



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
CAMPUS LUIZ MENEGHEL

RODRIGO ANGELINI FERRARI

**UTILIZAÇÃO DE RACIOCÍNIO BASEADO EM CASOS
PARA A SELEÇÃO DE PESSOAS EM PROJETOS DE
DESENVOLVIMENTO DE SOFTWARE**

Bandeirantes

2017

RODRIGO ANGELINI FERRARI

**UTILIZAÇÃO DE RACIOCÍNIO BASEADO EM CASOS
PARA A SELEÇÃO DE PESSOAS EM PROJETOS DE
DESENVOLVIMENTO DE SOFTWARE**

Monografia apresentada à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito parcial para aprovação na disciplina Projeto Final II do Curso de Sistemas de Informação.

Orientador: Prof. Me Glauco Carlos Silva

Bandeirantes

2017

RODRIGO ANGELINI FERRARI

**UTILIZAÇÃO DE RACIOCÍNIO BASEADO EM CASOS
PARA A SELEÇÃO DE PESSOAS EM PROJETOS DE
DESENVOLVIMENTO DE SOFTWARE**

Monografia apresentada à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito parcial para aprovação na disciplina Projeto Final II do Curso de Sistemas de Informação.

COMISSÃO EXAMINADORA

UENP – *Campus* Luiz Meneghel

UENP – *Campus* Luiz Meneghel

UENP – *Campus* Luiz Meneghel

Bandeirantes, de julho de 2017

RESUMO

A inteligência artificial é uma das formas atuais que busca auxiliar na resolução de problemas humanos. Existem dentro da IA várias técnicas que podem ser utilizadas para a resolução de novos problemas neste caso foi empregado a técnica chamada Raciocínio Baseado em Casos (RBC), esta técnica trabalha com casos concretos e não modelos matemáticos aos quais podem vir a prejudicar em relação a escolha de importância de determinada característica escolhida pelo usuário selecionador. Essa pesquisa consiste em uma proposta de ajuda em alocações de pessoas no projeto, para isso foi utilizado essa técnica de aplicação, que realiza uma visão de análise de casos com maior precisão pois leva em conta as habilidades dos indivíduos com as características necessários do projeto. O RBC leva em conta também os projetos anteriores, o que faz com que o crescimento da base seja de forma linear, isto é, a base alimenta-se de acordo com os novos projetos e características dos membros da equipe.

Palavras-chave: Raciocínio Baseado em Caso, Inteligência Artificial, G. Wangenheim

ABSTRACT

Artificial intelligence is one of the current forms that seeks to assist in the resolution of human problems. There are several techniques within AI that can be used to solve new problems. In this case, the technique called Case-Based Reasoning (RBC) has been employed. This technique works with concrete cases rather than mathematical models that may be detrimental to Choice of importance of a particular feature chosen by the user picker. This research consists of a proposal to help with the allocation of people in the project, using this application technique, which performs a view of case analysis with greater accuracy because it takes into account the abilities of individuals with the necessary characteristics of the project. The RBC also takes into account the previous projects, which means that the growth of the base is linear, that is, the base feeds according to the new projects and characteristics of the employees.

Keywords: Case Based Reasoning, Artificial Intelligence, G. Wangenheim

LISTA DE FIGURAS

Figura 1 – Investimento no Mercado de TI.....	7
Figura 2 – Ciclo de Vida RBC Adaptado.....	13
Figura 3 – Distância Nearest Neighbor.....	18
Figura 4 – Ranking de Busca Por Linguagem.....	23
Figura 5 – Java Scene Builder.....	25
Figura 6 – O modelo MVC.....	26
Figura 7 – Ranking de Popularidade SGBDs.....	27
Figura 8 – Utilização do Hibernate por Grandes Empresas.....	29
Figura 9 – Níveis de Programador.....	32
Figura 10 – Listagem dos Fatores.....	38
Figura 13 – Listagem dos Pesos.....	44

LISTA DE QUADROS

Quadro 1 – Características Humanas.....	36
Quadro 2 – Características Técnicas.....	37

Sumário

1 Introdução	7
1.1 Contextualização	7
1.2 Formulação e Escopo do Problema	8
1.3 Justificativas	9
1.4 Objetivos	10
1.4.1 Objetivo Geral	10
1.4.2 Objetivos Específicos	10
1.5 Organização do Trabalho	11
2 Fundamentação teórica.....	11
2.1 Gerência de Projetos.....	11
2.1.1 Perfil de Competência	11
2.1.2 Competências individuais em Gerência de Projetos	12
2.2 Raciocínio Baseado em Casos	12
2.2.1 Ciclo do RBC.....	13
2.3 Recuperação de informação	16
2.4 Recuperação de casos.....	16
2.4.1 Distância Euclidiana.....	17
2.4.2 Distância Euclidiana Média.....	18
2.4.3 Distância de Mahalanobis – D^2	18
2.5 Etapas de Elaboração	18
3 Desenvolvimento.....	20
3.1 Construção da base de casos	20
3.2 Características	21
3.2.1 Linguagem de Programação	21
3.2.2 JavaFX	24
3.2.3 Padrão MVC.....	25
3.2.4 Banco de Dados.....	25
3.2.5 Hibernate.....	27
3.2.6 CodeIgniter.....	28
3.2.7 AngularCLI	29
3.2.8 Servidores de Aplicação e WebContainer	29
3.2.9 Relatórios	30
3.2.10 Níveis de Programador.....	31

3.2.11 Nível de bilinguismo, trilinguismo ou poliglota	32
3.2.12 Certificações.....	33
3.2.13 Cloud Computing.....	35
3.3 Construção da base de casos	36
3.3.1 Construção de Tabelas	36
3.3.2 Características - Similaridades	40
3.4 Recuperação de Casos	42
3.4.1 Similaridade Local	42
3.4.2 Similaridade Global	43
3.5 Retenção	43
3.6 Ferramenta de Raciocínio Baseado em Casos	44
4 Conclusão e Trabalhos Futuros	52

1 Introdução

1.1 Contextualização

O mercado de TI passa por constante especializações em que o profissional precisa estar sempre se aprimorando, estando sempre precavido para novas tecnologias, podendo assim conseguir atuar em uma atividade de sua preferência e do qual está habituado.

Em uma área tão inovadora o profissional deve estar sujeito as mudanças para que assim esteja preparado para assumir novos projetos e tarefas. Segundo a ABES (2013)”. O mercado americano de TI representa mais de 30% do mercado mundial, com aproximadamente US\$659 bilhões. O Brasil tem um mercado 10 vezes menor, mas já aparece entre os 10 maiores, com US\$61 bilhões”. O gráfico da figura 1 abaixo mostra o investimento dos países na área de TI.

	1ª USA	429	40,2%		10ª Itália	20	1,9%
	2ª Japão	79	7,4%		11ª Holanda	19	1,9%
	3ª UK	75	7,1%		12ª Espanha	17	1,6%
	4ª Alemanha	64	6,0%		13ª Suíça	14	1,3%
	5ª França	46	4,4%		14ª Suécia	14	1,3%
	6ª Canadá	32	3,1%		15ª Rússia	12	1,2%
	7ª China	31	3,0%		16ª Índia	12	1,2%
	8ª Brasil	25	2,4%		17ª Coreia	10	1,0%
	9ª Austrália	23	2,2%		18ª ROW	127	11,9%
TOTAL US\$ 1.067 / 100%							

Figura 1 – Investimento no Mercado de TI (ABES,2014)

Em relação ao Brasil deve-se salientar o alto investimento superando grandes produtores de software do exterior. Parte desse capital brasileiro está localizado nas grandes capitais. Tendo em vista tamanho investimento, o indivíduo deve estar sempre adquirindo novas habilidades que atendem ao mercado, pois alterar técnicas de trabalho colocando novas tecnologias e metodologias além de ser um processo custoso não é um dos principais problemas, mas sim, os indivíduos que integram a equipe. (WHOLIN, 2004). A equipe deve ser montada de profissionais que atendam

ao investimento do mercado, nesse estudo foi buscado quais as tecnologias mais requisitadas no mercado atualmente.

As empresas por sua vez tem dificuldade em selecionar as pessoas para determinado tipo de projeto, buscando da melhor forma possível determinar os indivíduos para alocá-los nos projetos. Selecionar uma equipe nunca é uma tarefa fácil na área de desenvolvimento. Essa alocação além de demandar tempo, requer muita análise e comprometimento, pois se trata de um fator muito importante no desenvolvimento do projeto, falhar nessa etapa pode resultar em consequências custosas no orçamento do software (BARRETO, 2005).

Para o PMBOK (2014, p.255) :

[...] A equipe do projeto consiste das pessoas com papéis e responsabilidades designadas para completar o projeto. Os membros da equipe do projeto podem ter vários conjuntos de habilidades, atuar em regime de tempo integral ou parcial, e podem ser acrescentados ou removidos da equipe a medida que o projeto progride. [...] (Project Management Institute, 2014, p.255).

Existem diversas formas de se alocar pessoas em projetos, muitas das vezes, não basta dizer somente que se precisa de determinado membro pois ele contém as características técnicas que são necessários para participar do projeto. (PFLEEGER, 2004). Ao longo do desenvolvimento essas pessoas selecionadas trocam conhecimento podendo estar sempre se aprimorando, não somente no que se refere a empresa como metodologias de trabalho e melhora no desempenho da execução de tarefas mas em habilidades individuais (SOUZA, 2003). O objetivo dessa técnica de RBC é utilizá-la em prol do auxílio no processo de tomada de decisão de alocação na gestão de pessoas do projeto.

1.2 Formulação e Escopo do Problema

Atualmente o processo de seleção para as pessoas em um projeto é de grande importância e influência para o sucesso do desenvolvimento e conclusão do mesmo em tempo determinado. Selecionar as pessoas certas para ocuparem as vagas disponíveis pode se tornar um grande impasse. Isso só pode ser possível na avaliação

de características técnicas individuais e humanas de determinada pessoa. À vista disso pode-se surgir algumas questões como:

- Quais os melhores candidatos para alocar em determinado projeto?
- Existe algum histórico de projetos passados já realizados por aquela pessoa?
- Quais os habilidades e experiências dessa pessoa?
- Essa pessoa se encaixa em quais quesitos para esse projeto?
- Quantidade de tempo necessário para essa avaliação
- Quais os riscos de adicionar determinada pessoa no projeto?

Uma vez que a organização não possui informações dessas abordagens o projeto pode se comprometer antes mesmo de seu início, todavia uma boa estrutura de seleção é fundamental para amenizar vários problemas futuros recorrentes a má elaboração da equipe.

