de IA e sua aplicação. Na Seção 2.3 são apresentadas as etapas do RBC e seu uso. Por fim, na Seção 2.4 são apresentados três trabalhos correlatos.

### 2.1 RECRUTAMENTO E SELEÇÃO

O recrutamento é o conjunto de técnicas e procedimentos que visa atrair candidatos potencialmente qualificados e capazes de ocupar cargos dentro da organização. É basicamente um sistema de informação, através do qual a organização divulga e oferece ao mercado de recursos humanos oportunidades de emprego que pretende preencher (CHIAVENATO, 2002).

Segundo Pontelo e Cruz (2012), para planejar e executar o recrutamento são importantes as seguintes etapas:

- a) levantamento das necessidades para contratação de novos funcionários para a empresa;
- b) análise das atividades e funções referentes ao cargo;
- c) identificação do perfil adequado para ocupar a vaga;
- d) determinação do método de seleção dos candidatos e os critérios de aprovação;
- e) identificação da região onde se pretende fazer o recrutamento;
- f) definição dos veículos de divulgação.

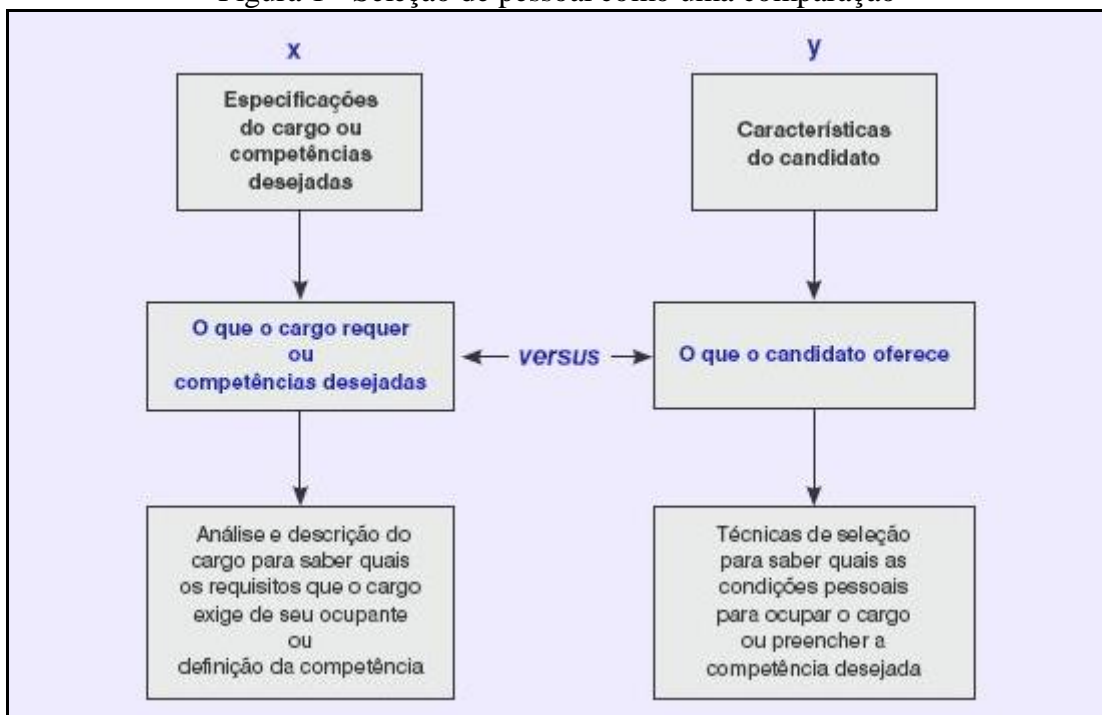
Após a definição da vaga, deve-se verificar quais candidatos podem ocupar as vagas abertas. Segundo Ribeiro (2005), se a empresa não possui funcionários aptos a preencher as vagas disponíveis, o cadastro de currículos é a primeira forma de recrutamento externo que se deve procurar, onde os currículos de bons candidatos que não forem aproveitados inicialmente devem ser mantidos para servir de fonte de consulta. Para Ribeiro (2005), é fundamental que a empresa se apresente sempre disponível para receber candidatos e efetuar o cadastro de seus currículos, para não perder a atratividade no mercado de trabalho e não desestimular os candidatos que a procuram.

Para Chiavenato (2002), a seleção de pessoas faz parte do processo de provisão de pessoal, vindo depois do recrutamento, ambos fazem parte do mesmo processo, referente a introdução de recursos humanos na empresa. A seleção tem como finalidade escolher o candidato certo para o cargo certo. Em função disso, torna-se de vital importância saber

selecionar corretamente os candidatos que irão representar a empresa nos cargos disponíveis, contribuindo de maneira eficaz para o crescimento da empresa (PONTELO; CRUZ, 2012).

A seleção é um processo de comparação entre duas variáveis, tendo de um lado os requisitos da vaga a ser preenchida e do outro lado o perfil das características dos candidatos que se apresentam. A primeira variável é fornecida pela descrição e análise do cargo, enquanto a segunda é obtida por meio de aplicações das técnicas de seleção, (CHIAVENATO, 2002). Na Figura 1 é apresentada a comparação entre as variáveis X (vaga) e Y (candidato) durante o processo de seleção.

Figura 1 - Seleção de pessoal como uma comparação



Fonte: Chiavenato (2002).

Conforme Chiavenato (2002):

Quando X é maior do que Y, dizemos que o candidato não atinge as condições ideais para ocupar determinado cargo e, portanto, é rejeitado para aquele cargo. Quando X e Y são iguais, dizemos que o candidato reúne as condições ideais para tanto e, portanto, é aprovado. Quando a variável Y for maior do que X, o candidato reúne mais do que as condições exigidas pelo cargo e, portanto, torna-se superdotado para aquele cargo. Na verdade, essa comparação não se concentra em um único ponto de igualdade entre as variáveis, mas sobretudo em uma faixa de aceitação, admitindo certa flexibilidade a mais ou a menos ao redor do ponto ideal (CHIAVENATO, 2002, p. 225).

Chiavenato (2002) defende que existe tolerância entre aos limites admitidos no processo de controle de igualdade. Uma vez feita a comparação entre as características exigidas pelo cargo e as oferecidas pelos candidatos, pode acontecer que vários candidatos se aproximem das exigências solicitadas para vaga e nesse caso fica a cargo do requerente selecionar qual candidato deve preencher a vaga.



Entre as formas de recrutamento e seleção, a Web é uma excelente alternativa para buscar novos candidatos. Segundo Chiavenato (2002), um site permite que a empresa esteja sempre acessível, tornando-se um importante meio de contato entre organizações e candidatos, proporcionando velocidade de informação, agilidade e economia, onde o candidato pode concorrer a várias oportunidades no mercado, mesmo estando em casa. Os serviços on-line de recrutamento abrem as portas de muitas empresas e permitem eliminar etapas da provisão, além de permitir a montagem de bancos de talentos, ao se guardam currículos de candidatos (CHIAVENATO, 2002).

Para Ribeiro (2005), o cadastro do candidato pela internet pode facilitar no processo de recrutamento, mas deve se tomar cuidado, pois isso pode implicar em centenas ou milhares de currículos para serem analisados. Nesse ponto pode ser fundamental a utilização de um software para auxiliar no processo, sendo que o mesmo pode utilizar técnicas de inteligência artificial para executar a tarefa, auxiliando na tomada de decisão.

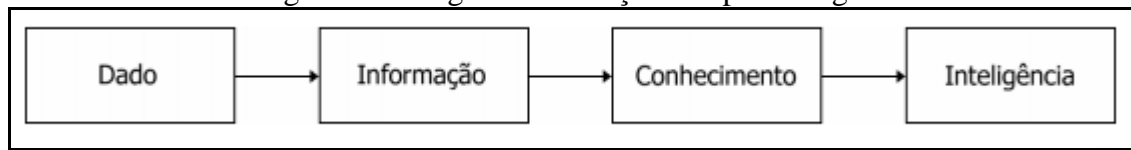
## 2.2 INTELIGÊNCIA ARTIFICIAL E SUA APLICAÇÃO

A Inteligência Artificial (IA) é um dos campos mais recentes em ciências e engenharia, tendo início logo após a Segunda Guerra Mundial. Atualmente ela abrange uma enorme variedade de campos, desde aprendizado e percepção até tarefas específicas, como jogos de xadrez, demonstração de teoremas matemáticos, diagnóstico de doenças, entre outras, sendo relevante para qualquer tarefa intelectual (RUSSEL; NORVIG, 2004).

A IA é um ramo cujo interesse é fazer os computadores pensar ou se comportar de forma inteligente. Existem quatro linhas de pensamentos que tentam definir inteligência, sendo duas relacionadas ao pensamento e raciocínio e duas relacionadas ao comportamento, onde uma das linhas do pensamento e comportamento se relaciona à fidelidade do desempenho humano, enquanto a outra ao conceito ideal de inteligência (RUSSEL; NORVIG, 2004).

Na primeira parte da história da IA, o aprendizado da máquina, assim como processos de aquisição de conhecimento, era realizado diretamente pelo ser humano. Posteriormente tornou-se possível ser realizado através de processos automatizados de extração de conhecimento (MONARD; BARANAUKAS, 2000). Segundo Santos (2008), o conhecimento pode ser interpretado como a informação baseada em experiências, habilidades e competências, já a inteligência é adquirida através das transformações das experiências e da aquisição de novos conhecimentos. A Figura 2 mostra os estágios da evolução da aprendizagem.

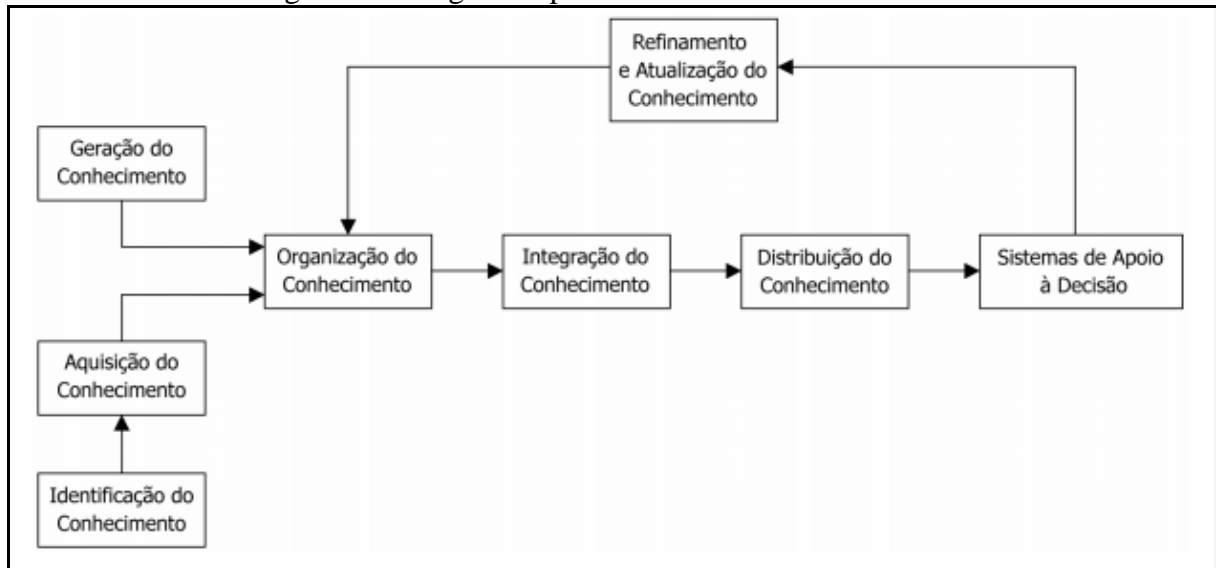
Figura 2 - Estágios de evolução da aprendizagem



Fonte: Santos (2008).

O conhecimento pode ser gerado por ferramentas de mineração de dados, adquirido de terceiros, através de sistemas gerenciadores de conteúdo, ou ser refinado e atualizado a partir de uma base de conhecimento. Esse conhecimento pode ser organizado e utilizado por aplicações de suporte à tomada de decisões, tendo como resultado aplicações que auxiliam na tomada de decisão e refinam o conhecimento já existente para apresentar o conhecimento solicitado (SANTOS, 2008). A Figura 3 apresenta os estágios do processamento do conhecimento.

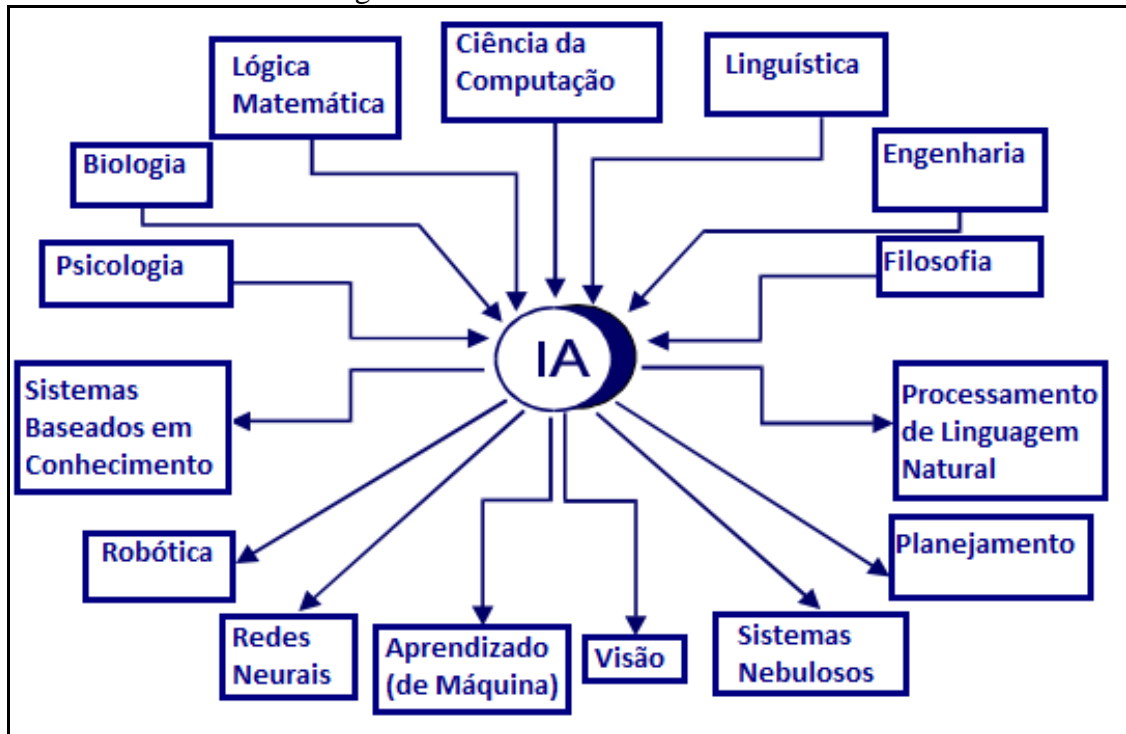
Figura 3 - Estágios do processamento do conhecimento



Fonte: Santos (2008).

Segundo Monard e Baranaukas (2000), devido a IA ser uma área muito ampla, ela também está relacionada com psicologia, biologia, lógica matemática, linguística, engenharia, filosofia, entre outras áreas, desenvolvendo várias linhas de pesquisa, onde tem como objetivo fornecer ao computador habilidades para efetuar funções antes desempenhadas apenas através da inteligência humana. Na Figura 4 está representado as áreas relacionadas a IA e suas linhas de pesquisa, geradas a partir desta relação.

Figura 4 - Áreas relacionadas com IA



Fonte: adaptado de Monard e Baranaukas (2000).