1.3 Justificativas

A alocação de determinada pessoa pode afetar diretamente o desenvolvimento do projeto dependendo das características e habilidades de cada pessoa, além de influenciar o projeto em alguns requisitos como:

- Confiabilidade

Muitas das vezes habilidades e técnicas individuais apesar de serem importantes não bastam para se realizar com sucesso um projeto, por isso a confiança em pessoas que estão dentro do projeto deve ser mútua. Pode-se causar perda de tempo em supervisões visto que um não confia em outro. (BIASOLI, 2016)

- Desenvolvimento

Etapa em que o software é construído. Em uma equipe bem estabelecida se torna um importante fator para se concretizar essa etapa sem muitos problemas. Fazer o desenvolvedor requisitar o máximo de suas habilidades através das atividades impostas faz com que o desenvolvimento possa fluir de modo muito produtivo (PMBOK, 2013)

- Desempenho

Os membros da equipe devem também possuir habilidades individuais que auxiliariam no desempenho de atividades alocadas para o desenvolvedor, essas habilidades evitariam diversos tipos de problemas no decorrer do desenvolvimento. Os objetivos da organização deixam de ser prioridade, ocasiona a perda de qualidade na execução das tarefas e também sua produtividade diminui consideravelmente. (MOURA, 2014)

- Riscos

Os riscos podem aumentar várias etapas do desenvolvimento caso a equipe de software não esteja bem formada. Uma das formas de diminuir os riscos do projeto é selecionar pessoas que aumentem as probabilidades de cumprir com os objetivos do projeto. (PMBOK, 2013)

1.4 Objetivos

Nas seguintes subseções serão apresentados os objetivos geral e específico que procuram ser atingidos no término do trabalho.

1.4.1 Objetivo Geral

O desenvolvimento deste trabalho tem como objetivo utilizar um RBC capaz de auxiliar na seleção de pessoas, para que possa agilizar a elaboração da equipe. A seleção foi feita na seleção de atributos humanos e características técnicas dos membros para que possa se chegar a conclusão de que o membros é apto para ingressar na equipe.

1.4.2 Objetivos Específicos

Para se alcançar o Objetivo Geral foi necessário passar por diversas etapas que foram elaboradas por (Wangenheim;Kolodner;Watson), essas etapas serão melhor exploradas no decorrer do trabalho.

- Realizar simulação de uma base de dados.
- Definição de que características da informação que serão relevantes e podem ser utilizados para a solução do problema.
- Definição de índices para os casos, para que seja possível a sua recuperação quando necessário.
- Definição dos métodos de recuperação de casos para verificação da similaridade entre os casos contidos na base e os novos problemas.
- Realizar experimentos
- Definir metodologia e validação
- Validar resultados.

1.5 Organização do Trabalho

O trabalho foi organizado da seguinte forma. Na seção 2 é apresentada a fundamentação teórica necessária para o trabalho. Na seção 3 é exibido o desenvolvimento da solução proposta. Por fim está a conclusão que foi possível obter com o desenvolvimento do projeto.

2 Fundamentação teórica

2.1 Gerência de Projetos

Segundo (MENGATO, 2014)

“Os gerentes de projetos precisam lidar com diversos tipos de informações diárias, e na maioria dos casos, eles trabalham em vários projetos ao mesmo tempo. Por isso é fundamental o uso de ferramentas que auxiliam nas tarefas de gerenciamento.”

Dessa forma é indispensável a presença de uma boa equipe para a realização de um bom trabalho, sendo necessário ao gerente registrar as habilidades de cada pessoa que está em seu setor, contudo isso é um fato constante que faz com o cliente em cada etapa se atualize em relações as plataformas utilizadas, conhecimentos de programação adquiridos, experiência enquanto profissional entre outros fatos.

2.1.1 Perfil de Competência

A palavra competência vem do latim com a junção das palavras com e petere seu significado é esforço. De acordo com estudos realizados por Hamel & Prahalad (1995) a competência é o conjunto de habilidades que se entrelaçam e não são isoladas. Para se estabelecer uma competência são necessários 3 elementos importantes. O valor percebido pelo cliente, aos quais as empresas utiliza para prestar uma atuação de qualidade ao cliente, o segundo é a capacidade única e por último a capacidade de expansão que é estar aberto a novas tecnologias como elementos de diferenciação e até mesmo vantagem competitiva.

2.1.2. Competências individuais em Gerência de Projetos

Segundo o modelo *stakeholders* elaborado por Cleland (1998), distinguindo-os em primários e secundários são indivíduos que tem maior visibilidade quando se faz um projeto desses, podem ser, gerente de projetos, desenvolvedor e até mesmo o pessoal de suporte e manutenção. Os secundários apesar de não abordarem o problema proposto são: indivíduos do governo, terceirizações, etc. Levando em conta o modelo feito por Sthub et all (1994) e Rabechini Jr (2001) características em relação ao gerente e equipe são referentes a liderança, facilidade de negociação, capacidade técnica. Outros fatores também podem ser notados por exemplo. Segundo o PMBOK(2008) a comunicação eficaz significa que as informações são fornecidas no

formato correto, no tempo adequado e com o impacto necessário. Comunicação eficiente significa fornecer somente as informações quando necessárias. Caso a troca de informações não for válida e eficaz isso pode se tornar uma confusão de fluxo de informações o que ocasionaria problemas no desenvolvimento de software.

2.2 Raciocínio Baseado em Casos

Segundo Wangenheim (2003, p.2):

[...] o Raciocínio Baseado em Casos diferencia-se radicalmente de outras metodologias para o desenvolvimento de programas e sistemas da área de Inteligência Artificial (IA). Ao contrário de enfoques tradicionais para encontrar uma solução para um problema em IA, em que se descreve conhecimento genérico na forma de regras, quadros, roteiros etc., no Raciocínio Baseado em Casos é o conhecimento específico, na forma de exemplos concretos de casos, que se encontra no centro do processo de solução de um problema.[...]

O conceito RBC, compreende-se que para um domínio, os problemas presentes podem ser semelhantes com pequenas alterações e que outros casos poderiam ser resgatados para solucionar o problema atual, assim sucessivamente. Vale ressaltar a grande semelhança com situações do cotidiano, um dos mais comuns exemplos que pode-se averiguar é o diagnóstico de doenças. Uma pessoa quando vai ao médico apresenta sintomas semelhantes ao de pacientes anteriores, logo, um ou mais casos é resgatado da base de dados apresentando as possíveis soluções para o problema.

2.2.1 Ciclo do RBC

A escolha para a resolução de problemas RBC contém um ciclo de vida conforme é demonstrado na figura 2.



Figura 2 – Ciclo de Vida RBC Adaptado (Wangenheim, 2003)

Ao receber um novo caso o sistema realiza uma busca em sua base de dados quais os casos que se assemelham ao novo problema, essa busca é feita pelo cálculo de similaridade. Leva-se em conta no processo de recuperação as situações do problema em que se considera com maior importância (Wangenheim, 2003).

Recuperar

Neste processo é possível recuperar os casos mais similares acerca do problema proposto.

Reutilizar

A reutilização consiste na adaptação de casos anteriores para sanarem o problema do caso atual.

Verifica-se que depois do processo de recuperação nem sempre os casos recuperados satisfazem completamente o caso problema, sendo necessário passar pelo processo de adaptação para ser aplicado ao mesmo.

A estratégia utilizado para suprir esse problema muitas vezes não é pela adaptação mas da inflação da base de casos completamente colocando o maior número de casos possíveis na base de casos para seja qual for a necessidade do problema ser suprida pela quantidade de casos.

A adaptação tem uma importância fundamental nos sistemas RBC, pois de sua capacidade de adaptação para resolver novos problemas se torna um fator de validade na qualidade de casos recuperados.

Segundo Wangenheim (2003) há várias formas de adaptar um caso desta maneira, à várias questões a serem centradas nesse quesito como:

- Quais os aspectos de uma situação que devem ser adaptados.
- Quais serão os métodos utilizados para adaptar-se ao caso.
- Como verificar e confirmar esse processo de modificações no caso.

Algumas vezes as diferenças encontradas entre o caso problema e o caso base indica que há uma necessidade de adaptar a solução proposta com base no que foi determinado para ser modificado. Tende-se apurar outros requisitos que iram afetar essa modificação, sendo: efeitos colaterais, quantidade de modificação para satisfazer os requisitos de modificação.

Revisar

Compreende-se por revisão etapa em que a solução que foi reutilizada é revisada. A revisão abrange duas tarefas:

Avalia de modo rígido a solução gerada a partir do reuso. Se por consequência a considerar correta então a próxima e ultima etapa é a retenção, caso o contrário, adaptar a solução em conjunto com o usuário.

A primeira etapa da revisão consiste na detecção de falhas que passaram a ser fornecidas pelo sistema, a revisão aborda os casos que foram recuperados e verifica caso não satisfizeram a necessidade do caso problema os mesmo tem de serem explicados, sendo apresentado aos usuários como o sistema chegou a aquela solução, sendo por fim resolvida manualmente pelo engenheiro de conhecimento para que a mesma não volte a ocorrer.

Reter

Processo que são retidos os casos com suas soluções para casos futuros, tornando o sistema cada vez mais inteligente, quanto mais o sistema incorporar conhecimento, mais chances há de solucionar os problemas do novo caso.

Os sistemas de retenção de casos são 3 que serão explicados para se verificar qual o melhor abordagem na construção desta aplicação.

Sem retenção de casos:

Busca o desenvolvimento com protótipos para um conjunto de sintomas relevantes no domínio da aplicação, tendo em vista que o sistema não precisa incorporar novos casos.

Retenção de soluções de problemas:

Uma forma de aprendizado comum nos RBCs, aos quais passam por um processo de aprendizado do conhecimento com base no armazenamento de casos. Segundo (Hinrichs, 1992) ou (Hammond, 1986) “alguns sistemas retêm casos representando tentativas de solução de problemas que falharam, para permitir que se evite utilizar aquelas soluções no futuro.”

Por fim com a retenção de novos casos na base de casos o sistema adquire cada vez mais conhecimento, além do aprimoramento ocasionado por implantação de peso nas entidades com base nos resultados aplicados a ela, para quando necessitado, verificar um melhor tratamento do caso para a resolução daquele problema, quanto mais importante a característica no ambiente em que está sendo aplicada mais importância ela possui na hora de escolhas relevantes das características.

2.3 Recuperação de informação

Processo em que são recuperados os casos na base de casos. Segundo Wangenheim (2003) a recuperação se dá por meio de sub tarefas das quais são:

Assessoramento da situação: Etapa que se formula um conjunto de informações do caso problema.

Casamento: Processo que se realiza a identificação dos casos que se assemelham suficientemente ao caso problema.

Seleção: que escolhe os melhores casamentos com base nos conjuntos selecionados.

Entretanto nem sempre pode-se encontrar os casos exatamente semelhantes ao caso problema por isso utiliza-se meios para verificar a importância de fatores que

serão levados em conta no processo de busca. Os descritores de entrada são índices que auxiliam na busca por fatores relevantes como as características de cada membro.

Segundo Wangenheim (2003) os descritores podem ser representados de vários tipos como valores numéricos, simbólicos, listas ou strings, ou mesmo descrições textuais em linguagem natural.

2.4 Recuperação de casos

A recuperação de casos consiste na fase em que são realizadas as tarefas de similaridade para recuperar os casos úteis ao objetivo proposto (MENDES, 2009). Nesta etapa, são aplicados os algoritmos para a identificação da similaridade global, como o Nearest Neighbor, Naive Bayes, Árvore de decisão entre outros, cada um desses algoritmos tem uma melhor atuação dependendo do tipo de tarefa a ser realizada. A similaridade global é uma medida que determina a similaridade entre o caso de entrada e o caso da base, levando em consideração todos os índices (VON WANGENHEIM; VON WANGENHEIM, 2003).