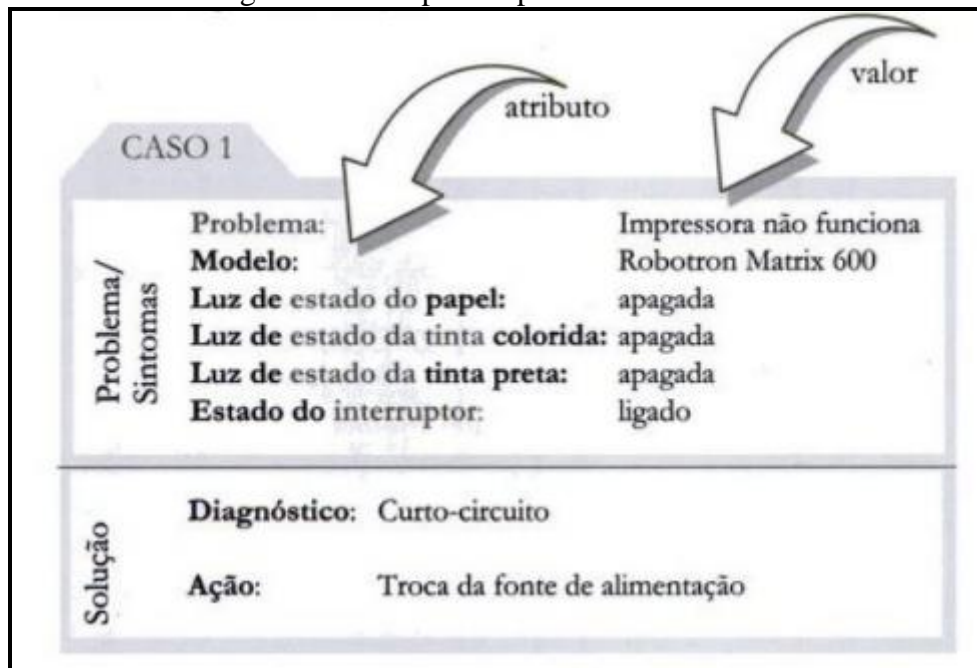
Existem algumas áreas dentro da inteligência artificial que auxiliam no processo de aprendizado de máquina e tomada de decisão, algumas dessas áreas são os Sistemas Especialistas, as Redes Neurais e os Sistemas Baseados em Conhecimento. A seguir será apresentada a definição e funcionamento da técnica Raciocínio Baseado em Casos (RBC), umas das técnicas de IA para Sistemas Baseados em Conhecimento.

### 2.3 RACIOCÍNIO BASEADO EM CASOS

O Raciocínio Baseado em Casos (RBC) surgiu como técnica poderosa para solução automática de problemas tendo como ideia básica a resolução de um novo problema relembrando uma situação anterior similar, bem como reutilizar a informação e conhecimento daquela situação (WANGENHEIM; WANGENHEIM, 2003).

Para Wangenheim e Wangenheim (2003), um caso representa a descrição de uma situação (problema), juntamente com as experiências adquiridas (solução) durante a sua resolução, sendo organizados e armazenados em uma base de casos. A Figura 5 apresenta um exemplo simplificado de um caso, onde representa um caso referente ao conserto de impressoras, descrevendo o problema e a respectiva solução do caso, que serão utilizadas durante o cálculo da similaridade para resolver o problema de um novo caso.

Figura 5 - Exemplo simplificado de um caso



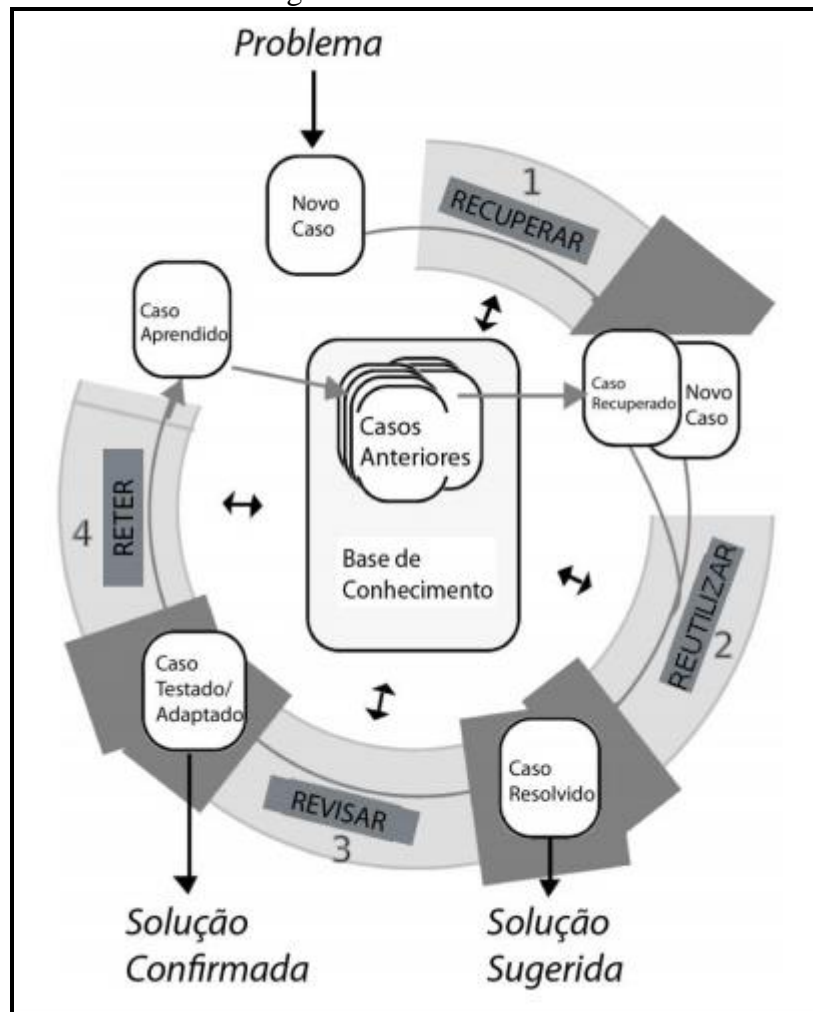
Fonte: Wangenheim e Wangenheim (2003).

Segundo Wangenheim e Wangenheim (2003), os elementos básicos de um sistema RBC são:

- representação do conhecimento: representar o conhecimento em forma de casos, onde descreve as experiências concretas;
- medida de similaridade: forma de identificar na base de casos um caso relevante ao problema atual;
- adaptação: dificilmente um caso é idêntico ao problema atual, para isso existem mecanismos para adaptar os casos recuperados, com o objetivo de verificar se satisfazem a situação presente;
- aprendizado: sempre que o sistema resolver um caso, deve ser capaz de armazenar e utilizar essa situação no futuro, como um novo caso de sua base.

Segundo Wangenheim e Wangenheim (2003), quando um novo caso é encontrado, o RBC relembra casos similares e adapta as soluções que funcionaram no passado ao problema corrente, usando como modelo do processo o Ciclo de RBC proposto por Aamondt e Plaza (1994), apresentado na Figura 6.

Figura 6 - Ciclo do RBC



Fonte: adaptado de Aamondt e Plaza (1994).

Segundo Aamondt e Plaza (1994), o ciclo do Raciocínio Baseado em Casos pode ser descrito pelos seguintes passos:

- a) RECUPERAR o caso ou os casos mais similares presentes na base;
- b) REUTILIZAR as informações e conhecimentos nesses casos para resolver o problema atual;
- c) REVISAR a solução proposta;
- d) RETER as partes desta experiência que possam ser úteis para a futura resolução de problemas.

Para Aamondt e Plaza (1994), um novo problema é resolvido recuperando um ou mais casos previamente salvos na base de conhecimento. Após recuperar os casos anteriores na base de casos, são adaptadas e reutilizadas as informações dos casos similares para solução do caso atual. A solução encontrada é revisada e se atender o problema do caso, a experiência do caso atual é retida e incorporada na base de conhecimento existente.

Para Kipper, Frozza e Urnau (2014), a avaliação da similaridade é o ponto crucial do raciocínio baseado em casos, ocorrendo após a identificação das características do problemas de entrada em comparação com as características dos casos contidos na base. Determinar o grau de similaridade de um caso consiste em estabelecer o quão parecido ou semelhante a situação atual se aproxima com alguma armazenada no banco de casos (CARVALHO, 2008).

Dessa maneira, Kipper, Frozza e Urnau (2014) argumentam que a técnica do vizinho mais próximo é frequentemente utilizada, onde a similaridade entre um novo caso e o caso existente é determinada para cada atributo. A medida de cada atributo deve ser multiplicada por um fator (peso) e calculada a somatória de todos os atributos. Na Figura 7 é apresentada a função de Similaridade entre dois casos.

Figura 7 - Função de Similaridade

$$\text{Função de Similaridade}(N, F) = \sum_{i=1}^n (N_i, F_i) * W_i$$

Legenda:

- N= Novo caso.
- F= Caso existentes na memória de casos.
- n= Número de atributos.
- i= Atributo individual variando de 1 a n.
- f= Função de similaridade para o atributo i nos casos N e F.
- W= Peso do atributo i.

Fonte: Kipper, Frozza e Urnau (2014).

Segundo Kipper, Frozza e Urnau (2014), o cálculo será repetido para toda a base de casos, para a obtenção da classificação da similaridade. Um conjunto dos casos é apresentado pela tarefa de seleção, considerando as melhores escolhas durante o processo. Nessa lista o caso que melhor satisfaz a consulta do problema, poderá ser o mais útil para solucionar o caso atual.

O Raciocínio Baseado em Casos tem sido usado para a criação de numerosas aplicações referentes a diversos domínios de conhecimento e utilização, sendo algumas dessas aplicações voltadas para análise financeira, assessoramento de riscos, manutenção técnica, controle de processos, controle da qualidade, diagnóstico médico, sistemas de suporte a software, previsão, planejamento, projeto, classificação, avaliação imobiliária, suporte à usuário, gestão do conhecimento, etc. (WANGENHEIM; WANGENHEIM, 2003).

Para Carvalho (2008) a flexibilidade dos sistemas com RBC, com sua adaptação em vários domínios e contextos para representar o conhecimento através de diversas formas, indica sua eficiência na resolução de problemas complexos. Uma vez que um problema

complexo é resolvido com sucesso, pode ser armazenado na base de casos e servir como ponto de partida a resolução de outros problemas similares (CARVALHO, 2008).

## 2.4 TRABALHOS CORRELATOS

A seguir estão relacionados três trabalhos correlatos. A Seção 2.4.1 detalha o Sistema RBC para Recrutamento e Seleção de Profissionais de Silva (2004), software para auxiliar o setor de RH da empresa ao analisar as competências dos candidatos aos requisitos solicitados para vaga disponível. Na Seção 2.4.2 aborda o Desenvolvimento de Sistema Online para Recrutamento de Pessoas elaborado por Preis (2013), que disponibiliza um sistema online para cadastro de candidatos e vagas oferecidas por empresas. A Seção 2.4.3, apresenta o SIIEE – Sistema de Raciocínio Baseado em Casos Para Análise de Perfil de Candidatos a Vagas de Estágio de Panceri (2012), para análise de perfil de candidatos para vaga de estágio.

### 2.4.1 Sistema RBC para recrutamento e seleção de profissionais

O trabalho de Silva (2004) apresenta um protótipo para auxiliar no recrutamento e seleção de profissionais, utilizado pelo administrador de recursos humanos. Nele é possível visualizar as vagas em aberto e buscar por candidatos cadastrados no banco de dados que se assemelhem aos requisitos solicitados pela vaga.

O projeto disponibiliza uma interface para cadastro dos currículos e vagas disponíveis na empresa, e utiliza técnicas de RBC e vizinho mais próximo para encontrar a similaridade entre a vaga e os currículos cadastrados. Para o desenvolvimento do sistema foi utilizado o ambiente de desenvolvimento Delphi e o banco de dados Interbase (SILVA, 2004).

Silva (2004) conclui que o protótipo desenvolvido de RBC possibilitou a utilização do conhecimento do especialista no apoio a decisões administrativas. Para Silva (2004), na área de gerência de recursos humanos, a utilização de sistemas de RBC apresenta-se como uma técnica adequada para a rápida escolha dos profissionais mais adequados para a vaga existente na organização, tornando mais ágil o processo de recrutamento e seleção. A Figura 8 apresenta a tela de cadastro do candidato, onde são preenchidas as informações pessoais do candidato, idiomas, formação, contatos, cursos, cargo pretendido e emprego anterior, para cadastro do currículo.

Figura 8 - Tela de cadastro do candidato

The screenshot shows a web application window titled "Cadastrados" with a menu bar containing "Dados Pessoais", "Idiomas", "Formação", "Contatos", "Cursos", "Cargo Pretendido", and "Emprego Anterior". The "Dados Pessoais" tab is selected, displaying a form with the following fields and values:

- Dados do Candidato:**
  - Código: 5
  - Nome: JANILE FERNANDA DE LIMA
  - Sexo: Femi.
  - Nascto: 18/10/1980
  - Est. Civil: Solteiro
  - RG: 4545454545
  - Emissão: 15/02/2004
  - Orgão: SSPSC
  - CPF: 155.155.155-1
  - Nacionalidade: Brasileiro
  - Possui Veículo:  Sim  Não
  - Possui Habilitação:  Sim  Não
  - Mod. Carteira: [empty]
- Endereço:**
  - Logradouro: RUA FREI JUSTINO
  - Número: 35
  - Compl.: CASA 2º ANDAR
  - Bairro: 1 Centro
  - CEP: 88.888-888
  - Cidade: 1 Leges
  - UF: 1 SC
- Outros Dados Pessoais:**
  - Deficiência: 3 Não possui
  - Conjuge: [empty]
  - Filhos:** [empty table with columns Nome and Nascimento]
  - Comentários:** teste

Buttons at the bottom include "Gravar", "Cancelar", and "Fechar".

Fonte: Silva (2004).

#### 2.4.2 Desenvolvimento de sistema online para recrutamento de pessoas

O sistema desenvolvido por Preis (2013) tem como propósito a informatização do serviço de recrutamento de pessoas realizado pelo SAE (Serviço de Apoio ao Estudante) da FAI faculdades. Através de sua interface online é possível ser acessado por pessoas e empresas, onde a pessoa pode cadastrar seu currículo e verificar as áreas e vagas de seu interesse, enquanto as empresas podem cadastrar suas vagas e verificar o currículo dos candidatos cadastrados (PREIS, 2013).

Na Figura 9 é apresentada a tela com as listagens de vagas cadastradas pela empresa. Nessa tela são apresentadas as informações sobre a empresa que cadastrou o cargo, a quantidade de candidatos interessados na vaga, o prazo final em que a vaga ficará disponível e opções para editar ou excluir a vaga. Nesta tela também está disponível a opção para cadastrar uma nova vaga.



Figura 9 - Lista de vagas cadastradas

Empresa	Cadastro feito por	Cargo	Quant. Interess	Disponível Até ↑	Opções
Empresa TI ...	Empresa TI ...	Programação...	0	30/10/2013	[Red] [Yellow]

Novo

Página 1 de 1 | Exibindo 1 - 1 de 1

Fonte: Preis (2013).

Ao entrar na aplicação com o perfil de candidato, é possível visualizar a listagem das vagas disponíveis cadastradas por outras empresas. Nessa tela o candidato pode informar o interesse ou não nas vagas disponíveis. A Figura 10 apresenta a listagem de vagas disponíveis visível para o candidato.