Para deixar o cálculo da similaridade global mais sensível é necessário obter o valor da similaridade de todos os índices individualmente, definido como similaridade local. Sem o cálculo individual os valores da similaridade de um atributo só podem assumir 0 ou 1, isto é, 0 se forem diferentes e 1 se forem iguais (VON WANGENHEIM; VON WANGENHEIM, 2003).

Os métodos de análise são separados em diferentes grupos como: classificação, regressão, clustering e análise de associações.

A classificação busca características que sejam semelhantes com as de documentos que já foram classificados. A classificação na mineração de textos é uma tarefa que identifica os tópicos principais de um documento e sua associação baseia-se em um algoritmo pré-definido, construído a partir de um conjunto de treinamento definido por pessoas experientes no assunto (BARION; LAGO, 2008).

2.4.1 Distância Euclidiana

Segundo VICINI (2005) “Considerando o caso mais simples, no qual existem n indivíduos, cada um dos quais possuem valores para p variáveis, a distância euclidiana entre eles é obtida mediante o teorema de Pitágoras, para um espaço

multidimensional.” Pois bem, visto isso, se pode notar que a distância entre o caso os casos se da pelo espaço multidimensional, sendo quanto maior a distância entre eles maior é a diferença. Para Manly (1986), “a distância euclidiana, quando for estimada a partir das variáveis originais, apresenta a inconveniência de ser influenciada pela escala, de medida pelo número de variáveis e pela correlação existente entre as mesmas.”

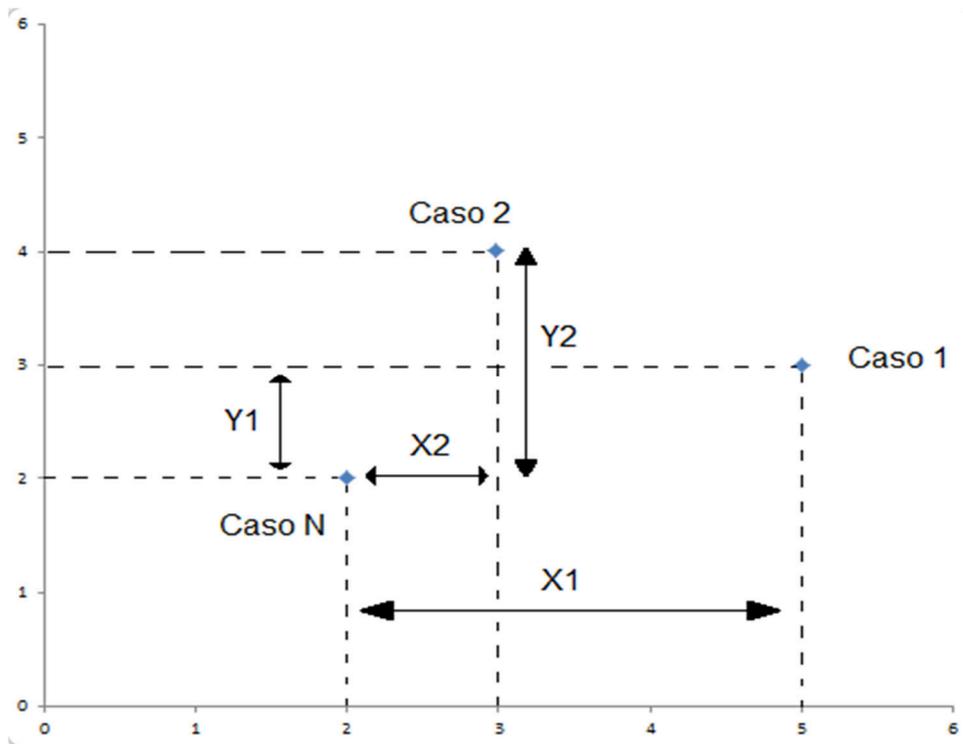


Figura 3 – Distância Nearest Neighbor (VON WANGENHEIM; VON WANGENHEIM, 2003)

Na figura apresentada mostra a distância entre os 3 casos chamados Caso N, Caso 1 e Caso 2. O Caso N possui em relação ao Caso 2 pontos de distância menor que em relação ao Caso 1 por isso o Caso 2 se torna mais próximo que o Caso 1 em relação ao Caso N.

Pode ser apresentado um exemplo também da seguinte forma: A distância euclidiana ente os indivíduos a e b é dada matricialmente por:

$$d_{ab} = [(X_a - X_b)' \cdot (X_a - X_b)]^{1/2}$$

$$X_a = [X_{a1} \ X_{a2} \ \dots \ X_{ap}]' = \text{vetor de características do indivíduo a};$$

$$X_b = [X_{b1} \ X_{b2} \ \dots \ X_{bp}]' = \text{vetor de características do indivíduo b}.$$

Quanto maior o número de características dentro do vetor for igual ou próxima maior é a similaridade entre X_a e X_b .

2.4.2 Distância Euclidiana Média

Conforme o crescimento do número de variáveis a distância aumenta, por isso quanto mais o número de comparações realizadas a chance de similaridade também vai diminuindo. Por isso a distância entre dois agrupamentos é obtida pela média das distâncias. (VICINI, 2005). A formula se dá da seguinte maneira:

$$d = \sum_{j=1}^p \left\{ \frac{(X_{ij} - X_{rj})^2}{X_{ij}} \right\}$$

2.4.3 Distância de Mahalanobis – D 2

Se trata de um conjunto de características correlacionadas considerando a distância entre dados amostrais feita pela verificação de dependência entre as variáveis. (VICINI, 2005). Conforme Cruz (1990),” a distância de Mahalanobis, considera a variabilidade de cada unidade amostral, sendo recomendada para dados provenientes de delineamento experimentais, e, principalmente, quando as variáveis são correlacionadas”.

3 Desenvolvimento

A etapa do desenvolvimento foi dividida em 4 etapas:

Explicação do desenvolvimento da ferramenta. O estudo realizado para verificar quais características são importantes. A simulação de uma base de informações que contenha implicitamente o conhecimento necessário na solução de problemas. E os testes a serem realizados para verificar resultados.

3.1 Etapas de Elaboração

Existem algumas etapas na elaboração de um RBC que são listadas por alguns autores (Wangenheim;Kolodner;Watson) sendo estas:

- Realizar simulação de uma base de dados.

A base foi simulada com a inserção de membros e de listagem dos mesmos em características para a realização do cálculo de similaridade.

- Definição de quais características da informação serão relevantes e podem ser utilizados para a solução do problema.

Foi utilizada a extração das características com base em pesquisa para listar quais são validas para o domínio proposto, com base nisso se pode questionar quais as características importantes na escolha de determinado membros para a alocação no projeto tendo em vista o problema proposto pelo usuário.

- Definição de índices para os casos, para que seja possível a sua recuperação quando necessário.

Segundo (Wangenheim, 2014) “os índices são aquelas entidades de informação relevantes para a determinação da similaridade entre um caso e a pergunta ou o problema”. Levanto em conta as questões propostas, isto é, a alocação de pessoas no projeto pode se perguntar o quanto é importante uma linguagem de programação para determinado projeto.

- Definição dos métodos de recuperação de casos para verificação da similaridade entre os casos contidos na base e os novos problemas.

Os métodos que serão utilizados são os de similaridade local e global que serão explicados mais detalhadamente no processo de desenvolvimento em que serão utilizados.

- Realizar experimentos

O processo de etapa de realização de experimentos foi feita depois que o processo de desenvolvimento estiver concluído, para validar os resultados obtidos.

3.2 Características

Atributos ou chamado também de característica de uma situação-problema é uma informação que auxiliou na busca pela melhor resolução possível do problema essa no que lhe diz respeito pode se tratar de sexo, idade, sintomas de pacientes etc. (Wangenheim, 2014).

Levando em conta o domínio proposto é preciso verificar quais as características relevantes que fez do sistema ser realmente eficiente.

No caso do domínio do projeto serão as funcionalidades e particularidades de cada membro da equipe. Esse tópico tem por objetivo demonstrar de forma simples quais são esses atributos e quais os motivos e significados que fazem eles se tornarem úteis ao domínio da aplicação.

3.2.1 Linguagem de Programação

É necessário saber inicialmente quais as áreas de atuação dos membros.

Linguagem de programação	Média salarial das vagas	Número de oportunidades	% do total de oportunidades
SQL	R\$ 3.458	4.874	33,54%
Javascript	R\$ 3.252	2.123	14,61%
PHP	R\$ 2.387	1.986	13,67%
CSS	R\$ 2.753	1.851	12,74%
C#	R\$ 3.609	1.269	8,73%
HTML5	R\$ 3.052	584	4,02%
C	R\$ 3.951	535	3,68%
Java	R\$ 4.211	416	2,86%
C++	R\$ 4.337	321	2,21%
Ruby	R\$ 3.667	251	1,73%
Python	R\$ 4.420	233	1,60%
Objective-C	R\$ 3.903	87	0,60%

Figura 4 – Ranking de Busca Por Linguagem (Exame.com 2016)

Segundo a pesquisa, o site demonstra quais as linguagens estão sendo mais requisitadas no mercado de trabalho e com isso tende-se uma base de quais são as importantes linguagens que serão necessárias para o projeto. De acordo com o que demonstra a figura, observa-se que o SQL como linguagem “padrão” para as empresas, que necessariamente precisam de analistas capazes de gerenciarem suas informações está na primeira colocação, além dessas as já populares linguagens como Java, C e PHP também aparecem entre as mais usadas, nota-se também o crescimento do uso de Java Script, contudo por mais que as mesmas possuam maior número de oportunidades nos mercados, essa habilitação não está entre as mais bem remuneradas do mercado, visto que o Java e o C possuem melhores salários.

Com o desenvolvimento das tecnologias e linguagens de programação, atualmente pode não ser suficiente estipular somente a linguagem de forma genérica como critério para seleção de membros da equipe, pode-se pedir um maior número de conhecimentos específicos usados dentro das próprias linguagens.

Os ambientes de programação também são importantes já que são utilizados para comportar as IDEs que estão presentes no desenvolvimento, os sistemas operacionais mais comuns que suportam esse tipo de ferramenta são o Windows e Linux. Segundo (GOMES; QUESADO, 2003) as vantagens aplicadas ao Linux em relação ao Windows é a vantagem do software livre e amenização de custos iniciais em determinada empresa, por outro lado o software proprietário pode-se encontrar diversos recursos disponibilizados com interface mais amigável o que gera a aceitação mais fácil da comunidade de desenvolvimento. Outro fator que aborda essa questão de preparação para o desenvolvimento é a chamadas classes utilitárias que são classes que contém métodos pré-definidos que além de adiantar também já facilita alguns aspectos dentro da programação, segundo (MEDEIROS, 2013)

Outro aspecto que deve ser levado em conta nas aplicações WEB ou Desktop são os fatores de interface gráfica que são atualmente importante, afetando tanto na performance quanto na satisfação do usuário (Cliente) que por sua vez exigem em grande parte uma interface descomplicada e visualmente atraente.