Figura 10 – Lista de vagas disponíveis

### Lista de Vagas Disponíveis

**Panfletagem - Vaga para Estágio**

**Descrição:** Realizar Panfletagem. Vestibular Em toda a região Oeste de Santa Catarina e Noroeste do Rio Grande do Sul

**Benefícios:** Vale Alimentação. Vale Transporte

**Salário:** R\$ 900,00

**Informações de Contato:** Falar com: Giovane Preis  
Telefone: (49) 8423-6859  
E-mail: giupreis@gmail.com

**Disponível:** De 05/10/2013 até 31/10/2013

Concorrer à Vaga

**Programação Web**

**Descrição:** Desenvolvimento de sistemas online

**Benefícios:** Este cargo é um teste do sistema

**Salário:** 1000,00

**Informações de Contato:** Empresa: Empresa TI Profissional  
Falar com: Giovane  
Telefone: (99) 9999-9999  
E-mail: empresa@tiprofissional.com

**Disponível:** De 27/10/2013 até 31/10/2013

Não Concorrer à Vaga

Fonte: Preis (2013).

Este estudo propôs o desenvolvimento de uma aplicação web capaz de informatizar a área de recrutamento de pessoas na gestão de Recursos Humanos aproximando interessados às vagas de empregos disponíveis. Este objetivo foi alcançado com o desenvolvimento do SisVC, Sistema de Vagas e Currículos (PREIS, 2013). Preis (2013) conclui que os sistemas online para recrutamento de pessoas serão grandes ferramentas de auxílio na busca por profissionais qualificados, sendo que as informações estarão desta forma, facilmente acessíveis tanto para as pessoas quanto para as empresas.

#### 2.4.3 SIIEE – Sistema de raciocínio baseado em casos para análise de perfil de candidatos a vagas de estágio

O sistema SIIEE desenvolvido por (PANCERI, 2012) disponibiliza telas para cadastro de cursos, alunos, currículos, funcionários, empresa e oportunidade de trabalho. Utiliza raciocínio baseado em casos para relacionar os alunos às vagas disponíveis.

A Figura 11 apresenta a tela que contém os dados que são utilizados para similaridade e o valor de sua importância, que define a porcentagem que cada dado terá na busca do currículo que melhor atende as condições da vaga. Nessa tela o usuário escolhe entre as oportunidades cadastradas que deseja realizar a busca. Após selecionar a vaga, podem ser selecionados os atributos e efetuado a busca dos currículos cadastrados.

Figura 11 - Tela de busca de currículos

SIIEE - Buscar Currículos

**Selecionar Oportunidade**

Código da Oportunidade 1

Horário

**Residência**

Cidade

**Conhecimentos Gerais**

Informática

Espanhol

Inglês

**Conhecimentos Específicos**

**Dados Acadêmicos**

Curso

Ano/Semestre

Nível Escolaridade

Cancel

Executar Busca

Importância 1.0

1.0

1.0

1.0

1.0

1.0

1.0

1.0

1.0

Fonte: Panceri (2012).

Para o desenvolvimento do sistema, Panceri (2012), utilizou a linguagem Java e o banco de dados MySQL, e com base nos estudos efetuados sobre as áreas de recrutamento e seleção, inteligência artificial, aquisição de conhecimento e RBC, foi possível implementar as regras de negócios básicas para o funcionamento do sistema, bem como a aplicação de algoritmos para auxiliar na atividade de seleção.

### 3 DESENVOLVIMENTO

Neste capítulo é apresentado o desenvolvimento da aplicação. A seção 3.1 apresenta os requisitos funcionais e não funcionais da aplicação desenvolvida. Na seção 3.2 está a especificação contendo os diagramas da parte principal da aplicação. Na seção 3.3 está descrita a implementação das principais partes da aplicação, assim como as ferramentas utilizadas e a operacionalidade da aplicação. Por fim, na seção 3.4 são apresentados os testes realizados e os resultados obtidos.

#### 3.1 REQUISITOS

Nessa seção são apresentados os principais Requisitos Funcionais (RF) e os Requisitos Não Funcionais (RNF).

- a) permitir realizar *login* no sistema (Requisito Funcional - RF);
- b) permitir cadastrar currículos recebidos pela empresa (RF);
- c) permitir cadastrar vagas disponíveis na empresa (RF);
- d) permitir configurar o peso dos atributos usados no cálculo da similaridade (RF);
- e) aplicar a técnica de RBC para comparar a similaridade entre os candidatos e as vagas disponíveis (RF);
- f) exibir um relatório com o resultado da similaridade da vaga processada (RF);
- g) permitir acesso ao candidato para executar o próprio cadastro (RF);
- h) possibilitar ao candidato executar a busca de informações no LinkedIn, para completar seu cadastro (RF);
- i) implementar a aplicação usando Java, JavaScript e HTML (Requisito Não Funcional - RNF);
- j) implementar as páginas da aplicação utilizando o framework Primefaces (RNF);
- k) utilizar Java Runtime Environment (JRE) versão 8 ou superior para que a ferramenta possa ser executada (RNF).

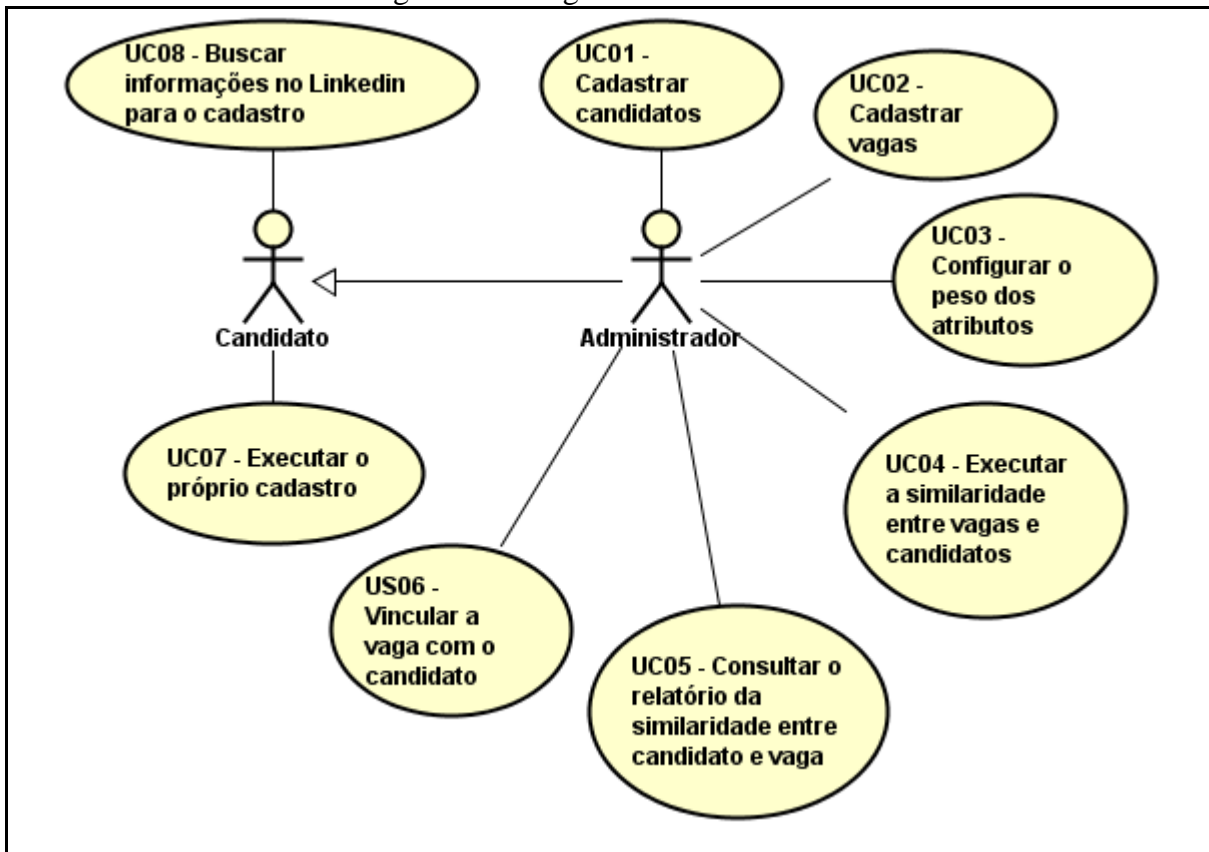
#### 3.2 ESPECIFICAÇÃO

Para o entendimento das funcionalidades e arquitetura da plataforma, foram desenvolvidos diagramas seguindo o padrão Unified Modeling Language (UML), utilizando as ferramentas Astah Community para o diagrama de casos de uso, classes e atividade. A ferramenta MySQL Workbench foi usada para o desenvolvimento do Modelo Entidade Relacionamento (MER). Também são apresentados detalhes da comunicação com a Application Programming Interface (API) do LinkedIn, para buscar dos dados do candidato.

### 3.2.1 Diagrama de casos de uso

A Figura 12 demonstra o diagrama de casos de uso, na qual o administrador é responsável por todos os cadastros, bem como configurar o peso dos atributos e executar a similaridade entre as vagas e os candidatos. O candidato pode executar o próprio cadastro, que ficará disponível para o Profissional de RH, para verificar a similaridade com as vagas.

Figura 12 - Diagrama de casos de uso



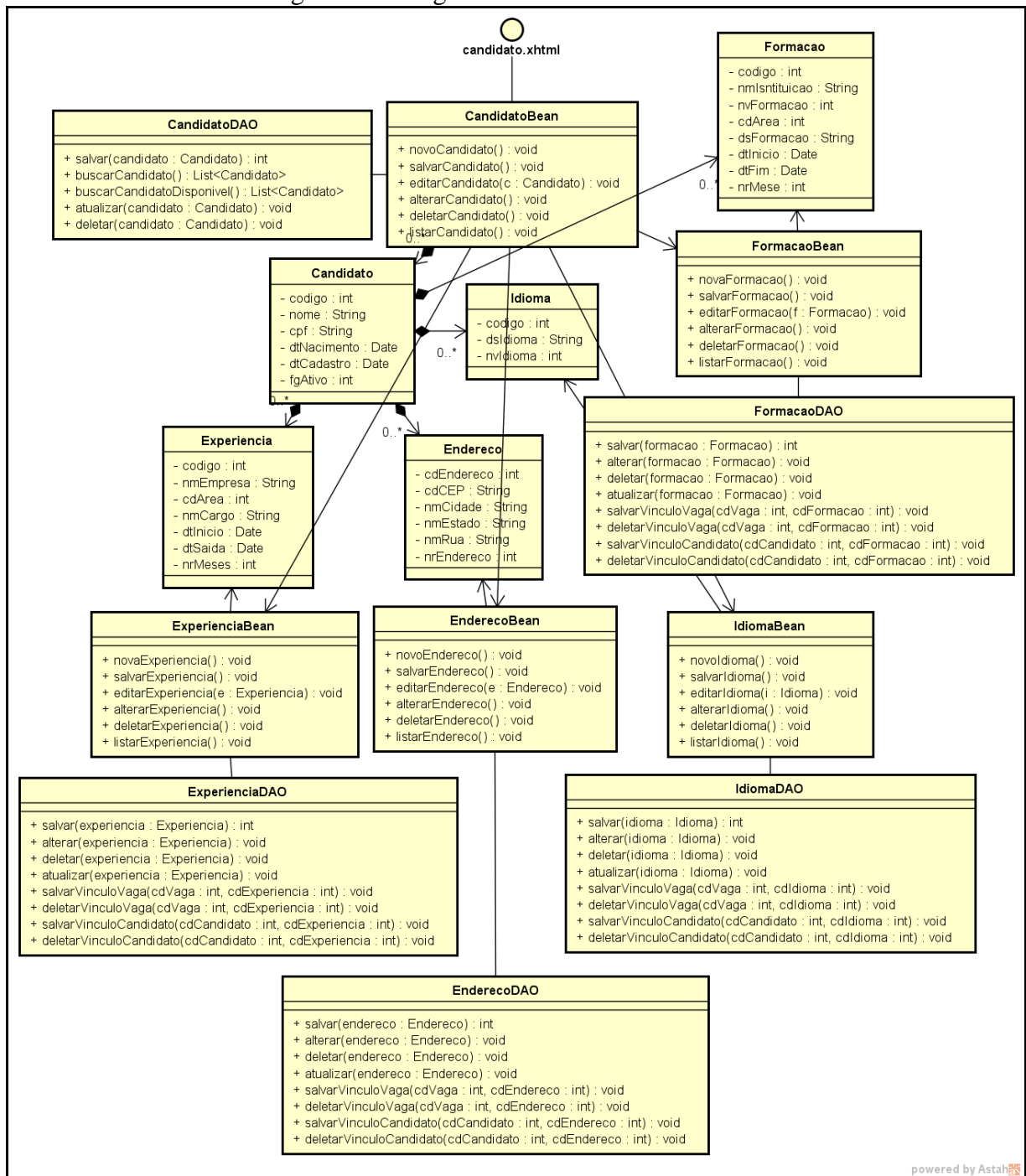
Fonte: elaborado pelo autor.

A Figura 12 apresentou as ações que o Administrador e Candidato podem executar na aplicação. O detalhamento de cada caso de uso pode ser visto no Apêndice A.

### 3.2.2 Diagrama de classes

O objetivo desta seção é apresentar os diagramas de classes implementadas para o desenvolvimento da aplicação. Os diagramas foram separados entre as classes utilizadas para o funcionamento das páginas Web, e as utilizadas para executar o processo do RBC. A Figura 14 apresenta as principais classes e métodos usados para o funcionamento da página Web de Candidatos.

Figura 13 - Diagrama de classes do candidato

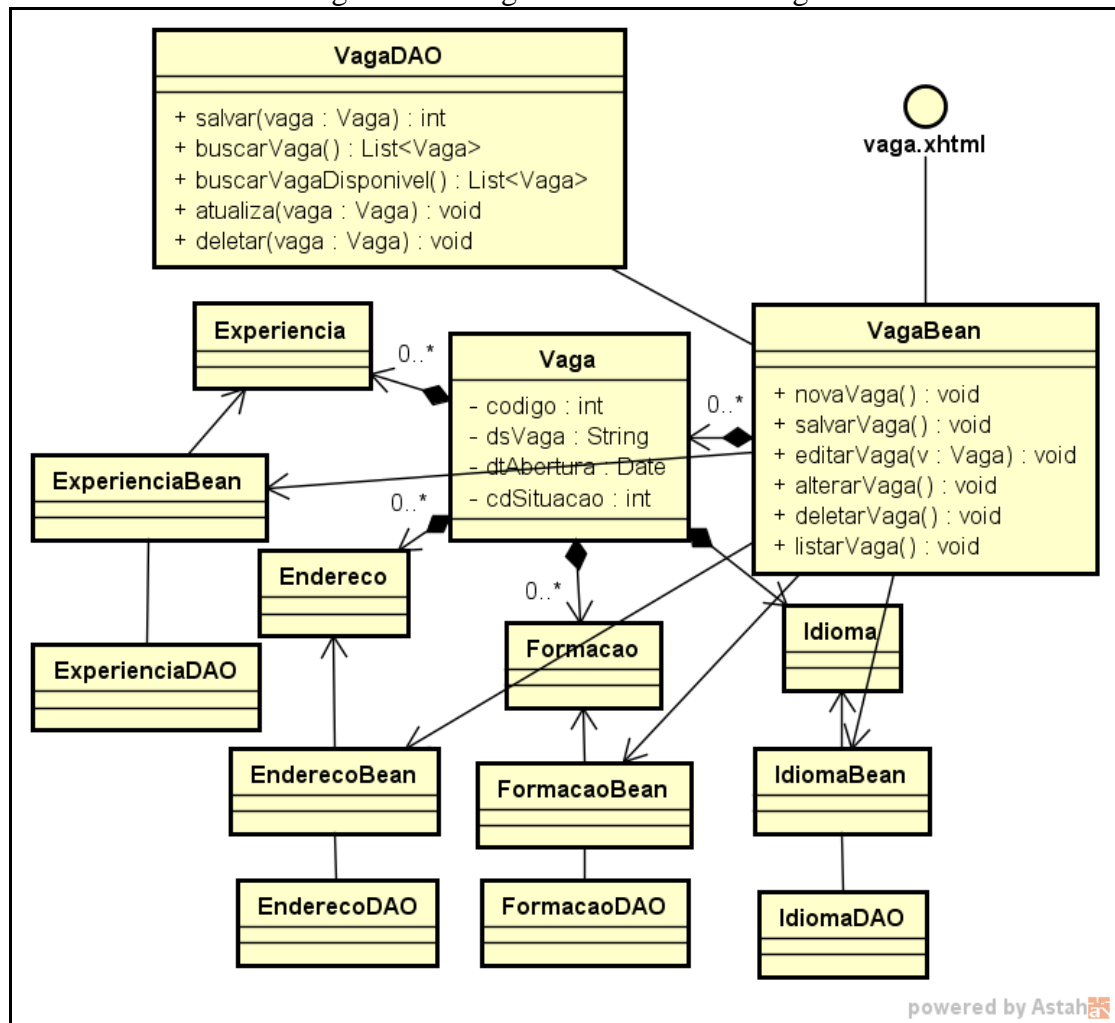


Fonte: elaborado pelo autor.