Dentro da linguagem Java por exemplo existem algumas especificações que auxiliam nessa etapa, e atualmente é exigido dos membros da equipe por grande parte das empresas que buscam empregados. O Java Server Faces é uma padrão na construção de interface que utiliza dos componentes visuais oferecidos por ele mesmo para facilitar e realizar tarefas que demandariam mais tempo e custo no decorrer do desenvolvimento.

Segundo Bueno (2014) o JSF nada mais é que um framework em que é efetuada a elaboração de interface de usuários para sistemas web, realizando a conexão de componentes e objetos Java pelos formulários, sendo assim ele faz a separação entre a lógica e a regra de negócio a navegação e conexões com serviços externos seguindo o modelo MVC.

Outras aplicações englobadas atualmente junto as linguagens de programação como PHP e JAVA são já conhecidas HTML, CSS e Ajax além do novo, esperançoso e precursor do Java Swing o JavaFX.

“HTML5 é certamente uma boa escolha em muitos casos, mas há e continuará a haver circunstâncias em aplicativos de desktop clássico fazem mais sentido. E para isso que o JavaFX é seguramente uma boa opção.”(MAUKY, 2016)

3.2.2 Java FX

É uma plataforma de software desenvolvida pela Oracle que auxilia na criação de ricas interfaces gráficas dentro da linguagem Java. Se destaca por apresentar total distinção entre código-fonte e a própria interface, entretanto a plataforma Java FX por mais que parece inicialmente interessante ainda passa por aceitação da comunidade, ao contrário do JSF que foi muito bem recebido, contudo a Oracle (2012) aposta de forma convincente na plataforma afirmando ser a evolução do Java por poder ser utilizada tanto em aplicações Web como Desktop.

A separação de código e interface é feita por uma linguagem XML chamada FXML que contém as informações referentes a determinados componentes (Métodos, Ids, Textos, Layout, Tamanho, entre outros). Outro fato favorável é que o JavaFX possui uma ferramenta que auxilia e facilita e muito a criação dessas interfaces chamado o Java Scene Builder.

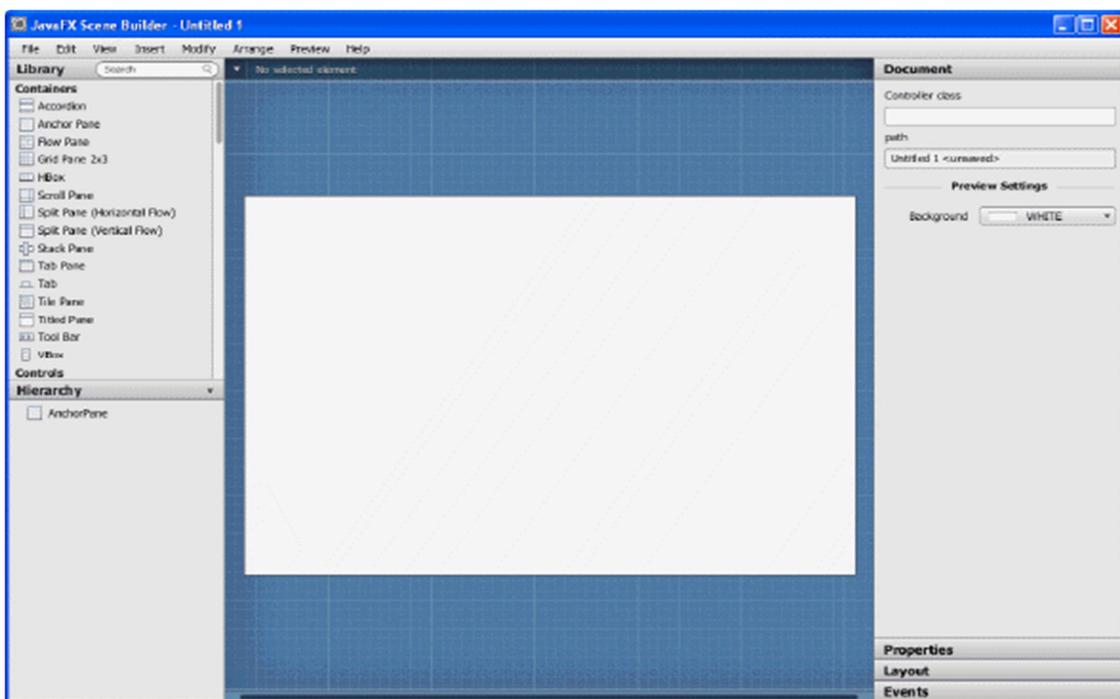


Figura 5 – Java Scene Builder

Como se pode observar, se trata de uma ferramenta que pode ser utilizadas em softwares que requerem uma alta performance gráfica exigida pelo cliente e o JavaFX pode vir a se tornar requisito para as empresas atualmente.

3.2.3 O Padrão MVC

O padrão Model-View-Controller ou somente MVC segundo Higor Medeiros (2015) é uma forma de quebrar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador.

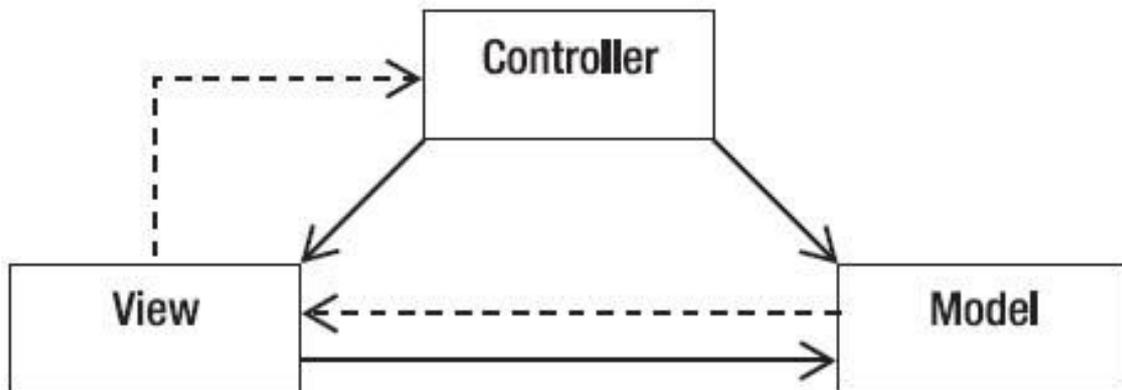


Figura 6 – O modelo MVC

O **Controller** realiza a ligação entre o modelo e o view, o controller é responsável pelas teclas de entrada dos usuários e que faz a ligação entre view e modelo.

O **Model** é o gerenciamento dos dados que são realizados se pode definir também o local ficam armazenados os atributos e informações referentes ao sistema.

A **View** é a interface do sistema em que o usuário interage para requerer ao controlador realizar as alterações requisitadas por ele.

O **padrão MVC** traz vários benefícios as empresas que o utilizam como redução e padronização do código fonte, ocasionando facilidade futura na manutenção e reorganização dos dados, além de diminuição de custo e tempo. Tendo em vista tamanha importância é requisitado dos usuários conhecimentos muita das vezes prévio para se trabalhar com tal ferramenta.

3.2.4 Banco de Dados

Fator importante também para as empresas como já demonstrada na figura 4 é o conhecimento hoje de banco de dados. Qualquer ambiente de desenvolvimento se requer uma base de dados para organizar e gerenciar dados das organizações, por isso as empresas solicitam também o conhecimento de linguagens comuns nos dias atuais como o SQL.

Observa-se a popularidade dos SGBDs que estão no mercado atualmente:

Rank			DBMS	Database Model	Score		
Oct 2016	Sep 2016	Oct 2015			Oct 2016	Sep 2016	Oct 2015
1.	1.	1.	Oracle +	Relational DBMS	1417.10	-8.46	-49.85
2.	2.	2.	MySQL +	Relational DBMS	1362.65	+8.62	+83.69
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1214.18	+2.62	+90.95
4.	↑5.	4.	MongoDB +	Document store	318.80	+2.81	+25.54
5.	↓4.	5.	PostgreSQL	Relational DBMS	318.69	+2.34	+36.56
6.	6.	6.	DB2	Relational DBMS	180.56	-0.62	-26.25
7.	7.	↑8.	Cassandra +	Wide column store	135.06	+4.57	+6.05
8.	8.	↓7.	Microsoft Access	Relational DBMS	124.68	+1.36	-17.16
9.	↑10.	↑10.	Redis	Key-value store	109.54	+1.75	+10.75
10.	↓9.	↓9.	SQLite	Relational DBMS	108.57	-0.05	+5.90

Figura 7 – Ranking de Popularidade SGBDs (DB-Enginner, 2016)

Conforme pode ser visualizado o ranking aponta o SGBD Oracle estando em primeiro lugar, visto ser muito usado atualmente nas empresas, seguido do MySQL que desde a um bom tempo vem sendo um dos mais utilizados, contudo percebe-se também um crescimento nos SGBDs não relacionais, tendo em vista que as grandes empresas têm de possuir uma alta escala de armazenamento e acesso rápido a informação. Por isso o conhecimento desses SGBDs vem sendo observado pelas empresas que realizarão o contrato de pessoas na área e desenvolvimento e administração de SGBDs.

Segundo o Simpósio Brasileiro de Banco de Dados(2016) a várias áreas de estudo em que os SGBDs vem se desenvolvendo para assim realizar uma melhora nos fatores dos mesmo, algumas áreas são: Data WhareHousing and OLAP, NoSQL DataBase, Peer-To-Peer Paralelo e DataBase Distribuído). Esses são algumas áreas de atuação em pesquisa mas que já vale ser observado pelas empresas.

“A criação de Sistemas de Gerenciamento de Banco de Dados Distribuído contribui de forma significativa para o aumento da produtividade em desenvolvimento de aplicações, um fator importante desde longa data” (Moura; Casa Nova, 1999).

Como se pode perceber empresas que geram conteúdo já transgrediram para esse novo conceito de armazenamento e gerência. As empresas responsáveis por esses sistemas já não podem depender de um conhecimento que ainda por mais que seja utilizado por sistema de médio e pequeno porte não pode ser mais proficiente quando se trata de grandes quantidade de dados.

Empresas já sofrem para administrar grandes quantidade de dados usando um SGBD tradicional, por esses fatores as empresas de desenvolvimento buscam pessoas cada vez mais especializadas em novos conceitos de armazenamento e gerenciamento dos dados para sistemas de grande e até médio porte.

3.2.5 Hibernate

Com o Java estando entre as linguagens de programação mais requeridas e que a pouco tempo era o grande destaque do mercado segundo uma pesquisa realizada a anos atrás diz que o Java pode não ser a linguagem mais atraente ou mesmo a mais popular, mas é de longe a mais rentável de se envolver por ser uma habilidade altamente desejada, além de ser a mais bem paga do mercado (JAX ENTER, 2013)

Se levarmos em conta a importância e a constância do Java no mercado se torna visivelmente considerável também o uso de uma ferramenta de auxílio para aperfeiçoar a prática e gerenciar o controle dos dados dentro dessa aplicação.