Conforme apresentado na Figura 14, é aplicado o estilo de arquitetura MVC (Model-View-Controller). A View da aplicação é representada pelas páginas XHTML representada pela página `candidato.xhtml`, essa camada é responsável pela interação do usuário com o sistema. A parte Controller pode ser vista através das classes `CandidatoBean`, `EnderecoBean`, `ExperienciaBean`, `FormacaoBean`, `IdiomaBean` e suas respectivas classes de Objetos: `Candidato`, `Endereco`, `Experiencia`, `Formacao` e `Idioma`. Essas classes são utilizadas para manipulação e exibição das informações sobre o candidato em tela. Também é

executado o controle dos registros no banco, onde as classes `Bean` fazem a chamada dos métodos presentes nas classes `DAO` para registro na base. As classes `CandidatoDAO`, `ExperienciaDAO`, `EnderecoDAO`, `FormacaoDAO` e `IdiomaDAO` representa a camada `Modal` da arquitetura `MVC`, efetuando a comunicação com o banco de dados. A Figura 14 apresenta as classes relacionadas a página da vaga. As classes já detalhadas nos diagramas anteriores foram simplificadas neste diagrama, indicando somente sua utilização.

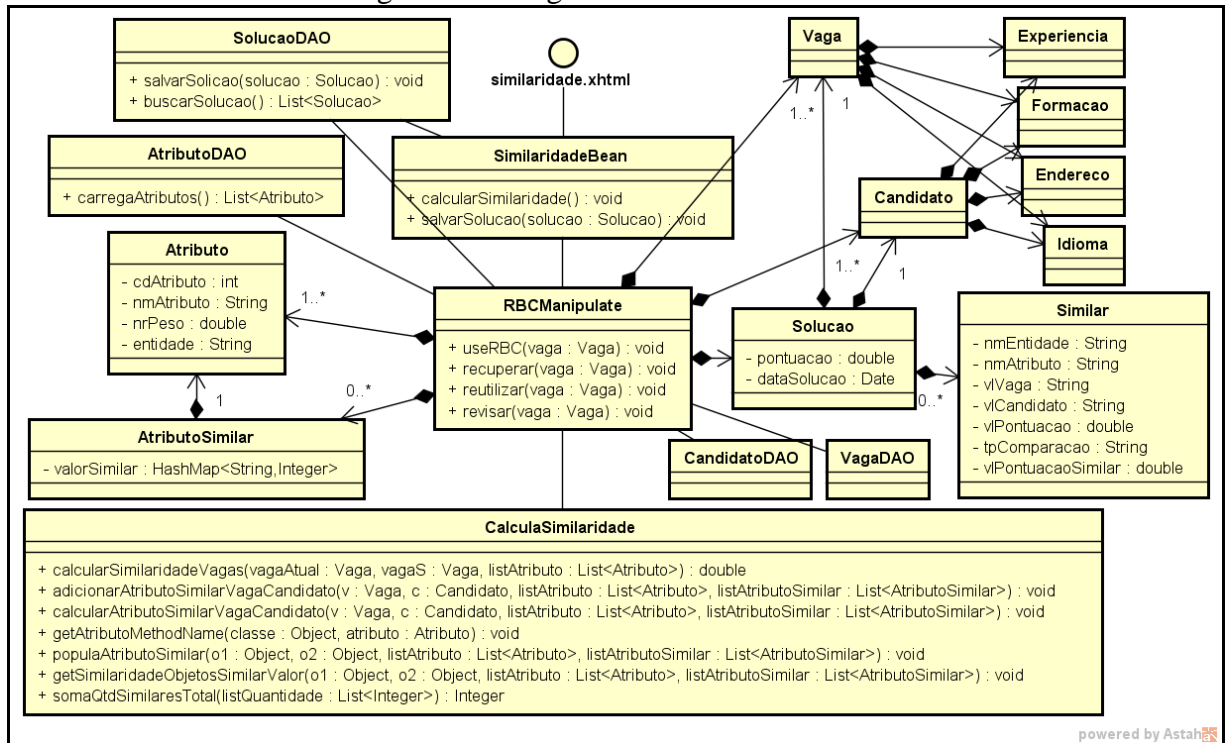
Figura 14 - Diagrama de classes da vaga



Fonte: elaborado pelo autor.

Na Figura 15 é apresentado o diagrama de classe sobre a utilização das técnicas de `RBC`. No diagrama são apresentadas as principais classes e métodos utilizados para executar o cálculo da similaridade entre a vaga e os candidatos. As classes já detalhadas nos diagramas anteriores foram simplificadas neste diagrama, indicando somente sua utilização.

Figura 15 - Diagrama de classes do RBC



Fonte: elaborado pelo autor.

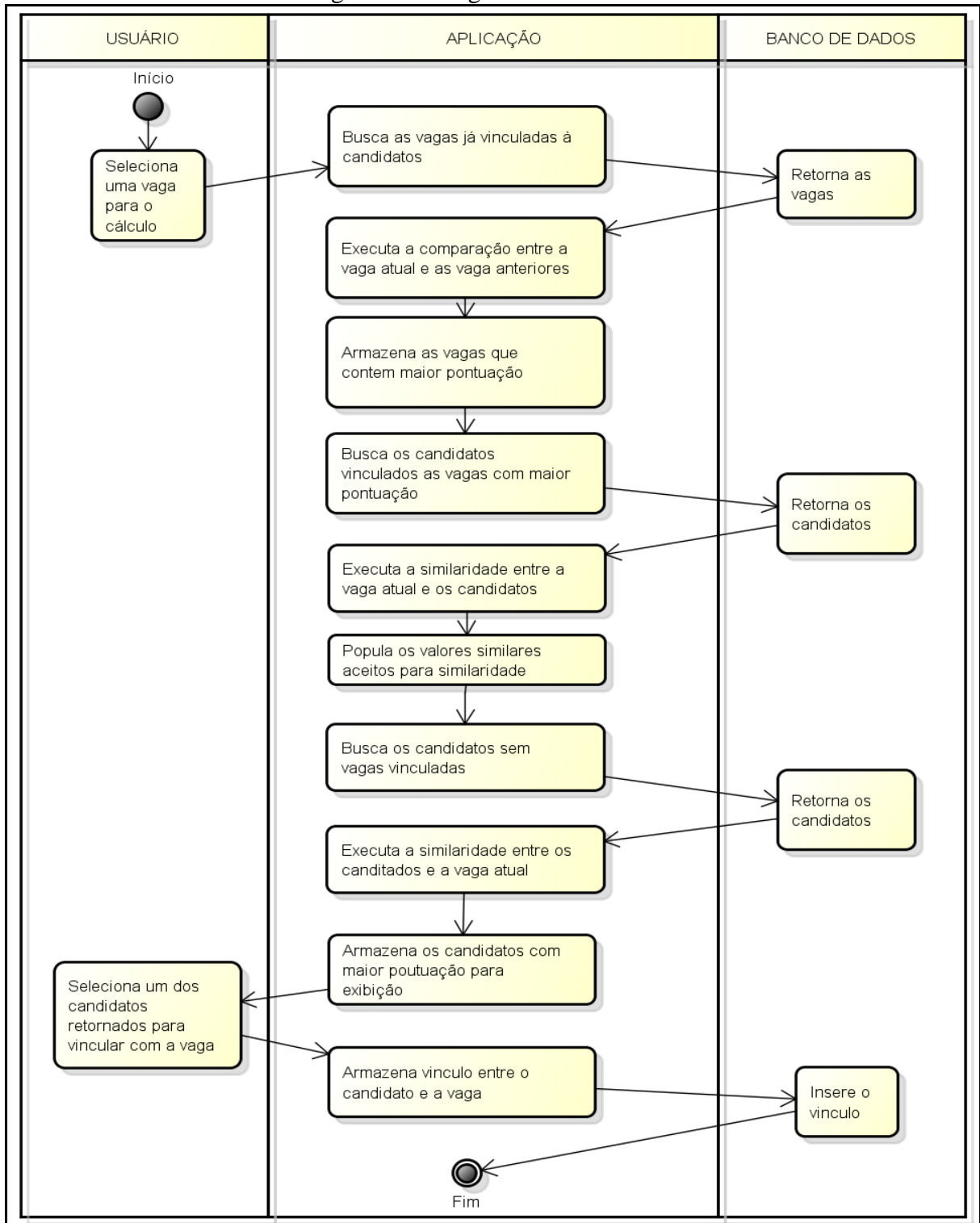
Conforme visto na Figura 15, a página `similaridade` é a responsável por chamar a comparação entre a vaga que o usuário do RH selecionar e os candidatos disponíveis no banco de dados. O processo detalhado será descrito a seguir.

### 3.2.3 Diagrama de atividade

Nesta seção é apresentado o diagrama de atividades responsável pelo fluxo da execução da similaridade. Esse processo necessita que já existam vagas e candidatos registrados na base de dados. A Figura 16 demonstra o funcionamento da similaridade com as técnicas de RBC.



Figura 16 - Diagrama de atividades



Fonte: elaborada pelo autor.

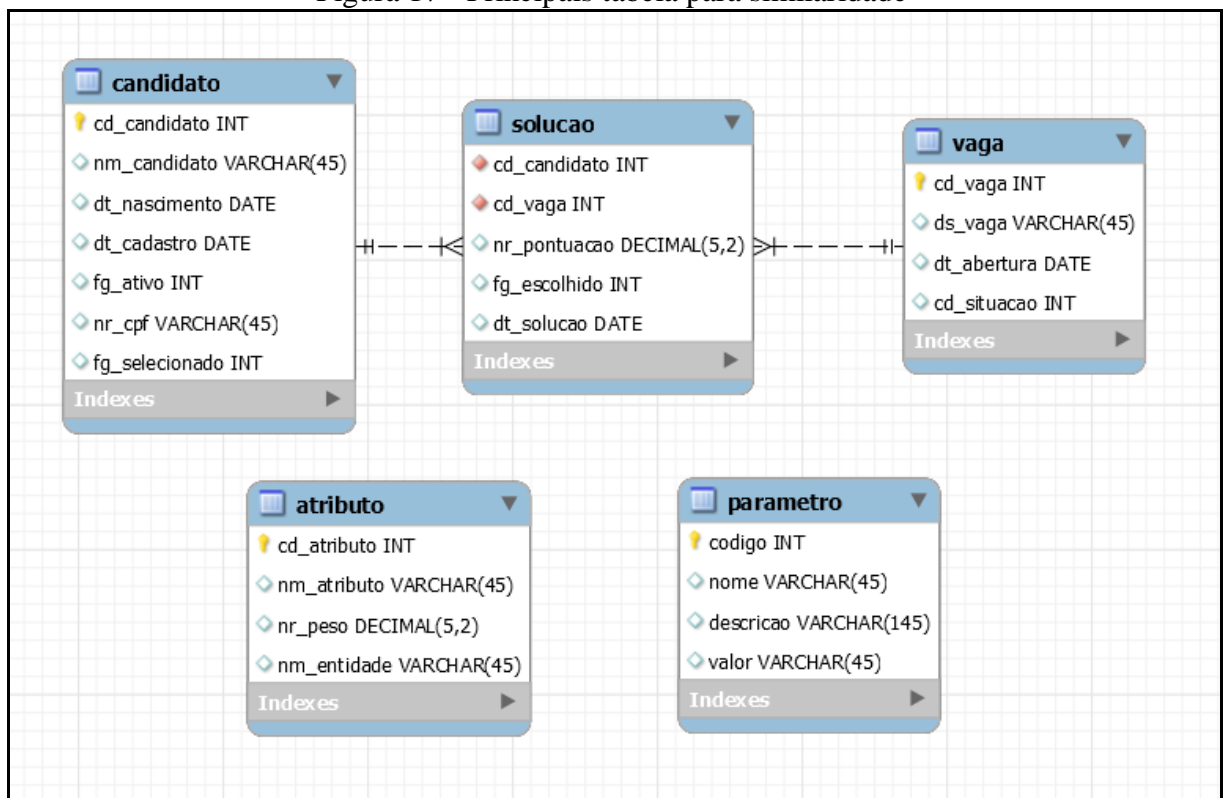
Na Figura 16 pode-se observar que o processo de cálculo se inicia com a seleção da vaga, referente ao início do ciclo do RBC, representando a entrada de um novo problema para procura de uma solução. Para que essa solução seja encontrada é utilizado a base de dados, consultando os vínculos entre vagas e candidatos anteriores, utilizando essa informação para resolver o caso atual.

Na parte final do cálculo da similaridade entre a vaga e os candidatos, são retornados os candidatos que conseguiram a maior pontuação, auxiliando na tomada de decisão executada pelo usuário. A escolha final sobre qual candidato será utilizado é tomada pelo administrador, onde sua escolha será salva e usada para solucionar os casos futuros.

### 3.2.4 Modelo Entidade Relacionamento

A Figura 17 apresenta o MER que modela as principais tabelas utilizadas pela aplicação para cadastro e configuração das informações necessárias na similaridade entre o candidato e a vaga.