“O Hibernate é um framework para mapeamento objeto-relacional para linguagem Java.” Na prática, ele é um conjunto de classes, interfaces e arquivos de configuração pré-acabados que permitem a criação de uma camada de serviço capaz de abstrair a existência do banco de dados para sistemas java” (MRACK, 2016).

As vantagens na utilização deste framework pelas empresas é que o mesmo traz veracidade, praticidade, e facilidade de manutenção no decorrer do projeto.

Grandes empresas atualmente utilizam o hibernate relatando grande melhora no desenvolvimento e tempo de implementação assim como a diminuição de código-fonte por não precisar utilizar simplesmente a linguagem SQL.

Segue abaixo a lista das empresas de grandes empresas que utilizam o hibernate em seus projetos.

Ubik-Ingénierie, ubik-ingenierie.com, Roubaix, França
 Fedelta POS, fedeltapos.com, Brisbane, Austrália
 Skillserv, skillserv.com, São Francisco, Califórnia, EUA
 Nome da empresa, Local: SoftSlate Commerce, NY, EUA
 Projeto Open Source: Wilos - <http://www.wilos-project.org/>
 Nome da empresa, Localidade: GPI Argentina, La PLata, Buenos Aires, Argentina
 Projeto Open Source: itracker
 Nome da empresa, Local: TerraContact Inc., Montreal, Canadá
 Nome da empresa, Local: LF Inc., Tampa, FL
 Nome da Empresa, Local: Elastic Path Software, Vancouver, BC, Canadá
 Nome da empresa, Localidade: argus Barcelona, Europa
 Nome da empresa, Local: AT & T Labs, Tampa, Flórida
 Nome da Empresa, Local: JTeam, Amsterdam, Holanda
 Nome da empresa, Local: 1Genia, Paris, França
 Nome da empresa, Localidade: TDC Internet, Warsaw, Poland
 Nome da Empresa, Local: PriceWaterhouseCoopers, Tampa, Flórida
 Nome da empresa, Local: 2Fi Business Solutions Ltd., Hong Kong
 Nome da empresa, Local: Intrasoft International, Bélgica, Bruxelas
 Nome da Empresa, Local: Burgerweeshuis, Holanda, Deventer
 Nome da empresa, Local: Cisco Learning Institute, Phoenix, AZ, EUA
 Nome da empresa, Local: Open Lab Srl, Florence I
 Nome da empresa, Localidade: DriveNow, Austrália
 Sony Computer Entertainment Europa, SCEE, Studio Liverpool, Liverpool, Reino Unido
 AonCHOR, <http://www.aonchor.aon.com>, Serviço de Risco Aon, EUA
 Nome da empresa, Localidade: Church and People, Nova York
 Crank Roupa, t-shirts e fato

Figura 8 – Utilização do Hibernate por Grandes Empresas (PATRICIO, 2013)

“Quando se mapeia os dados do modelo relacional cria-se o efeito de que o banco de dados seja orientado a objetos, de modo que o programador não precisa se preocupar com a disposição dos dados nas tabelas e se focar na manipulação de objetos e nos problemas de negócio” (BAUER, KING 2007).

Essas são algumas das condições que faz do Hibernate estar presente como um fator importante no desenvolvimento de software das empresas, e isso deve ser levado em conta na seleção de indivíduo a determinado tipos de projeto, o que faz ele estar entre as características do projeto.

3.2.6 Code Igniter

Se tratando da linguagem para criação de sites, no PHP também observa-se frameworks das quais as empresas utilizam no ambiente de programação para uma maior performance no sentido de codificação e organização, o Code Igniter oferece sua atividade sobre o MVC que torna simples implementação de algumas classes por justamente trazer vários tipos de tarefas comuns em qualquer sistema que esteja sendo desenvolvido como: conexões com o banco de dados, formulários prontos em

HTML entre outras ferramentas prontas. Por isso o Code Igniter pode ser encontrado entre os frameworks mais utilizados em 2015. (Avoyan, 2015)

Dessa forma conclui-se que das empresas de desenvolvimento Web o crescimento dos frameworks deve ser visto pelo membro da equipe como uma forma de adquirir conhecimento e melhorar o bom andamento tanto individual quanto da equipe de desenvolvimento, principalmente os mais usados como o já referenciado Code Igniter deve ser de alta escalabilidade entre os empregados para assim diminuir tanto o custo como tempo e também a equipe poder fazer com que a equipe possa compartilhar uma mesma ideia de desenvolvimento.

3.2.7 Angular CLI

Segundo (Rodrigo Waltenberg, 2016) “A premissa do Angular é ser um Framework MV* (Model – View – Qualquer Coisa) para desenvolvimento do front-end de aplicações web as quais rodam dentro do navegador do cliente.”

De acordo com os dados no repositório do GitHub o Angular possui 49 mil estrelas e mais de mil contribuintes além de mais de 150 mil repositórios que utilizam essa tecnologia. (Rodrigo Waltenberg, 2016). Devido a essa alta comunidade o framework lista-se sendo uma das características dos desenvolvedores.

3.2.8 Servidores de Aplicação e Web Container

Segundo um artigo publicado pela (IWEB, 2003)

“Os servidores de aplicação são executados em servidores e são acessados pelos clientes através de uma conexão de rede. As vantagens dos servidores de aplicação em relação ao modelo cliente/servidor residem nos serviços implementados por eles e disponíveis aos desenvolvedores fazendo com o que eles possam concentrar a maior parte do tempo no desenvolvimento da lógica de negócio. Em geral estes serviços diminuem a complexidade do desenvolvimento, controlam o fluxo de dados, incrementam a performance, gerenciam a segurança.”

Os servidores de aplicação e Web Container estão presente nos desenvolvimentos web entre os quais se destacam os que serão listados como um fator importante de conhecimento do membros da equipe sendo eles: GlassFish,

Apache/Tomcat e JBoss por serem os mais utilizados nas empresas e que possuam maior credibilidade, no caso do Apache/Tomcat o mesmo é utilizado em cerca de 60% dos servidores web do mundo (IWEB, 2003).

Veja abaixo o que são e porque os servidores de aplicação e web container são importantes:

Tomcat: Foi desenvolvido pela Apache e permite a execução de aplicações para a Web seu foco é nas tecnologias JSP e Servlets entretanto não incorpora a tecnologia EJB ao contrário do GlassFish por isso o Tomcat é recomendado para projetos mais simples que não requerem maior performance, entretanto por englobar maior número de funcionalidades em relação ao Tomcat o GlassFish se torna mais pesado.

ClassFish: Utilizados na criação de aplicativos JavaEE que diferente do tomcat inclui diversas tecnologias como JSF, EJB, JPA.

JBoss: Consiste em um servidor de aplicação J2EE além de ser gratuito e de código aberto é considerado um dos mais usados entre as empresas.

Dadas as diferenças esses servidores também possuem suas semelhanças como possuem código-fonte aberto e são livres para usar, para as empresas, esses, são considerados fatores de grande benefícios por auxiliarem de forma impactante no desenvolvimento sem nenhum custo por parte da mesma.

3.2.9 Relatórios

Outrora os relatórios eram criados a partir de linguagem XML ou até mesmo em programas de estatísticas, atualmente existem softwares comerciais que junto deles estão incorporados relatórios criados a partir de software especializados, esses softwares são muitas das vezes requisitos para as empresas diante de seus empregados.

As empresas quando procuram um software principalmente no sentido comercial, convém de obterem relatórios por esses softwares de maneira mais simples e eficaz, as empresas de desenvolvimento analisando os fatos, hoje recruta membross com conhecimentos para poderem criar esses relatórios nos softwares

dos clientes. Há vários softwares que auxiliam na criação de relatórios e são bastante utilizados para as empresas que contam como requisito tanto para a seleção de pessoas nos projetos como também a própria contratação de membros.

Utilizados tanto em aplicações Web como Desktop o iReports criado pela JasperReport é um dos softwares de criação de relatórios que necessitam do conhecimento do desenvolvedor para que os clientes obtenham informações de custos, contas, pagamentos e o que mais desejarem, para que auxiliam na tomada de decisão dessas empresas.

É possível para esse software gerar, imprimir e exibir relatórios em formatos XML, HTML, PDF.

3.2.10 Níveis de Programador

Dentro dessas formações se tem algumas hierarquias que foram levadas em conta no processo de escolha de determinado membro para a equipe, pois por mais que determinado membro possua conhecimento de determinada linguagem não quer dizer que necessariamente está apto a ser responsabilizado com um cargo no projeto. Por isso é necessário saber ainda qual é o nível de profundidade de determinado candidato a aquela vaga no projeto, com base nisso tende-se no nível grande importância na hora da escolha, podendo ser **Trainee, Junior, Pleno, Sênior, Master**.

Para uma melhor compreensão visual e simplificada demonstra-se abaixo:

Classificação “padrão”

Nível	Tempo de experiência	Formação	Responsabilidades
Trainee	2 a 2 anos e meio	Recém-graduado	Tarefas de pequena ou média complexidade em área(s) específica(s). Elabora projetos (sob supervisão)
Junior (JR)	até 5 anos	Recém-graduado	Funções de procedimentos simples ou que não exigem profundo conhecimento em um ramo de atuação
Pleno (PL)	6 a 9 anos	Pós-graduado	Atividades específicas, que exigem profundo conhecimento. Toma decisões endossadas por um superior.
Sênior (SR)	a partir de 10 anos	Pós-graduado + Gestor	Toma decisões. Age de forma autônoma, com base no conhecimento e experiências adquiridos ao longo da carreira. Gere pessoas e projetos.
Master	15 anos ou mais	Pós-graduado + Gestor + Certificações	Atua fora do processo de supervisão ou por demandas. Gere projetos / negócios. Possui autonomia plena.

Figura 9 – Níveis de Programador (Oficina Da Net, 2011)

Enquanto os profissionais Trainee e Junior ficam responsáveis por tarefas relativamente simples, os programadores Pleno e Sênior assumem complexidades maiores muitas das vezes de alto risco e com pouco prazo de desenvolvimento, por fim tem-se também o nível máster que é responsável por orientar, gerir processos e negócios do sistema.

Outros fatores relevantes que devem ser citados é conforme o cargo ao qual o membro está, a faixa salarial do mesmo vem a se tornar mais alta que a anterior, isso se deve pela responsabilidade atribuída a determinada pessoa como mostra a figura 11.

Essas faixas podem se dar por período de trabalho, experiência na área, grau de formação ou nível de bilinguismo.

3.2.11 Nível de bilinguismo, trilinguismo ou poliglota.

A língua inglesa é considerada uma grande aliada quando o assunto é buscar uma vaga de emprego ou estar competente para assumir um projeto, muitas

das vezes um projeto de software não é feito somente dentro da empresa mas necessita de pessoas do exterior para a realização de tal, isso vai desde o próprio cliente ou até mesmo uma comunicação entre empresas de outras nacionalidades e linguagens de programação.

“E muitas das palavras e dos termos que são utilizados na língua inglesa para a área de TI são termos que estão interligados uns com os outros, e por isso, não basta traduzir essa palavra e tentar entender o que isso significa, mas sim, entender o conjunto como um todo” (MIRANDA, 2015).