Figura 17 - Principais tabelas para similaridade

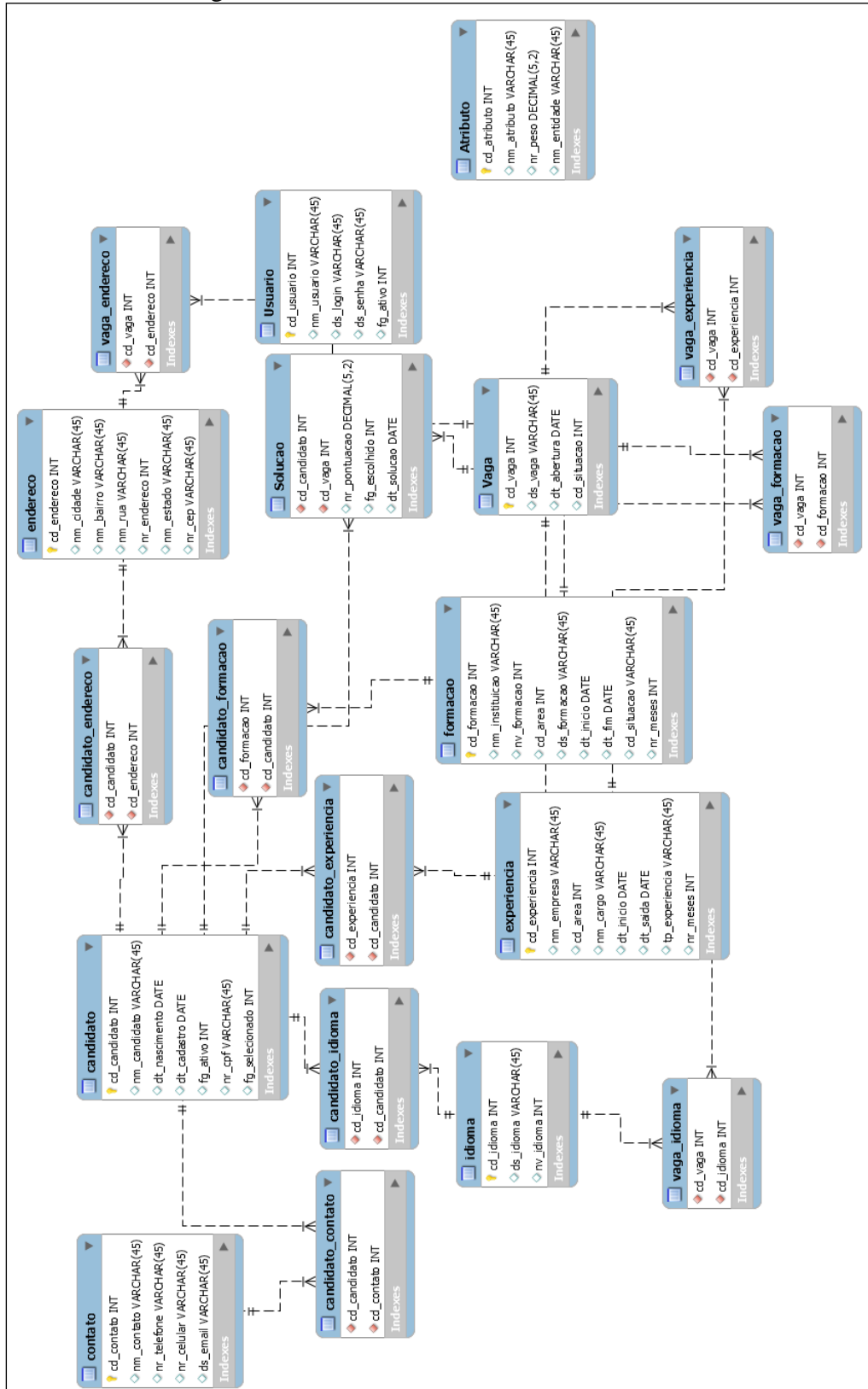


Fonte: elaborada pelo autor.

Conforme apresentado na Figura 17 a tabela *solucao* é fundamental para utilização das técnicas de RBC. Esta tabela armazena o vínculo entre as tabelas *candidato* e *vaga*, onde são recuperadas as soluções armazenadas para auxiliar na solução da nova vaga. A tabela *parametro* mantém algumas configurações para pesquisa de registros no banco de dados usados durante o cálculo da similaridade. Para pontuação do candidato durante a similaridade é utilizada a tabela *atributo*, onde são armazenados os registros das características utilizadas para pontuação e o peso referente a cada atributo. No Apêndice B é apresentado o dicionário

de dados referente as tabelas apresentadas na Figura 17. Na Figura 18 é apresentado o MER completo do banco de dados da aplicação.

Figura 18 - Modelo Entidade Relacionamento



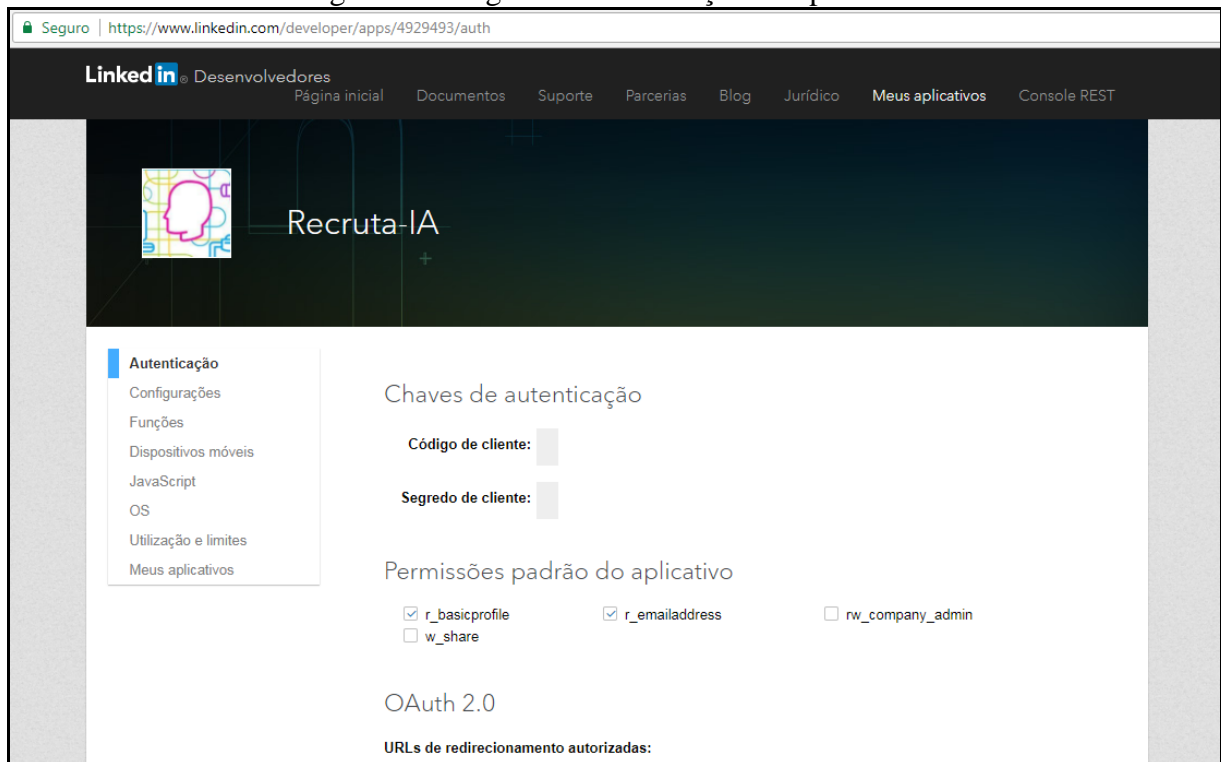
Fonte: elaborada pelo autor.

### 3.2.5 Comunicação com LinkedIn

O LinkedIn disponibiliza uma API para comunicação com outras aplicações, sendo possível através do uso da mesma, executar *login*, compartilhar imagens e outros conteúdos, adicionar certificados ao perfil do usuário, gerenciar a página da empresa, entre outras possibilidades (LINKEDIN, 2017).

Para uso da API é necessário criar uma conta no LinkedIn e cadastrar a aplicação que executará a comunicação com o LinkedIn. Fazendo isso será disponibilizado um código de cliente, utilizado para a aplicação se comunicar com o LinkedIn. A Figura 19 demonstra a tela onde é possível configurar alguns dados de autenticação utilizados pelo aplicativo como permissões do aplicativo e URL permitidas para executar a comunicação.

Figura 19 - Página de autenticação do aplicativo



Fonte: elaborada pelo autor.

A integração do Recruta-IA com a API foi executada com o uso da linguagem JavaScript, mas o LinkedIn também disponibiliza a integração através de API REST, JavaScript SDK, Android SDK ou iOS SDK (LINKEDIN, 2017). Através das permissões `r_basicprofile` e `r_emailaddress`, é possível buscar algumas informações sobre o candidato e seus dados profissionais, como nome completo, e-mail, trabalho atual e outras informações. O Quadro 1 descreve os dados que podem ser consultados usando a permissão `r_basicprofile`.

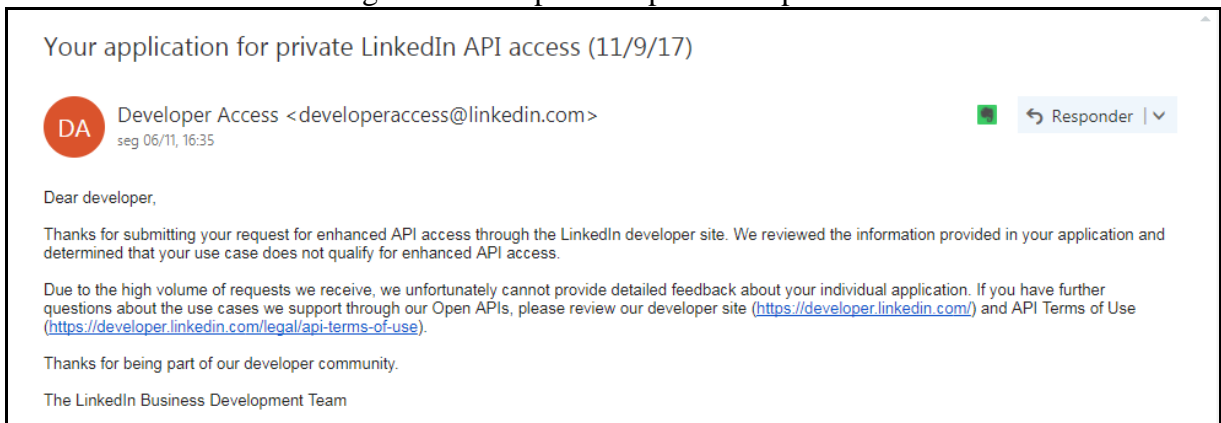
Quadro 1 - Campos acessados com a permissão básica

Nome do campo	Descrição
id	Valor de identificação exclusivo para o membro.
first-name	O primeiro nome do membro
last-name	O sobrenome do membro.
maiden-name	O nome de solteiro do membro.
formatted-name	O nome do membro, formatado com base no idioma.
phonetic-first-name	Primeiro nome do membro, soletrado foneticamente.
phonetic-last-name	Sobrenome do membro, soletrado foneticamente.
formatted-phonetic-name	O nome do membro, soletrado foneticamente e formatado com base em linguagem.
headline	O título do membro.
location	Um objeto que representa a localização física do usuário.
industry	Indústria à qual o membro pertence.
current-share	O item mais recente que o membro compartilhou no LinkedIn.
num-connections	O número de conexões do LinkedIn que o membro possui.
num-connections-capped	Retorna 'true' se o valor 'num-connections' do membro tiver sido limitado a 500 '
summary	Uma área de texto de longa duração que descreve o perfil profissional do membro.
specialties	Uma área de texto de formato curto que descreve as especialidades do membro.
positions	Um objeto representando a posição atual do membro.
picture-url	URL para a imagem de perfil formatada.
picture-urls::(original)	URL para a imagem de perfil original não formatada.
site-standard-profile-request	URL para o perfil autenticado do membro no LinkedIn.
api-standard-profile-request	URL que representa o recurso que você solicitaria para acesso programático ao perfil do membro.
public-profile-url	URL do perfil público do membro no LinkedIn.

Fonte: adaptado de LinkedIn (2017).

Utilizando a permissão `r_basicprofile` não é possível acessar informações como empregos anteriores, formação educacional ou idiomas que o usuário possui. Para isso é necessário se tornar parceiro do LinkedIn, onde é solicitado através de um formulário preenchido no site do LinkedIn, que será analisado e caso seja aprovado, habilitará a permissão `r_fullprofile`, permitindo funcionalidade e acesso a dados que antes não eram possíveis (LINKEDIN, 2017). Infelizmente para aplicação Recruta-IA, não foi possível adquirir a permissão para acesso total as informações do perfil, conforme apresenta o e-mail de resposta do LinkedIn, retratado na Figura 20.

Figura 20 - Resposta ao pedido de parceria



Fonte: elaborada pelo autor.

Dessa forma, foi utilizada a permissão `r_basicprofile` e `r_emailaddress` para busca de informações básicas do usuário e demonstrar o funcionamento da aplicação com a API. A próxima seção traz os detalhes sobre a implementação da aplicação.

### 3.3 IMPLEMENTAÇÃO

A seguir são apresentadas as técnicas e ferramentas utilizadas para uso do RBC no recrutamento de candidatos, bem como o gerenciamento dos dados e implementação do Recruta-IA. Também é apresentado o cálculo da similaridade e o uso dos casos anteriores para solução do problema atual.

#### 3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento da aplicação utilizou-se a linguagem Java, juntamente com o *framework* PrimeFaces e a ferramenta Maven para o gerenciamento de dependências. O desenvolvimento das páginas web foram construídas utilizando a especificação JavaServer Faces (JSF), junto ao uso de códigos JavaScript, HTML e CSS. Para armazenar e manipular dados utilizados pela aplicação foi utilizado o banco de dados MySQL, modelado com o uso da ferramenta MySQL Workbench 6.3, utilizando o servidor de aplicações Apache Tomcat, tornando possível acessar a aplicação através da URL configurada.

Nas próximas seções são detalhados os passos adotados na implementação das técnicas do RBC, descrevendo cada passo da técnica e a fórmula para pontuação da similaridade. Também é apresentado como funciona a comunicação do Recruta-IA com a API do LinkedIn, permitindo a busca de alguns dados sobre o candidato.

### 3.3.1.1 Raciocínio baseado em casos

A descrição das técnicas do RBC para calcular a similaridade entre a vaga e os candidatos disponíveis, foram divididas em 4 etapas: recuperação, reutilização, revisão e retenção. Na etapa de recuperação, são utilizadas as soluções anteriores já presentes no banco de dados. As soluções são compostas pela vaga e o candidato que melhor atendeu os requisitos solicitados, segundo critérios definidos pelo RH. Após a busca das soluções é comparada a vaga que se deseja solucionar com as vagas presentes nas soluções anteriores, para utilizar as soluções com maior similaridade da vaga atual. No Quadro 2 é apresentado a implementação do método `recuperar`, referente a primeira etapa do RBC.

Quadro 2 - Método para implementar a etapa recuperar do RBC

```

public static void recuperar(Vaga vagaAtual) {
    int nrQtdVagas =
Integer.parseInt(ParametroDAO.buscarValorParametro("recruta.qtd.vagas.similares"));
    List<Solucao> listSolucaoConsultada = new ArrayList<Solucao>();
    List<Solucao> listSolucaoPontuacaoCalculada = new ArrayList<Solucao>();
    listSolucaoConsultada = SolucaoDAO.buscarVagaComSolucao();
    listSolucoes = new ArrayList<Solucao>();
    for (Solucao solucaoUnica : listSolucaoConsultada) {
        solucaoUnica.setVaga(carregaVaga(solucaoUnica.getVaga().getCodigo()));
        solucaoUnica.setCandidato(carregaCandidato(
solucaoUnica.getCandidato().getCodigo()));
        solucaoUnica.setNrPontuacao(
            CalculaSimilaridade.calcularSimilaridadeVagas(vagaAtual,
solucaoUnica.getVaga(), listAtributos));
        listSolucaoPontuacaoCalculada.add(solucaoUnica);
    }
    SolucaoController.ordenarNrPontuacaoDesc(listSolucaoPontuacaoCalculada);
    for (Solucao solucaoUnica : listSolucaoPontuacaoCalculada) {
        listSolucoes.add(solucaoUnica);
        if (listSolucoes.size() >= nrQtdVagas) {
            break;
        }
    }
    reutilizar(vagaAtual);
}
}

```

Fonte: elaborado pelo autor.

Conforme pode ser visto no Quadro 2, após armazenar as soluções com maior pontuação, é utilizada a chamada do método `reutilizar`, referente a próxima etapa do RBC. Nessa etapa são utilizadas as soluções que tiveram maior similaridade com a vaga atual, para que seja aproveitado o conhecimento anterior dos casos passados na resolução do caso atual. Para carregar esse conhecimento é executada uma comparação entre a vaga atual e os candidatos vinculados às soluções anteriores, para ser criada uma lista dos possíveis valores aceitáveis durante a comparação dos atributos. O armazenamento dos valores aceitos auxilia na comparação final entre a vaga disponível e os candidatos habilitados, onde os candidatos não necessitam ter o valor exato solicitado na vaga para pontuarem durante o cálculo da similaridade, pois é possível a pontuação se o candidato possuir a mesma informação que um

candidato contratado anteriormente. No Quadro 3 é apresentado o método utilizado para popular os valores aceitos da similaridade.

Quadro 3 - Método para popular os valores aceitos

```

public static void populaAtributoSimilar(Object objeto1, Object objeto2, List<Atributo>
listAtributos, List<AtributoSimilar> listAtributoSimilar)
throws IllegalAccessException, IllegalArgumentException, InvocationTargetException {
    boolean fg_encontrado = false;
    for (Atributo atributo : listAtributos) {

        //verifica se o Objeto possui o metodo referente ao atributo atual.
        Method metodoAcessivel = getAtributoMethodName(objeto1, atributo);
        if (!(metodoAcessivel == null)) {
            Object valorObjeto = getAtributoMethodName(objeto1, atributo).invoke(objeto1);
            Object valorObjeto2 = getAtributoMethodName(objeto1, atributo).invoke(objeto2);
            if (valorObjeto != null && valorObjeto2 != null) {
                String valor = getAtributoMethodName(objeto1, atributo).invoke(objeto1).toString();
                String valor2 = getAtributoMethodName(objeto1, atributo).invoke(objeto2).toString();

                //se valores forem diferentes, é armazenado como um valor aceito.
                if (!valor.equalsIgnoreCase(valor2)) {
                    for (AtributoSimilar similarUnico : listAtributoSimilar) {
                        //verifica se já existe o atributoSimilar para o atributo.
                        if
                        (similarUnico.getAtributo().getNmAtributo().equalsIgnoreCase(atributo.getNmAtributo())) {

                            //verifica se já exista o valor similar para o valor do candidato.
                            if (similarUnico.getValorSimilar() != null
                            && similarUnico.getValorSimilar().containsKey(valor2)) {
                                similarUnico.getValorSimilar().put(valor2,
                                similarUnico.getValorSimilar().get(valor2) + 1);
                            } else {
                                similarUnico.getValorSimilar().put(valor2, 1);
                            }
                        }
                        fg_encontrado = true;
                        break;
                    }
                }
                //se não existir, adiciona um novo atributoSimilar com o valor do candidato.
                if (!fg_encontrado) {
                    AtributoSimilar atributoSimilar = new AtributoSimilar();
                    atributoSimilar.setAtributo(atributo);
                    atributoSimilar.getValorSimilar().put(valor2, 1);
                    listAtributoSimilar.add(atributoSimilar);
                }
            }
        }
        fg_encontrado = false;
    }
}

```

Fonte: elaborado pelo autor.

Na próxima etapa, revisão, são recuperados os candidatos disponíveis no banco de dados que não foram vinculados a nenhuma vaga. Os candidatos recuperados são utilizados na comparação com a vaga durante o cálculo de similaridade. Após o cálculo são apresentados ao usuário os candidatos que alcançaram a maior pontuação na similaridade. O cálculo da similaridade será melhor descrito na próxima seção.

Na última etapa do RBC, chamada retenção, é armazenada a solução selecionado pelo RH entre as que possuíram a maior pontuação na similaridade, possibilitando que a mesma seja consultada para auxiliar na resolução dos casos, utilizando a experiência do caso atual para futuras verificações.



### 3.3.1.1.1 Cálculo da similaridade

O processo do cálculo da similaridade utiliza as características de dois objetos para comparação. Cada característica é representada por um atributo que possui um fator de importância, definido como peso. Para efetuar o relacionamento dos atributos usados na similaridade com os atributos declarados nas classes são utilizadas técnicas de *Reflection* e *Annotations* da linguagem Java. O Quadro 4 apresenta um trecho de código da classe *Endereco*, onde são usadas *Annotations* para definir qual entidade e atributo a classe e o método se referem.

Quadro 4 - Uso das Anotações em classes

```
@AnotacaoEntidade(nome="endereco")
public class Endereco {
    private Integer cdEndereco;
    private String cdCEP;
    private String nmCidade;
    private String nmEstado;
    private String nmBairro;
    private String nmRua;
    private int nrEndereco;

    @AnotacaoAtributo(atributo="cidade", comparacao=TipoComparacaoEnum.IGUAL)
    public String getNmCidade() {
        return nmCidade;
    }
    public void setNmCidade(String nmCidade) {
        this.nmCidade = nmCidade;
    }

    @AnotacaoAtributo(atributo="estado", comparacao= TipoComparacaoEnum.IGUAL)
    public String getNmEstado() {
        return nmEstado;
    }
    public void setNmEstado(String nmEstado) {
        this.nmEstado = nmEstado;
    }
}
```

Fonte: elaborado pelo autor.

Conforme visto no Quadro 4, a *Annotation* *AnotacaoEntidade* define qual a entidade que a classe se refere, já o uso da *AnotacaoAtributo* define qual atributo se refere o método. Desta forma ao executar a comparação entre dois objetos, é possível buscar os valores dos atributos através do uso de técnicas de reflexão, conforme é demonstrado no Quadro 5.

Quadro 5 – Código para buscar o método referente ao atributo

```
public static Method getAtributoMethodName(Object classe, Atributo atributo) {
    AnotacaoEntidade anotacaoEnt = classe.getClass().getAnnotation(AnotacaoEntidade.class);

    //verifica se a classe utiliza a Anotacao de Entidade do atributo.
    if (anotacaoEnt.nome().equalsIgnoreCase(atributo.getEntidade())) {
        for (Method metodo : classe.getClass().getMethods()) {
            if (metodo.isAnnotationPresent(AnotacaoAtributo.class)) {
                AnotacaoAtributo anotacao = metodo.getAnnotation(AnotacaoAtributo.class);

                // verifica se a Anotacao do metodo é a mesma do Atributo.
                if (anotacao.atributo().equalsIgnoreCase(atributo.getNmAtributo())) {
                    return metodo;
                }
            }
        }
    }
}
```

```

    }
    return null;
}

```

Fonte: elaborado pelo autor.

Através do Quadro 5 é possível verificar o uso das *Annotations* *AnotacaoEntidade* e *AnotacaoAtributo* para retornar o método do objeto que contém o valor do atributo utilizado na comparação da similaridade. O Quadro 6 apresenta o trecho do código que executada a comparação entre os objetos e que retorna a pontuação da similaridade.

Quadro 6 - Método para retornar a pontuação da similaridade entre dois objetos

```

public static BigDecimal getSimilaridadeObjetosSimilarValor(Object objeto1, Object objeto2,
List<Atributo> listAtributos, List<AtributoSimilar> listAtributosSimilares, List<Similar>
listSimilares) throws IllegalAccessException, IllegalArgumentException, InvocationTargetException {
    BigDecimal somaSimilaridade = new BigDecimal(0);

    // percorre a lista de atributos para similaridade.
    for (Atributo atributo : listAtributos) {
        Similar similarUnico = new Similar();

        // verifica se o Objeto possui o metodo referente ao atributo atual.
        Method metodoAcessivel = getAtributoMethodName(objeto1, atributo);
        if (!(metodoAcessivel == null)) {
            Object valorObjeto = getAtributoMethodName(objeto1, atributo).invoke(objeto1);
            Object valorObjeto2 = getAtributoMethodName(objeto1, atributo).invoke(objeto2);
            if (valorObjeto != null && valorObjeto2 != null) {
                String valor = getAtributoMethodName(objeto1,
atributo).invoke(objeto1).toString();
                String valo2r = getAtributoMethodName(objeto1,
atributo).invoke(objeto2).toString();
                TipoComparacaoEnum tipoValor = getAtributoCompareName(objeto1, atributo);

                // caso comparação do atributo for igualdade, verifica valor exato e valores
aceitos.
                if (tipoValor == TipoComparacaoEnum.IGUAL) {
                    if (valor.equalsIgnoreCase(valo2r)) {
                        somaSimilaridade = somaSimilaridade.add(atributo.getNrPeso());
                        populaSimilarUnico(similarUnico, atributo, valor, valo2r, "Igualdade",
atributo.getNrPeso());
                    } else {
                        for (AtributoSimilar unicoAtrSimilar : listAtributosSimilares) {
                            if
(unicoAtrSimilar.getAtributo().getEntidade().equalsIgnoreCase(atributo.getEntidade())
&& unicoAtrSimilar.getAtributo().getNmAtributo().equalsIgnoreCase(atributo.getNmAtributo())) {
                                if (unicoAtrSimilar.getValorSimilar().containsKey(valo2r)) {

// recupera a quantidade de ocorrências do valor aceito para utilizar no cálculo.
Integer qtdValoresSimilares = somaQtdSimilaresTotal(unicoAtrSimilar.getValorSimilar().values());

// recupera a porcentagem do peso que será aplicada no cálculo.
BigDecimal vlPorcentagemSimilar = new
BigDecimal(ParametroDAO.buscarValorParametro("recruta.valor.peso.atributo.similar"));

// formula para calcular a pontuação do valor aceito.
BigDecimal pontuacao = new BigDecimal(unicoAtrSimilar.getValorSimilar().get(valo2r))
.divide(new BigDecimal(qtdValoresSimilares), 3, RoundingMode.HALF_UP)
.multiply(atributo.getNrPeso()).multiply(vlPorcentagemSimilar)
.divide(new BigDecimal(100), 3, RoundingMode.HALF_UP));

                                somaSimilaridade = somaSimilaridade.add(pontuacao);
                                populaSimilarUnico(similarUnico, atributo, valor, valo2r,
"Similar", pontuacao);
                                break;
                            }
                        }
                    }
                }
            }
        } else if (tipoValor == TipoComparacaoEnum.IGUAL_OU_MAIOR) {
            BigDecimal valorDecimal = new BigDecimal(valor);

```

```

        BigDecimal valorDecimal2 = new BigDecimal(valor2r);
        if (valorDecimal2.compareTo(valorDecimal) >= 0) {
            somaSimilaridade = somaSimilaridade.add(atributo.getNrPeso());
            populaSimilarUnico(similarUnico, atributo, valor, valor2r, "Igual ou
maior", atributo.getNrPeso());
        }
        } else if (tipoValor == TipoComparacaoEnum.IGUAL_OU_MENOR) {
            BigDecimal valorDecimal = new BigDecimal(valor);
            BigDecimal valorDecimal2 = new BigDecimal(valor2r);
            if (valorDecimal2.compareTo(valorDecimal) <= 0) {
                somaSimilaridade = somaSimilaridade.add(atributo.getNrPeso());
                populaSimilarUnico(similarUnico, atributo, valor, valor2r, "Igual ou
menor",
atributo.getNrPeso());
            }
        }
        if (similarUnico == null || similarUnico.getNmAtributo() == null) {
            populaSimilarUnico(similarUnico, atributo, valor, valor2r, "Não encontrado",
BigDecimal.ZERO);
        }
        listSimilares.add(similarUnico);
    }
}
return somaSimilaridade;
}
}

```

Fonte: elaborado pelo autor.

Como pode ser visto no Quadro 6, é percorrido cada Atributo da lista `listAtributos` para executar o cálculo da similaridade, onde é feito o uso do método `getAtributoMethodName` para buscar o valor dos atributos que serão utilizados na comparação. Quando o atributo possui tipo de comparação `Igual`, é verificado se o valor do objeto2 (Candidato) é igual ao valor do objeto1 (Vaga), ganhando o valor da pontuação total, mas caso não sejam iguais, é verificado se o valor do objeto2 está presente na `listAtributosSimilares` que contém os valores possíveis de serem aceitos, carregados com base nas soluções passadas.

Quando a comparação do atributo é definida como `IGUAL_OU_MAIOR` ou `IGUAL_OU_MENOR`, não são utilizadas a função de valores aceitos, pois nessa comparação são numéricos e não textuais, dessa forma, caso atenda a condição nesses dois casos é atribuído o peso total do atributo. Os valores aceitáveis das experiências passadas para o cálculo da similaridade, possuem peso diferente em comparado com o peso definido nos atributos, isso pois a pontuação dos valores aceitáveis é calculada levando a quantidade de vezes que o valor é aceitável. A Figura 21 detalha a fórmula usada para o cálculo da pontuação quando o valor do objeto2 é encontrado nos valores similares aceitos.

Figura 21 - Formula para cálculo da similaridade

$$P = \frac{qps}{qts} \times \left( pa \times \frac{pp}{100} \right)$$

**P** = pontuação  
**qps** = quantidade de ocorrência da palavra aceita  
**qts** = quantidade total de palavras aceitas para o atributo  
**pa** = pontuação do atributo  
**pp** = percentual da pontuação

Fonte: elaborado pelo autor.

A fórmula apresentada na Figura 21 representa as variáveis e operações para calcular a pontuação dos atributos similares demonstrado anteriormente no Quadro 6. Para formular o valor final do peso são consideradas a quantidade de vezes que o valor é encontrado nas soluções consultadas do banco de dados dividindo pelo quantidade total das palavras aceitas encontradas para o atributo da comparação, multiplicado pelo valor da porcentagem configurada por parâmetro para o atributo.

### 3.3.1.2 LinkedIn

Ao executar o *login* no LinkedIn através do link disponível na tela de cadastro do candidato, é disparada a função para popular os campos de cadastro com as informações do perfil do candidato. No Quadro 7 é apresentada a função JavaScript responsável por popular os campos em tela com as informações recuperadas do perfil do LinkedIn.

Quadro 7 - Função para recuperar dados do perfil

```

function OnLinkedInAuth() {
  IN.API.Profile("me").fields(["firstName", "lastName", "emailAddress", "positions"])
  .result(function(result) {
    var dataFim;
    var dataInicio;
    var nm_candidato = result.values[0].firstName + ' ' + result.values[0].lastName;
    $("#id$=nome_completo").val(nm_candidato);
    $("#id$=email_contato").val(result.values[0].emailAddress);
    if(result.values[0].positions.values[0].startDate != null){
      dataInicio = '01';
      dataInicio += '/' + result.values[0].positions.values[0].startDate.month ;
      dataInicio += '/' + result.values[0].positions.values[0].startDate.year ;
      $("#id$=exp_dt_entrada_input").val(dataInicio);
    }
    if(result.values[0].positions.values[0].endDate != null){
      dataFim = '01';
      dataFim += '/' + result.values[0].positions.values[0].endDate.month ;
      dataFim += '/' + result.values[0].positions.values[0].endDate.year ;
      $("#id$=exp_dt_saida_input").val(dataFim);
    }
    $("#id$=exp_nm_empresa").val(result.values[0].positions.values[0].company.name);
    $("#id$=exp_nm_cargo").val(result.values[0].positions.values[0].title);
    closeSession();
  });
}

```

Fonte: elaborado pelo autor.

Conforme apresentado no Quadro 7, é possível verificar a busca das informações `firstName`, `lastName`, `emailAdress` e `positions`, referentes ao perfil do usuário. Após efetuar a autenticação no LinkedIn, o objeto `positions` retorna as experiências profissionais do perfil, onde com a permissão `r_basicprofile`, retorna somente a experiência atual.

### 3.3.2 Operacionalidade da implementação

Neste tópico serão apresentadas as telas responsáveis pelo cadastro das vagas e candidatos, assim como as telas para configuração e parâmetros e atributos. Também será apresentada a utilização da tela responsável por executar a similaridade entre as vagas e candidatos.