Vale se notar também que no ambiente de desenvolvimento de software o inglês é um dos requisitos mais importantes por estar presente constantemente devido aos fatores já citados, pois desde o início da caminhada na área de TI pessoas se deparam com tais termos.

Outras línguas também são úteis quando se trabalham para outros países na Europa como por exemplo o espanhol, francês e alemão também podem ser utilizados.

3.2.12 Certificações

Com o crescimento das tecnologias as empresas estão sempre mudando e requerendo maior experiência de habilidades específicas.

As certificações podem ser um fator importante na hora da escolha de determinado membro da equipe, pois estabelece requisitos específicos de qualidade de determinado indivíduo além de ser também um fator comprovador de determinado conhecimento na contratação do membro, pois a empresa desse modo tem plena certeza que a habilidade apresentada é realmente dominada pelo entrevistado. As certificações também podem ser em relação a formação de determinado membro, por exemplo, o membro da equipe ao qual está sendo selecionado tendo em vista sua formação (Sistemas de Informação, Análise e Desenvolvimento de Sistemas, Ciência da Computação etc.) tem o conhecimento que necessita na função em que está buscando? De acordo com Mckillip e Cox (2017), as certificações profissionais estão crescendo em importância no mercado de trabalho, sendo usadas para identificar e fortalecer a competência do profissional, cortar custos e atrair clientes, atestando para um nível mais avançado de desempenho. Outros exemplos que podem ser citados é qual tipo de conhecimento em outras áreas esse candidato possui? Segundo Almeida (2014) Como

profissionais de TI temos a liberdade de relacionar áreas de interesse aparentemente distintas. Podemos, por exemplo, gostar de TI e meio ambiente, por isso é possível atuar na área de **TI Verde** alinhando necessidades das empresas com o mínimo impacto no meio ambiente. Contudo no que se refere a certificações abaixo são listadas as certificações relacionadas a conhecimentos específicos dos membros da equipe.

Oracle Certified Associate, Java SE 7 Programmer:

Segundo Kengi (2015) é a certificação básica que resume os conceitos mais simples da programação java a definição de variáveis estruturas de decisão, classes, métodos etc.

Oracle Certified Professional, Java SE 7 Programmer

Considerada por muitos o nível médio esta já aborda conceitos mais aprofundados da programação java como hash, toString, Threads entre outros temas.

Oracle Certified Expert, Java EE 6 Web Component Developer

Como o nome já apresenta a certificação voltada para Web no Java verifica os conceitos de JSP e Servlets, podendo encontrar MVC(Model-View-Controller), que também consta na verificação de escolha de candidatos.

Oracle Certified Expert, Java EE 6 Enterprise JavaBeans Developer

De acordo com Kengi (2015) essa certificação é voltada para os componentes de negócio podendo apresentar questões de Java avançado como: arquitetura JEE, JEE containers, Componentes EJB, Session Beans.

Oracle Certified Expert, Java EE 6 Java Persistence API Developer

Voltada mais a integração com Banco de Dados a certificação Persistence API engloba conceitos como: EntityManager, Persistence, Entity.

Zend PHP 5 (Zend PHP 5.3 Certification)

Necessita conhecer de forma bem clara a linguagem além de recursos externos como Banco de dados e verificações de código para a resolução dos exercícios propostos.

Oracle Certified Associate, MySQL 5

Aborda Conceitos de SQL básico recomendado fortemente direcionada para os iniciantes nos conceitos MySQL. Para realizar essa avaliação não é necessário nenhum tipo de pré-requisito

Oracle Certified Professional, MySQL5 Database Administrator

Certificação de nível intermediária nos conceitos de MySQL nesse nível a pessoa é capaz de criar e gerenciar um banco de dados a partir do seu ponto inicial, buscar e realizar aprimoramentos que iram render uma melhora na performance do banco, analise de consultas etc.

MCSA: SQL Server

Quando se tem essa certificação a pessoa estará qualificada a desenvolver ou gerenciar um banco de dados.

3.2.13 Cloud Computing

Cloud Computing é um conjunto de tecnologias de computação que estão moldando um novo ambiente de oferecimento de serviço e trabalho a partir dos anos 2000. (Joyanes, 2009). Pode-se verificar que de acordo com esse novo sistema de atuação que as empresas de TI tendem a trabalhar não somente por softwares e hardwares que demandam maior tempo e espaço nas empresas contratantes pois o software precisava de assistência e manutenção nos ambientes empresarias, a Cloud vem com a proposta de isentar as empresas de desenvolvimento de diversos fatores externos, como os exemplos aqui citados.

Por fim, apresentada as características foi realizado a construção de uma tabela a qual foi demonstrada e explicada de uma forma mais detalhada na etapa de

desenvolvimento. Esta tabela contém as características e graus de similaridade de forma a ser mais fácil de ser demonstrada.

3.3 Construção da base de casos

Casos consistem na utilização de soluções anteriores para um novo problema apresentado.

Nesse projeto foi realizado com base em um domínio estabelecido, a ferramenta de seleção de pessoas em projetos, buscando sempre montar uma equipe da melhor forma possível.

3.3.1 Construção de Tabelas

Inicialmente a busca por uma construção de uma base bem elaborada junto de um especialista é essencial para a construção de uma elaboração do mesmo, tendo em vista que quanto maior as características nesse caso, maior a satisfação final do resultado com base no problema proposto. Levando em conta o que foi pesquisado, a tabela a seguir apresenta os fatores que estarão presentes na base de dados:

Características Técnicas

Característica	Descrição	Referência Bibliográfica
Linguagem de Programação	Linguagem de Programação principal de um projeto a ser desenvolvido.	CAMILA, Pati; 12 linguagens de programação muito requisitadas no mercado: Disponível em < http://exame.abril.com.br/carreira/12-linguagens-de-programacao-muito-requisitadas-no-mercado/ >
Linguagem Complementares	Linguagens Complementares a serem escolhidas com base nas linguagens principais. Por isso seu peso se torna um pouco menor	CAMILA, Pati; 12 linguagens de programação muito requisitadas no mercado: Disponível em < http://exame.abril.com.br/carreira/12-linguagens-de-programacao-muito-requisitadas-no-mercado/ >
Banco de Dados	Um banco de dados é uma coleção de dados relacionados, possuindo as seguintes propriedades implícitas: 1. Representa algum aspecto do mundo real (minimundo ou universo de discurso); 2. É uma coleção de dados logicamente coerentes e que possuem algum significado inerente; 3. É projetado, construído e instanciado ("povoado") para uma aplicação específica.	Elmasri, R. & Navathe, S.B. Sistemas de Banco de Dados, 4ª Ed., Pearson/Addison Wesley, 2005.
FrameWorks	Wolfgang define frameworks como "arquiteturas semifinadas reutilizáveis para vários domínios de aplicação" e arquiteturas para serem "vários componentes individuais e a cola apropriada entre eles	Design Patterns for OO Software Development, Wolfgang Pree, Addison Wesley.
Servidores de Aplicação e Web Container	Os servidores de aplicação são executados em servidores e são acessados pelos clientes através de uma conexão de rede.	Site: IWEB; Servidores de Aplicação.
Relatórios	Criador e Gerador de relatórios para os clientes.	MRACK, M. (2006) "Hibernate, Uma visão geral sobre o framework padrão de fato para mapeamento objeto-relacional" Disponível em: < http://www.slideshare.net/mrack/workshop-hibernate-com-comentarios >
Níveis de Programador	Grande parte das empresas classificam a carreira de um profissional . Geralmente essa classificação ocorre pelo tempo de experiência dentro de uma empresa.	FERNANDO, Bagno; Níveis profissionais Júnior, Pleno e Sênior na Carreira de Programador: Disponível em < https://www.profissionaisti.com.br/2015/06/niveis-profissionais-junior-pleno-e-senior-na-carreira-de-programador/ >
Certificações	De acordo com Mckillip e Cox [14], as certificações profissionais estão crescendo em importância no mercado de trabalho, sendo usadas para identificar e fortalecer a competência do profissional, cortar custos e atrair clientes, atestando para um nível mais avançado de desempenho.	MCKILLIP, Jack, COX, Caryl. Strengthening the criterion-related validity of professional certifications. Evaluation and Programming Planning,
Sistemas Operacionais	Ambiente computacional que são desenvolvidos os sistemas.	GOMES, Fabio Naveca; QUESADO, Eduardo Filgueiras. Estudo Comparativo Entre Ferramentas Gratuitas Versus Proprietárias em um Processo de Desenvolvimento de Software Orientado a Objetos - Centro Universitário Luterano de Manaus, 2003.
Utilitários	Padrões e Classes que compõem um ambiente de programação.	As classes utilitárias são bastante utilizadas em diversos frameworks e diversas aplicações corporativas.
Cloud Computing	Cloud Computing é um conjunto de tecnologias de computação que estão moldando um novo ambiente de oferecimento de serviço e trabalho a partir dos anos 2000.	JOYANES, AGUILAR Luiz La Cloud Computing (Computação em Nuvem): O novo paradigma tecnológico para as empresas e organizações na Sociedade do Conhecimento
Conhecimento de Cloud	Níveis de Conhecimento: Baixo, Intermediário, Avançado.	JOYANES, AGUILAR Luiz La Cloud Computing (Computação em Nuvem): O novo paradigma tecnológico para as empresas e organizações na Sociedade do Conhecimento
Conhecimento de Redes	Níveis de Conhecimento: Baixo, Intermediário, Avançado.	JOYANES, AGUILAR Luiz La Cloud Computing (Computação em Nuvem): O novo paradigma tecnológico para as empresas e organizações na Sociedade do Conhecimento

Características Humanas

Característica	Descrição	Referência Bibliográfica
Facilidade de Negociação	Capacidade de negociar durante o projeto.	PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000
Áreas de Conhecimento	Área em que o indivíduo possui conhecimento fora do ambiente computacional	BRUNO, Almeida;Carreira de TI em outras áreas do conhecimento: Disponível em< https://www.professionaisti.com.br/2014/02/carreira-de-ti-em-outras-areas-do-conhecimento/ > PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.
Capacidade de Lidar com Mudanças	Níveis de Conhecimento: Baixo, Intermediário, Avançado.	PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.
Facilidade de Trabalhar em Equipe	Níveis de Conhecimento: Baixo, Intermediário, Avançado.	PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.
Área Preferencial de Atuação	Área em que o indivíduo sonha alcançar dentro do ambiente de trabalho.	PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.
Facilidade De Comunicação	Troca de Informações entre a equipe durante o desenvolvimento.	PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.
Formação	Podendo estar junto a certificações a busca por aprimoramento tende a ser constante para realizar os preenchimentos de vagas no mercado de trabalho.	ABES; Mercado Brasileiro de Software. Disponível em < http://www.abessoftware.com.br/dados-do-setor/dados-2014 >
Capacidade de Liderança	Caso o funcionário pode tomar certas decisões durante o desenvolvimento em relação a equipe ou a outro funcionário.	PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.
Línguas	Caso o software a ser desenvolvido tenha ligações com empresas do exterior.	MIRANDA, Willian; A importância do inglês para um profissional de Ti Disponível em< http://aprendaplsql.com/2015/02/importancia-ingles-para-um-profissional-de-ti/ > Acesso em 01 out. 2016

Quadro 2 – Listagem das Características Humanas

As figuras demonstram os tipos de características que serão listadas no projeto. Na coluna de Descrição é verificado em alguns casos o nível de qualidade do conhecimento do membro naquela área sendo quanto maior o nível maior é o peso

em que ele se encontra caso for um nível mais baixo menor é seu peso em relação ao nível alto.