#### 3.3.2.1 Acesso e LinkedIn

Através da página de *login* é possível ter acesso a duas áreas da aplicação, uma dedicada ao usuário candidato, que deseja efetuar o cadastro de seu currículo na aplicação, e outra acessada pelo usuário administrador, destinada ao profissional de RH para cadastro, alteração e manipulação das vagas e candidatos presentes na aplicação. A Figura 22 apresenta a página de *login*, onde é possível selecionar a área de acesso.

Figura 22 - Página de *login*

Área do Candidato		Acesso Restrito	
CPF:	<input type="text"/>	Login:	<input type="text"/>
E-mail	<input type="text"/>	Senha:	<input type="text"/>
<input type="button" value="Acessar"/>	<input type="button" value="Novo Currículo"/>	<input type="button" value="Entrar"/>	

Fonte: elaborada pelo autor.

Para acessar com perfil candidato clicar em `Novo Currículo`, onde será apresentado a tela para cadastrar as informações do currículo do candidato. Caso o candidato já tenha executado seu cadastro anteriormente é possível informar o `CPF` e o `E-mail`, e clicar em `Acessar`, dessa forma será carregado os dados já cadastrados, permitindo a edição de seus dados. A Figura 23 apresenta a tela de cadastro do currículo acessando pelo perfil do candidato.

Figura 23 - Página para o candidato cadastrar seu currículo

**Recruta-IA**

Curriculo

Dados Pessoais Contatos Endereços Formações Experiências Idiomas

Faça login no LinkedIn para facilitar seu cadastro!

Sign in with LinkedIn

Nome Completo

CPF

Data Nascimento

Novo Atualizar

Fonte: elaborada pelo autor.

Conforme apresentado na Figura 23, é possível efetuar o *login* no LinkedIn para auxiliar no preenchimento dos dados do candidato, onde após o candidato efetuar sua autenticação é possível buscar informações em seu perfil para adicionar no seu cadastro.

### 3.3.2.2 Páginas de cadastro

A Figura 24 demonstra a página de candidato através do acesso por um usuário administrador, onde é possível cadastrar, alterar e deletar o candidato da aplicação.

Figura 24 - Página de candidatos

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

Candidatos

Dados Pessoais Contatos Endereços Formações Experiências Idiomas

Nome Completo

CPF

Data Nascimento

Listar Novo Atualizar

Lista de Candidatos

	Nome	CPF	Data Nascimento	Data Cadastro
<input type="button" value="✎"/> <input type="button" value="✕"/>	Mariana	555.555.555-55	2000-01-01	2017-09-11
<input type="button" value="✎"/> <input type="button" value="✕"/>	João	111.111.111-11	1993-02-01	2017-09-11
<input type="button" value="✎"/> <input type="button" value="✕"/>	Eduardo	987.654.321-00	1984-05-11	2017-09-11
<input type="button" value="✎"/> <input type="button" value="✕"/>	Gabriel	888.888.888-88	1992-02-21	2017-09-11

Fonte: elaborada pelo autor.

Conforme visto na Figura 24, o cadastro do candidato disponibiliza informações sobre Dados Pessoais, Endereços, Experiências, Formações e Idiomas. As informações de

Endereços, Experiências, Formações e Idiomas também fazem parte do cadastro da vaga, conforme pode ser visto na Figura 25.

Figura 25 - Página de vagas

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

Vagas

Dados da Vaga Endereços Formações Experiências Idiomas

Descrição

Data Abertura

Situação

Listar Novo Atualizar Salvar

Lista de Vagas

	Descrição	Data Abertura	Situação
<input type="text" value="edit"/> <input type="text" value="delete"/>	Programador java	2017-09-10	1
<input type="text" value="edit"/> <input type="text" value="delete"/>	Programador java	2017-09-10	1

Fonte: elaborada pelo autor.

As informações de Endereços, Experiências, Formações e Idiomas são representadas por listas de dados, onde o usuário pode deixar sem informação ou adicionar vários registros. A Figura 26 apresenta a aba de Experiências do candidato, onde pode ser visualizado a lista dos trabalhos anteriores, permitindo adição, edição e exclusão de dados.

Figura 26 - Página de experiências do candidato

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

Candidatos

Dados Pessoais Contatos Endereços Formações Experiências Idiomas

Empresa

Área

Cargo

Data Inicio

Data Saída

Novo Alterar Salvar

	Empresa	Área	Cargo	Data Inicio	Data Saída
<input type="text" value="edit"/> <input type="text" value="delete"/>	Empresa 1	1	testes	2015-11-01	2016-06-14
<input type="text" value="edit"/> <input type="text" value="delete"/>	Empresa2	1	programador java	2016-08-17	2017-11-14

Fonte: elaborada pelo autor.

### 3.3.2.3 Parâmetros e configurações

Para configuração da aplicação existem duas páginas que permitem a edição das informações pelo usuário, sendo elas a página de Atributos e a página de Parâmetros. A Figura 27 apresenta a página de atributos.

Figura 27 - Página de atributos para similaridade

Lista de Atributos			
Nome Entidade	Nome Atributo	Valor Peso	
experiencia	area	5.00	✎
experiencia	cargo	5.00	✎
experiencia	meses	1.00	✎
endereço	cidade	5.00	✎
endereço	estado	5.00	✎
endereço	bairro	5.00	✎
endereço	rua	5.00	✎
endereço	cep	5.00	✎
formacao	nivel	5.00	✎
formacao	area	5.00	✎
formacao	descricao	5.00	✎
formacao	meses	5.00	✎
idioma	lingua	5.00	✎

Fonte: elaborada pelo autor.

Conforme pode ser visto na Figura 27, a página apresenta a lista completa dos atributos utilizados para o cálculo da similaridade entre dois objetos. A partir dessa tela é possível visualizar a entidade, atributo e peso de cada atributo, sendo possível a edição do peso pelo usuário de RH.

A segunda tela de configuração apresenta os parâmetros disponíveis na aplicação. Através dessa tela é possível visualizar o nome e descrição dos parâmetros, assim como editar seu valor. A Figura 28 apresenta a tela de parâmetros do Recruta-IA.



Figura 28 - Página de parâmetros

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

Parâmetros

Lista de Parâmetros			
Nome	Descrição	Valor	
recruta.qtd.vagas.similares	Quantidade de vagas carregadas para comparação com a atual	2	↕
recruta.qtd.candidatos.similares	Quantidade de candidatos similares a vaga atual	2	↕
recruta.data.vaga.inicio.similaridade	Data para inicio do carregamento das vagas. Exemplo: dd/MM/yyyy)	01/01/2001	↕
recruta.data.candidato.inicio.similaridade	Data para inicio do carregamento dos candidatos. Exemplo: dd/MM/yyyy)	01/01/2001	↕
recruta.valor.peso.atributo.similar	Porcentagem do valor similar de acordo com a quantidade de ocorrencias. 0 até 100	50	↕

Fonte: elaborada pelo autor.

Conforme visto na Figura 28, a aplicação disponibiliza cinco parâmetros que podem ser configurados pelo usuário, sendo que o valor configurado influencia diretamente no resultado da similaridade entre a vaga e o candidato.

### 3.3.2.4 Similaridade

A página de similaridade permite que o usuário verifique qual candidato disponível na aplicação pode preencher a vaga em aberto. É nesta tela que ocorre a utilização de IA para executar a comparação entre a vaga e candidato. A Figura 29 apresenta a listagem da vagas disponíveis onde já se executou a similaridade para buscar os candidatos que melhor atendem os requisitos da vaga.

Figura 29 - Página de similaridade

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

Similaridade

Listar

Lista de Vagas disponíveis			
	Descrição	Data Abertura	Situação
⊞	Programador delphi	2017-09-10	1
⊞	Programador delphi	2017-08-04	1

Calc. Similar

Lista de Candidatos possíveis			
	Pontuação	Nome Candidato	CPF
✓ ⊞	16.00	Eduardo	987.654.321-00
✓ ⊞	14.75	Gabriel	888.888.888-88
✓ ⊞	11.00	Ana	222.222.222-22

Fonte: elaborada pelo autor.

Conforme visto na Figura 29, após executar o cálculo da similaridade, é apresentada a lista dos candidatos que melhor atenderam os atributos da vaga. Para cada candidato dessa lista é possível visualizar a pontuação da comparação com a vaga ou executar o vínculo do candidato com a vaga. A Figura 30 apresenta o modal com os detalhes da comparação da vaga com o candidato.

Figura 30 - Detalhes da pontuação

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

**Detalhes da Solução**

Solução

Candidato 4 Vaga 3 Pontuação 14.75

Lista de similaridade						
Entidade	Atributo	Comparacao	Vaga	Candidato	Peso	Pontuação
endereco	cidade	Similar	Blumenau	Indaial	5.00	2.50
endereco	estado	Igualdade	Santa Catarina	Santa Catarina	5.00	5.00
experiencia	area	Igualdade	1	1	5.00	5.00
experiencia	cargo	Similar	programador delphi	programador java	5.00	1.25
experiencia	meses	Igual ou maior	24	36	1.00	1.00

<input checked="" type="checkbox"/>		14.75	Gabriel	888.888.888-88
<input checked="" type="checkbox"/>		11.00	Ana	222.222.222-22

Fonte: elaborada pelo autor.

Conforme visto na Figura 30, o detalhamento da pontuação apresenta cada atributo verificado durante o cálculo da similaridade entre a vaga e o candidato, exibindo dados como a entidade do atributo, o atributo, o tipo de comparação executada, a informação da vaga, a informação do candidato, o peso do atributo e a pontuação adicionada pela comparação.

Caso seja clicado no botão para vincular o candidato com a vaga, o vínculo é efetuado e apresentado na página de soluções, conforme apresenta a Figura 31.

Figura 31 - Página de soluções

**Recruta-IA**

Home Candidatos Vagas Similaridade Soluções Atributos Parâmetros

**Soluções**

Listar

Lista de Soluções			
	Vaga	Candidato	Data Solução
	1	1	2017-08-08
	2	2	2017-07-08
	3	4	2017-11-14

Fonte: elaborada pelo autor.

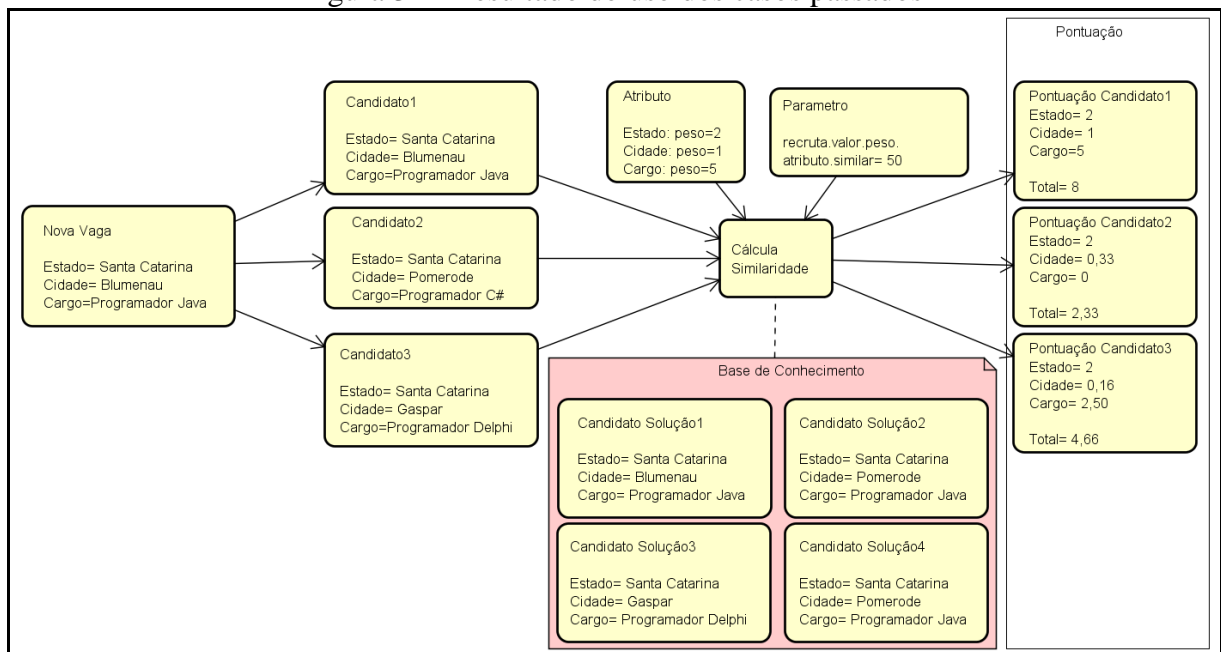
A partir do vínculo entre um candidato e uma vaga, a solução gerada pode ser usada para resolver as próximas comparações. Caso seja desfeito a solução através da página de soluções, o candidato e a vaga voltam a ficar disponíveis para comparação.

### 3.4 ANÁLISE DOS RESULTADOS

A solução desenvolvida apresentou uma aplicação Web utilizando inteligência artificial para análise de currículos, auxiliando o processo de tomada de decisão do setor de RH durante o recrutamento e seleção de novos funcionários.

Em testes simulando um ambiente com vagas em aberto, candidatos disponíveis e soluções já cadastradas no banco de dados, foi possível executar a similaridade utilizando as contratações anteriores, para pontuação dos candidatos, caso os mesmos atendam as características solicitadas na vaga, ou possuam os dados de contratações anteriores similares a vaga atual. Para isso são utilizadas técnicas de RBC, permitindo a utilização de experiências anteriores para solução de problemas atuais. A Figura 32 demonstra um exemplo simples da utilização dos casos anteriores na aplicação Recruta-IA para solucionar a nova vaga, verificando as características dos candidatos disponíveis em relação à pontuação do endereço.

Figura 32 - Resultado do uso dos casos passados



Fonte: elaborado pelo autor.

Conforme visto na Figura 32, além da comparação direta entre as características do Estado e Cidade presente na Nova Vaga e nos candidatos disponíveis Candidato1, Candidato2 e Candidato3, são utilizadas as soluções anteriores presentes na Base de Conhecimento para buscar os valores aceitáveis dos atributos da similaridade. Para o exemplo

apresentado, quando o `Candidato1` possui a mesma informação solicitada na `Nova Vaga`, é adicionado o peso total do atributo `Cidade` para o somatório da similaridade, diferente do `Candidato2` que possui a `Cidade` diferente da solicitada na `Nova Vaga`, mas presente na `Base de Conhecimento`, dividindo o peso do atributo pela quantidade de ocorrências e a porcentagem definida nos parâmetros da aplicação.

O uso dos casos passados para pontuação no processo de similaridade, utiliza a experiência do profissional de RH aplicada nas soluções anteriores para resolver o caso atual, criando uma lista de valores aceitáveis para os candidatos atuais, com base nas vagas já preenchidas similares a nova vaga.

Para cadastro dos candidatos, a aplicação disponibiliza de uma integração com a API do LinkedIn, onde é possível a consulta de alguns dados após o candidato executar o *login* na aplicação. Durante a implementação desta funcionalidade, foi encontrada uma dificuldade para buscar as informações completas presentes no perfil do usuário, onde o mesmo é possível somente após a solicitação e aprovação de uma parceria da aplicação do desenvolvedor com o LinkedIn, onde habilita a permissão para busca completa dos dados no perfil. A solicitação de parceria foi enviada, mas recusada pelos analistas do LinkedIn. Dessa forma, o desenvolvimento foi continuado através da permissão básica, sendo um acesso mais restrito aos dados do candidato.