Linguagem de Programação	SIM	Linguagem Programação Secundária	SIM	Banco de Dados	SIM	Frameworks	SIM	Servidores de Aplicação e Web Container	SIM
Java	1	JavaScript	1	Oracle	1	Hibernate	1	GlassFish	1
PHP	2	CSS	3	MySQL	1,5	CodeIgniter	5	Tomcat	2
.dotNet	3	HTML 5	2	PostgreSQL	2	AngularCLI	6	Jboss	3
Python	3,5			MongoDB	3				
C#	4			Microsoft SQL Server	1,5				
C++	4,5			DB2	4				
C	5			Cassandra	3,2				
Ruby	8			Microsoft Access	4				
Objective-C	9			Redis	5				
				SQLite	1,7				

Relatórios	SIM	Níveis de Programador	SIM	Linguas	Nível da Linguagem	SIM	Certificações	SIM	Sistemas Operacionais	SIM
iReports	1	Trainee	0,5	Inglês	Ótimo	1	Oracle Certified Associate, Java SE 7 Programmer	1	Windows	1
		Junior	1	Espanhol	Regular	2	Oracle Certified Professional, Java SE 7 Programmer:	2	Linux	2
		Pleno	1,5	Francês	Baixo	3	Oracle Certified Expert, Java EE 6 Web Component Developer	3		
		Senior	2	Alemão			Oracle Certified Expert, Java EE 6 Enterprise JavaBeans Developer	4		
		Master	3				Oracle Certified Expert, Java EE 6 Java Persistence API Developer	5		
							Zend PHP 5 (Zend PHP 5.3 Certification)	6		
							Oracle Certified Associate, MySQL 5	9		
							Oracle Certified Professional, MySQL5 Database Administrator	9,5		
							MCSA: SQL Server	10		

Utilitários	SIM	Facilidade de Negociação	SIM	Capacidade de Liderança	SIM	Facilidade de Comunicação	SIM	Áreas de Conhecimento	SIM
JavaFX	1	Ótimo	1	Ótimo	1	Ótimo	1	Contabilidade	1
Padrão MVC	5	Regular	2	Regular	2	Regular	2	Finanças	2
		Baixa	3	Baixa	3	Baixa	3	Logística	3
								Agronegócio	7
								Comercio Exterior	6
								Direito	5
								Relações Internacionais	5
								Física	4
								Estatística	4,5
								Administração	5
								Ciências Exatas	3,5

Capacidade de Lidar com Mudanças	SIM	Facilidade de Trabalhar em equipe	SIM	Area preferencial de atuação	SIM
Ótimo	1	Ótimo	1	Desenvolvedor	2
Regular	2	Regular	2	Gerente	5
Baixa	3	Baixa	3	Tester	3
				Analista de Requisitos	4
				Suporte	1

Conhecimento de Redes	SIM	Cloud Computing	SIM	Conhecimentos de Cloud	SIM	Formação	SIM
Ótimo	1	Ótimo	1	SaaS	1	Ciências da Computação	4
Regular	2	Regular	2	PaaS	2	Engenharia da Computação	5
Baixa	3	Baixa	3	IaaS	3	Sistemas da Informação	3
						Cursos Tecnólogos	2
						Cursos Técnicos	1

Figura 10 – Listagem de Características e Similaridade

No quadro são listados as tabelas de características por fatores, cada característica tem um peso específico que pode ser maior ou menor em um nível hierárquico. Segue no tópico abaixo a relação de cada características e suas similaridades.

3.3.2 Características – Similaridades

Características Humanas

As características Línguas, Facilidade de Comunicação, Facilidade de Negociação, Capacidade de Lidar com Mudanças, Facilidade de Trabalhar em Equipe, são casos que apresentam aspectos iguais, todas tem um nível hierárquico de qualidade distribuídos em 3 que são Ótimo, Regular e Ruim. As similaridades foram definidas em igualdade em todas as características sendo da seguinte forma. 1 Para Ótimo, visto que Ótimo é próximo de Regular, ao qual foi atribuído o valor 2, entretanto está distante de Ruim que possui o valor 3, ao qual está próximo de Regular que possui valor 2.

Linguagem de Programação primária e secundária.

Com relação a Linguagem de programação foi definida com base na lista de programações mais utilizadas e padrões de desenvolvimento utilizados nessas linguagem como o MVC. Tudo isso foi definido como sendo proximidades entre as linguagens visto que se comparada a sintaxe a mesmas podem vir a ser diferentes, mas alguns padrões para utiliza não. As linguagens de programação secundária foi definida suas similaridade pelo uso de TAGs encontradas tanto em JavaScript e HTML, entretanto todas as 3 podem ser utilizadas em conjunto isso se dá pela melhora na interface gráfica por isso estão entre as mais utilizadas como se verifica na figura 4.

Banco de Dados

Na similaridade entre os banco de dados foi levado em conta a tabela de ranking que se encontra no tópico de banco de dados e fatores como o tipo de SGBD, levando em conta o que cada tipo de banco propõem.

Conhecimento de Redes.

Cloud Computing, Conhecimento de Cloud, Conhecimento de Redes são listados também como base hierárquica de conhecimento sendo 1, 2 e 3 para Ótimo, Regular, Ruim.

Áreas de Conhecimento e Certificações.

Nas características de Certificação foi baseado na linguagem em que a pessoa se certifica, as certificações tem sua relação na linguagem de programação e hierarquia, no caso das certificações Java tem suas proximidades no que diz respeito ao nível do certificado que a pessoa possui um exemplo disso são as certificações Java EE ou Enterprise Edition que são vinculadas a programação Web por isso suas similaridades são bem próximas. Com relação a área de conhecimento foi utilizado como base a linguagem matemática, isto é, as características que possuem em sua base a matemática, essas são mais similares que as que possuem conhecimento do direito, relações internacionais e Comércio Exterior.

Níveis de Programador

Conforme a figura 9 os níveis de programador tem suas similaridades com base na hierarquia dos níveis de programação, quanto maior o nível de programador menor similaridade ele tem com as características mais abaixo, por exemplo, o nível programador Master tem uma maior similaridade com Pleno e Sênior do que com o Junior e Trainee.

Formação

Suas similaridades são feitas a partir da dificuldade e valor da certificação que os cursos. Por isso não foi dada uma diferença muito grande nos cursos com Bacharel entretanto o nível de tecnólogo e Bacharel se pode salientar.

Frameworks

A similaridade entre framework Hibernate para CodeIgniter e AngularCLI é bastante alta pois enquanto as outras são voltadas para a programação WEB, o Hibernate é um Framework de Mapeamento Objeto-Relacional como demonstra o tópico Hibernate.

Servidores de Aplicação e Web Container

Conforme citado no t3pico de Servidores de Aplica33o e Web Container os servidores listados s3o bastantes similares mantendo algumas particularidades e caracter3sticas individuais mas por serem bastante parecidos suas similaridades tamb3m se assemelham.

Sistemas Operacionais.

Os sistemas operacionais s3o os mais utilizados nas empresas conforme o t3pico de linguagem de programa33o por isso tem sua similaridade aproximada, outro fator que pode a vir a ter import3ncia 3 a similaridade de interface entre eles.

Utilit3rios

Como listados nos t3picos respectivos de cada caracter3stica fica n3tido a falta de semelhan3a entre as caracter3sticas por isso a similaridade entre as mesmas est3o altas.

3.4 Recupera33o de casos

Se d3 pela busca na base de casos por casos similares ao problema proposto, tendo em vista, suas similaridades, com base em suas caracter3sticas. No fim da aplica33o da t3cnica retorna n+1 casos poss3veis, ou at3 mesmo nenhum caso n3o encontre similaridade alguma.

As similaridades tanto local quanto global foi tratada no decorrer da se33o abaixo 3.2.1 e 3.2.2 exemplificando de modo mais amplo suas import3ncias.

3.4.1 Similaridade Local

An3lise de pares de caracter3sticas individualmente, tendo o seu valor variante entre 0 e 1. A f3rmula utilizada para realizar a medi33o e a obten33o do resultado da similaridade local 3 vista de acordo com Von Wangenheim e Von Wangenheim (2003) pela seguinte equa33o:

$$\{C_1, C_2, C_3 \dots \dots, C_k\}$$

- Em que V 3 a similaridade local entre dois termos v_i, v_j ;
- v_{max} 3 o maior valor que aquele termo pode se estender nas caracter3sticas; e
- v_{min} o menor valor que o termo pode assumir.

Para se definir a proximidade entre os termos decorrentes o valor varia entre os 2 números citados, tendo em vista quanto maior e mais próximo de 1 o valor está maior é sua proximidade, evidentemente, quanto menor o valor se aproxima de 0 menor é sua similaridade.

Tendo obtido a similaridade local com base em cada termo individual ainda se faz necessário realizar o cálculo de similaridade global que está na próxima seção.

3.4.2 Similaridade Global

A similaridade global diferentemente da similaridade local se

$$\frac{\sum_{i=1}^n f(Q_i, C_i) \times w_i}{n}$$

$(Q_i, C_i) =$

A similaridade local entre Q_i, C_i é representada por $f(Q_i, C_i)$, em que Q_i é o valor do termo da resposta de entrada e C_i é o valor do mesmo termo de um caso armazenado na base. Essa similaridade é multiplicada pelo peso w_i e o resultado é dividido pelo soma de todos os pesos, representado por n sendo está a quantidade dos pesos. O peso dos termos neste projeto foi estimado por meio do tf-idf apresentado anteriormente. Com base nesse cálculo seu resultado é a média dos valores da similaridade local, tendo em vista seu pré-processamento.

3.5 Retenção

Processo que são retidos os casos com suas soluções para casos futuros, tornando o sistema cada vez mais inteligente, quanto mais o sistema incorporar conhecimento, mais chances o mesmo possui de solucionar os problemas do novo caso.

Os sistemas de retenção de casos são 3 aos quais serão explicados para se verificar qual o melhor abordagem na construção desta aplicação.

Sem retenção de casos:

Busca o desenvolvimento com protótipos para um conjunto de sintomas relevantes no domínio da aplicação, tendo em vista que o sistema não precisa incorporar novos casos.

Retenção de soluções de problemas:

Uma forma de aprendizado comum nos RBCs, aos quais passam por um processo de aprendizado do conhecimento com base no armazenamento de casos. Segundo (Hinrichs, 1992) ou (Hammond, 1986) “alguns sistemas retêm casos representando tentativas de solução de problemas que falharam, para permitir que se evite utilizar aquelas soluções no futuro.”