O Quadro 8 apresenta uma comparação entre as características dos trabalhos correlatos e o trabalho desenvolvido.

Quadro 8 - Características dos trabalhos correlatos e do trabalho proposto

Característica/Trabalho Relacionados	Sistema RBC para Recrutamento e Seleção de Profissionais (SILVA, 2004)	Desenvolvimento de sistema online para recrutamento de pessoas (PREIS, 2013)	SIIEE - Sistema de Raciocínio baseado em casos para análise de perfil de candidatos a vagas de estágio (PANCERI, 2012)	Recruta-IA: trabalho proposto
Técnica de inteligência artificial utilizada	RBC	Nenhuma	RBC	RBC
Usuários com acesso à aplicação	Profissionais do RH da empresa específica	Profissionais do RH e candidatos a vagas	Profissionais do RH e candidatos a vagas de estágio	Profissionais do RH e candidatos à busca de emprego
Possui versão Web	Não	Sim	Não	Sim
Linguagem de programação	Delphi	C#	Java	Java
Foco do recrutamento	Candidatos interessados nas vagas de uma empresa específica	Candidatos para vagas cadastradas no site por diversas empresas	Alunos para estágios em empresas	Foco específico na análise de candidatos que se encaixem nas vagas cadastradas
Integração com Aplicação externa	Não	Não	Não	Linkedin

Fonte: elaborado pelo autor.

Conforme demonstrado no Quadro 8, os trabalhos correlatos que utilizam inteligência artificial fazem o uso de técnicas de RBC para análise dos currículos, assim como o Recruta-IA. Para isso aplicam algoritmos de similaridade e definem pesos para as características que são analisadas entre as comparações, fazendo uso das soluções passadas para encontrar a solução do caso atual.

A principal diferença entre o trabalho desenvolvido e os correlatos ocorre no desenvolvido ser utilizado a linguagem Java possuindo uma versão Web, além de ser o único a utilizar a API do LinkedIn para buscar as informações do candidato. O RBC aplicado a busca de características similares e a apresentação da pontuação final entre a comparação da vaga e o candidato também facilitam na análise pelo profissional de RH, além de tornar a tomada de decisão mais ágil, listando somente os candidatos que melhor atendem as características presentes na vaga.

Para o cadastro do candidato, foi disponibilizado uma interface de acesso facilitando o contato do candidato com a empresa, durante a execução do próprio cadastro, o candidato pode usar as informações presentes no LinkedIn para preencher os dados necessário do formulário de cadastro. Dessa forma, o candidato executando seu próprio cadastro, poupa tempo do profissional do RH, que não precisa mais executar o cadastro manualmente no sistema, sendo que o mesmo já se encontra disponível para as análises.

## 4 CONCLUSÕES

O objetivo principal foi disponibilizar uma aplicação Web que auxilie o setor de RH no recrutamento de candidatos para preencherem as vagas em aberto, utilizando IA na análise e comparação. O objetivo proposto foi alcançado, onde se fez o uso de técnicas do RBC para auxiliar na tomada de decisão.

Ao final da implementação foi possível desenvolver uma aplicação Web, atendendo o objetivo geral e os objetivos específicos, através de uma aplicação que possibilita o acesso do candidato e do profissional de recursos humanos, sendo possível o cadastro de novos candidatos e vagas, configuração dos pesos para o cálculo da similaridade e o acesso ao LinkedIn para buscar informações no perfil do candidato, facilitando o preenchimento do formulário de cadastro.

As técnicas de RBC foram usadas para buscar valores que podem ser aceitos na comparação entre o candidato e a vaga, onde os valores aceitos representam experiências passadas do administrador de RH em outras contratações para vagas que são semelhantes a vaga atual. Com esse conhecimento é possível a pontuação de características diferentes das solicitadas na vaga, mas já presentes em outras contratações.

O cálculo da similaridade entre as características do candidato e da vaga em aberto, possibilitou uma análise ágil e detalhada de quais candidatos melhor atendem as condições exigidas pela vaga. Dessa forma, poupando o profissional de RH da análise de todos os currículos recebidos. Comparado com o processo manual, o uso de IA para o processo de avaliação dos candidatos torna o processo mais rápido e preciso, visto que a pontuação é atribuída quando as características do candidato são as mesmas solicitadas na vaga atual, ou similares a características aceitas em contratações anteriores.

O desenvolvimento da aplicação teve grande contribuição ao aprendizado acadêmico do autor e demonstrou ser possível o uso da IA para auxiliar no processo de recrutamento e seleção, assim como na tomada de decisão executada pelo RH.

### 4.1 EXTENSÕES

Como sugestões para trabalhos futuros propõem-se:

- a) aprimorar os cadastros das vagas e candidatos;
- b) desenvolver um Webservice para integrar informações adicionais de currículos vindos de outras aplicações, disponibilizando uma interface para comunicação entre as aplicações.
- c) utilizar a permissão completa do LinkedIn para buscar os dados do candidato.

## REFERÊNCIAS

- AAMODT, A.; PLAZA, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. In: ARTIFICIAL INTELLIGENCE COMMUNICATIONS, IOS Press, 7, 1994, Bellaterra. **Proceedings...** Bellaterra: IIIA-CSIC, 1994. p. 39–59.
- CARVALHO, Alexey. Revista de Ciências Gerencias. **Raciocínio baseado em casos aplicado ao processo decisório**. São Paulo, v. 12, n. 16, p. 25-36, 2008.
- CHIAVENATO, Idalberto. **Recursos Humanos**. 7. ed. São Paulo: Editora Atlas, 2002. 631 p.
- G1, **Concorrência por vagas de emprego é a maior em 7 anos, diz pesquisa**. Disponível em; < <http://g1.globo.com/economia/concursos-e-emprego/noticia/2016/08/concorrenca-por-vagas-de-emprego-e-maior-em-7-anos-diz-pesquisa.html> >. Acessado em: 19 mar. 2017.
- KIPPER, Liane M.; FROZZA Rejane; URNAU Eduardo. Modelagem de um sistema para apoio à tomada de decisão com uso de técnicas de raciocínio baseado em casos. In: TECNOLOGIA, 18, 2014, Santa Cruz do Sul. **Anais...** Santa Cruz do Sul: UNISC, 2014. p. 49-59.
- LINKEDIN. [S.l.], 2017. Disponível em: < <https://developer.linkedin.com> >. Acesso em: 7 novembro 2017.
- MONARD, Maria C.; BARANAUKAS, José A. **Aplicações de Inteligência Artificial: Uma Visão Geral**. São Carlos: Instituto de Ciências Matemáticas e de Computação de São Carlos, 2000.
- PANCERI, Sabrina. **SIIEE – Sistema de raciocínio baseado em casos para análise de perfil de candidatos a vagas de estágio**. 2012. 101. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Vila Velha, Espírito Santo.
- PONTELO, Juliana F.; CRUZ, Lucineide. A. M. **Gestão de pessoas: manual de rotinas trabalhistas**. Brasília: Editora Senac; 2012. 360 p.
- PREIS, Giovane. **Desenvolvimento de sistema online para recrutamento de pessoas**. 2013. 131. Trabalho de Conclusão de Curso (Tecnólogo em Gestão de Tecnologia) – Faculdade de Itapiranga (FAI), Itapiranga.
- RIBEIRO, Antônio. **Gestão de pessoas**. São Paulo: Editora Saraiva; 2005. 310 p.
- RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 2. Ed. Rio de Janeiro: Campos, 2004.
- SANTOS, Fernando Chagas; DE CARVALHO, Cedric Luiz. **Aplicação da Inteligência Artificial em Sistemas de Gerenciamento de Conteúdo**. Goiás: UGF, 2008. 16 p.
- SILVA, Rondinele. Sistema RBC para Recrutamento e Seleção de Profissionais. In: CONGRESSO BRASILEIRO DE COMPUTAÇÃO, 4, 2004, Itajaí. **Anais...** Itajaí: UNIVALI, 2004. p. 294-300.
- WANGENHEIM, Christiane Gresse Von, WANGENHEIM, Aldo Von. **Raciocínio Baseado em Casos**. Barueri: Manole, 2003. 293 P.

## APÊNDICE A – Descrição dos Casos de Uso

Nesta seção são apresentados os detalhes dos casos de uso, com descrição, pré-condição e cenário. Os quadros a seguir representam as ações possíveis de serem executadas pelo Administrador de RH e o Candidato, ao acessarem a aplicação.

Quadro 9 - Caso de uso: Cadastrar candidatos

Número	UC01
Caso de Uso	Cadastrar candidatos
Ator	Administrador de RH
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário entra na aplicação.</li> <li>2. O usuário acessa o menu de Candidatos</li> <li>3. O usuário clica na opção de Novo</li> <li>4. O usuário informa os dados sobre o endereço, experiências profissionais, formações e idiomas do candidato.</li> <li>5. O usuário clica em Salvar para completar o cadastro do candidato.</li> </ol>

Fonte: elaborada pelo autor.

Quadro 10 - Caso de uso: Cadastrar vaga

Número	UC02
Caso de Uso	Cadastrar vagas
Ator	Administrador de RH
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário entra na aplicação.</li> <li>2. O usuário acessa o menu de Vagas</li> <li>3. O usuário clica na opção de Nova</li> <li>4. O usuário informa os dados sobre o endereço, experiências profissionais, formações e idiomas necessários para vaga.</li> <li>5. O usuário clica em Salvar para completar o cadastro da vaga.</li> </ol>

Fonte: elaborada pelo autor.

Quadro 11 – Caso de uso: Configurar o peso dos atributos

Número	UC03
Caso de Uso	Configurar o peso dos atributos
Ator	Administrador de RH
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário entra na aplicação.</li> <li>2. O usuário acessa o menu de Atributos</li> <li>3. É apresentado a lista de atributos disponíveis para o cálculo da similaridade</li> <li>4. O usuário clica em Editar na linha do atributo desejado</li> <li>5. O usuário informa o novo valor.</li> <li>6. O usuário clica em Salvar, alterando o valor do atributo desejado.</li> </ol>

Fonte: elaborada pelo autor.



Quadro 12 – Caso de uso: Executar a similaridade entre vagas e candidatos

Número	UC04
Caso de Uso	Executar a similaridade entre vagas e candidatos
Ator	Administrador de RH
Pré-condição	Existir vagas em aberto e candidatos disponíveis no bando de dados
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário entra na aplicação.</li> <li>2. O usuário acessa o menu de Similaridade</li> <li>3. É apresentado a lista de vagas disponíveis para o calcula da similaridade</li> <li>4. O usuário seleciona a vaga desejada para executar a similaridade</li> <li>5. A aplicação busca os candidatos disponíveis no banco é executa a similaridade com a vaga, retornando à pontuação de cada candidato.</li> </ol>

Fonte: elaborada pelo autor.

Quadro 13 - Caso de uso: Consultar o relatório da similaridade entre candidato e vaga

Número	UC05
Caso de Uso	Consultar o relatório da similaridade entre candidato e vaga
Ator	Administrador de RH
Pré-condição	Executar o UC04
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário seleciona um candidato da lista dos candidatos similares a vaga.</li> <li>2. O sistema apresenta um modal com o detalhamento da pontuação entre a vaga e o candidato.</li> <li>3. O modal pode ser fechado voltado para página de similaridade.</li> </ol>

Fonte: elaborada pelo autor.

Quadro 14 – Caso de uso: Vincular a vaga com o candidato

Número	UC06
Caso de Uso	Executar a similaridade entre vagas e candidatos
Ator	Administrador de RH
Pré-condição	Executar o UC04
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário na opção de vincular em um dos candidatos similares retornados do cálculo.</li> <li>2. É gerado o vínculo entre o candidato e a vaga gravando no banco de dados a nova solução.</li> <li>3. A página é recarregada com as vagas que ainda estão em aberto.</li> </ol>

Fonte: elaborada pelo autor.

Quadro 15 – Caso de uso: Executar o próprio cadastro

Número	UC07
Caso de Uso	Executar o próprio cadastro
Ator	Candidato
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário acessa a aplicação.</li> <li>2. O usuário clica no botão Novo Currículo.</li> <li>3. É apresentado a tela de cadastro do candidato sem a listagem</li> <li>4. O usuário preenche os campos solicitados do cadastro.</li> <li>5. Ao salvar o registro, fica disponível no banco de dados para comparação com as vagas em aberto.</li> </ol>

Fonte: elaborada pelo autor.

Quadro 16 – Caso de uso: Buscar informações no LinkedIn para o cadastro

Número	UC08
Caso de Uso	Buscar informações no LinkedIn para o cadastro
Ator	Candidato
Pré-condição	Possuir uma conta no LinkedIn
Cenário Principal	<ol style="list-style-type: none"> <li>1. O usuário acessa a aplicação.</li> <li>2. O usuário clica no botão Novo Currículo.</li> <li>3. É apresentado a tela de cadastro onde possibilita fazer <i>login</i> no LinkedIn.</li> <li>4. O usuário informa seu e-mail e senha no modal de autenticação da API.</li> <li>5. Após autenticado, são buscados os dados no perfil do candidato para preenchimento das informações solicitadas.</li> <li>6. O usuário preenche as informações faltantes e clica em salvar, finalizando a geração de seu cadastro.</li> </ol>

Fonte: elaborada pelo autor.

## APÊNDICE B – Dicionário de dados

A seguir é apresentado o Dicionário de dados do MER sobre as principais tabelas da aplicação.

Quadro 17 - Dicionário da tabela *candidato*

<b>Tabela: candidato</b>	
Nesta tabela são cadastrados os candidatos que serão utilizados para preencherem as vagas.	
<b>Coluna</b>	<b>Significado</b>
cd_candidato	Chave primária da tabela
nm_candidato	Nome do candidato
dt_nascimento	Data de nascimento
dt_cadastro	Data de cadastro
nr_cpf	Número do CPF
fg_selecionado	Identificação do uso do candidato

Fonte: elaborada pelo autor.

Quadro 18 - Dicionário da tabela *vaga*

<b>Tabela: vaga</b>	
Nesta tabela são cadastrados as vagas abertas pela empresa.	
<b>Coluna</b>	<b>Significado</b>
cd_vaga	Chave primária da tabela
ds_vaga	Descrição da vaga
dt_abertura	Data de abertura
cd_situacao	Identificação do uso da vaga

Fonte: elaborada pelo autor.

Quadro 19 - Dicionário da tabela *solucao*

<b>Tabela: solucao</b>	
Nesta tabela são cadastrados os vínculos entre as vagas e candidatos de contratações anteriores.	
<b>Coluna</b>	<b>Significado</b>
cd_candidato	Chave estrangeira da tabela <i>candidato</i>
cd_vaga	Chave estrangeira da tabela <i>vaga</i>
nr_pontuacao	Pontuação da solução
fg_escolhido	Identificação de seleção do vínculo
dt_solucao	Data da solução

Fonte: elaborada pelo autor.

Quadro 20 - Dicionário da tabela *parametro*

<b>Tabela: parametro</b>	
Nesta tabela são armazenadas as configurações utilizadas para similaridade.	
<b>Coluna</b>	<b>Significado</b>
codigo	Chave primária da tabela
nome	Nome do parâmetro
descricao	Descrição da utilização do parâmetro
valor	Valor do parâmetro

Fonte: elaborada pelo autor.

Quadro 21 - Dicionário da tabela atributo

<b>Tabela: atributo</b>	
Nesta tabela são armazenados os atributos utilizados para comparação entre a vaga e o candidato.	
<b>Coluna</b>	<b>Significado</b>
cd_atributo	Chave primária da tabela
nm_atributo	Nome do atributo
nr_peso	Valor do peso para pontuação
nm_entidade	Nome da entidade referente ao atributo

Fonte: elaborada pelo autor.