3.6 Ferramenta De Alocação de Pessoas em Projetos

Para ratificar e validar os resultados aqui apresentado realizou-se uma ferramenta para se apresentar a seleção e verificação dos cálculos mostrando os resultados com base no que foi selecionado.

Segue a tela de seleção de características apresentadas neste trabalho.

The screenshot shows a window titled "Alocação De Pessoas Em Projetos" with a light blue background. It contains two columns of dropdown menus. The left column includes: Linguagem de Programação: Java; Banco De Dados: MySQL; FrameWork: Hibernate; Servidores de Aplicação: Tomcat; Níveis de Programador: Junior; Relatórios: iReports; Línguas: Inglês - Intermediário; Certificação: Oracle Certified Associ...; Liderança: Ruim; Área de Conhecimento: Ciências Exatas; Formação: Sistemas da Informação. The right column includes: Utilitários: JavaFX; Facilidade de Negociação: Selecione; Capacidade de Lidar com Mudança: Otimo; Facilidade de Trabalhar em Equipe: Otimo; Área preferencial de atuação: Desenvolvedor; Conhecimento de Redes: Selecione; Conhecimento de Cloud: Selecione; Sistemas Operacionais: Windows; Facilidade de Comunicação: Otimo; Conhecimentos Especificos Cloud: Selecione; Linguagem Programação Secundária: CSS. At the bottom right, there are two buttons: "Próximo" and "Sair".

Figura 11 – Seleção das Características

A tela acima apresentada se pode verificar quais as opções de características o usuário pode selecionar, podendo-se perguntar, quais as características importantes para se fazer neste projeto específico? Ao Selecionar e a posterior clicar no botão “Emitir Relatório” o usuário verifica as opções de membro da equipe com base nas características selecionadas.

Foi adicionado 5 membros de equipe a base de casos e com base na similaridade apresentada se pode verificar qual é mais similar ao problema proposto pelo usuário. Os termos de **Linguagem de Programação** e **Linguagem de Programação Secundária** apesar de a mesma apresentar o mesmo sentido, estando presente ambas em um ambiente de programação, foi dividido em 2 caixas de seleção pois uma tende a complementar a outra, tendo em vista, citando-o já como exemplo, Java – CSS, PHP – CSS.

Essa processo de exemplificação tem por objetivo tentar encontrar membros de equipe em um ambiente de programação voltado a programação Java, como se

pode salientar na figura 10 apresentada acima as características foram selecionadas levando em conta essa premissa. Ao clicar em “Próximo” o usuário vai para a tela de definição dos pesos a qual se encontra abaixo na figura 11.

The screenshot shows a window titled "Definição de Pesos" with the subtitle "Definição de peso dos fatores". It contains a list of 20 factors, each with an input field. The factors are arranged in two columns. The first column includes: Linguagem de Programação, Banco De Dados, FrameWork, Servidores de Aplicação, Níveis de Programador, Relatórios, Linguas, Certificação, Liderança, Área de Conhecimento, and Formação. The second column includes: Utilitários, Facilidade de Negociação, Capacidade de Lidar com Mudança, Facilidade de Trabalhar em Equipe, Área preferencial de atuação, Conhecimento de Redes, Conhecimento de Cloud, Sistemas Operacionais, Facilidade de Comunicação, and Conhecimentos Específicos Cloud. The input fields for "Facilidade de Negociação", "Conhecimento de Redes", "Conhecimento de Cloud", and "Conhecimentos Específicos Cloud" are disabled (grayed out). A button labeled "Emitir Relatório" is positioned at the bottom right of the window.

Figura 12 – Listagem dos Pesos

Em relação a definição dos pesos, esta tabela foi utilizada para a listagem no que o usuário deseja ser mais relevante na pesquisa. Alguns campos podem não estar selecionados pelo fato de que o usuário não desejou que aquela determinada característica não fizesse parte da pesquisa, sendo para ele a característica desabilitada não é de fato relevante.

Por fim, tem-se a última etapa da exibição de resultados, local em que os resultados aplicados são demonstrados, ao clicar no botão “**Emitir Relatório**” o usuário pode verificar quais os candidatos são melhores a determinada vaga pré defenida por ele. Segue abaixo na **figura 12** a tela de exibição de resultados encontrados.

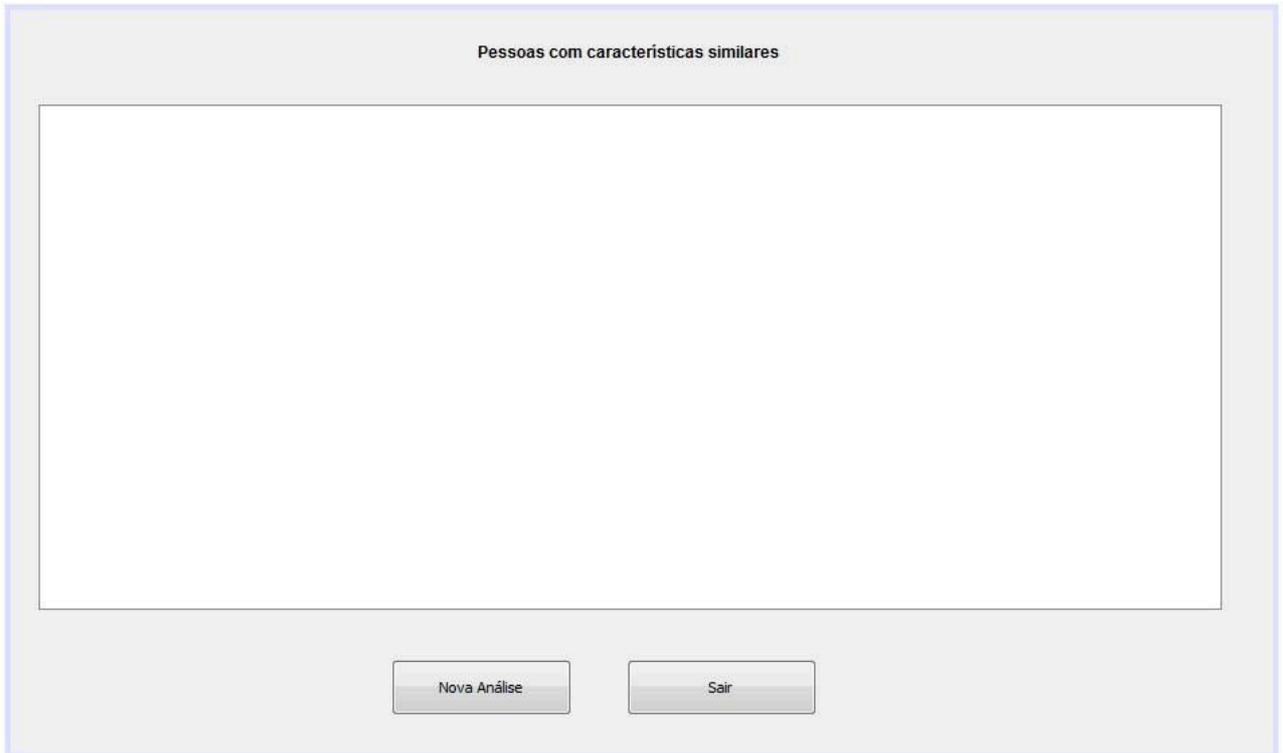


Figura 13 – Tela Resultados Encontrados.

A tela acima se encontra o painel é o local em que foi exibido o resultado como exemplo de testes segue abaixo o resultado de um teste de desenvolvedor voltado para a área de java:

Nome: Funcionario 01 CPF: 123456789

VS Local Linguagem de Programação: 1.0

VS Local Banco de dados: 0.0

VS Local Framework: 0.800000011920929

VS Local Servidor de Aplicação e Web Container: 0.0

VS Local Relatórios: 1.0

VS Local Níveis de Programador: 0.800000011920929

VS Local Linguas: 1.0
VS Local Certificação: 1.0
VS Local Sistema Operacional: 0.0
VS Local Utilitários: 1.0
VS Local Facilidade de negociação: 1.0
VS Local Capacidade de liderança: 0.6600000262260437
VS Local Facilidade de Comunicação: 0.33000001311302185
VS Local Áreas de conhecimento: 0.0
VS Local Capacidade de lidar com mudanças: 0.6600000262260437
VS Local Facilidade de trabalhar em equipe: 0.6600000262260437
VS Local Área preferencial de atuação: 0.0
VS Local Formação: 0.0
VS Linguagem de programação secundária: 0.0

VS Global: 46%

Nome: Funcionario 02 CPF: 123456789

VS Local Linguagem de Programação: 0.800000011920929
VS Local Banco de dados: 0.0
VS Local Framework: 1.0
VS Local Servidor de Aplicação e Web Container: 1.0
VS Local Relatórios: 1.0
VS Local Níveis de Programador: 0.6000000238418579
VS Local Linguas: 0.0
VS Local Certificação: 1.0
VS Local Sistema Operacional: 1.0
VS Local Utilitários: 0.0
VS Local Facilidade de negociação: 0.6600000262260437
VS Local Capacidade de liderança: 0.6600000262260437
VS Local Facilidade de Comunicação: 0.6600000262260437
VS Local Áreas de conhecimento: 0.0

VS Local Capacidade de lidar com mudanças: 0.6600000262260437
VS Local Facilidade de trabalhar em equipe: 0.33000001311302185
VS Local Área preferencial de atuação: 0.0
VS Local Formação: 0.0
VS Linguagem de programação secundária: 0.0

VS Global: 50%

Nome: Funcionario 03 CPF: 123456789

VS Local Linguagem de Programação: 1.0
VS Local Banco de dados: 0.0
VS Local Framework: 0.0
VS Local Servidor de Aplicação e Web Container: 0.0
VS Local Relatórios: 0.0
VS Local Níveis de Programador: 0.0
VS Local Linguas: 0.0
VS Local Certificação: 1.0
VS Local Sistema Operacional: 1.0
VS Local Utilitários: 0.0
VS Local Facilidade de negociação: 0.0
VS Local Capacidade de liderança: 0.0
VS Local Facilidade de Comunicação: 0.0
VS Local Áreas de conhecimento: 0.0
VS Local Capacidade de lidar com mudanças: 0.0
VS Local Facilidade de trabalhar em equipe: 0.0
VS Local Área preferencial de atuação: 0.0
VS Local Formação: 1.0
VS Linguagem de programação secundária: 0.0

VS Global: 27%

Nome: Funcionario 04 CPF: 123456789

VS Local Linguagem de Programação: 1.0
VS Local Banco de dados: 0.0
VS Local Framework: 1.0
VS Local Servidor de Aplicação e Web Container: 0.0
VS Local Relatórios: 0.0
VS Local Níveis de Programador: 0.0
VS Local Linguas: 0.0
VS Local Certificação: 0.33000001311302185
VS Local Sistema Operacional: 0.0
VS Local Utilitários: 0.0
VS Local Facilidade de negociação: 0.0
VS Local Capacidade de liderança: 0.0
VS Local Facilidade de Comunicação: 0.0
VS Local Áreas de conhecimento: 0.0
VS Local Capacidade de lidar com mudanças: 0.0
VS Local Facilidade de trabalhar em equipe: 0.0
VS Local Área preferencial de atuação: 0.0
VS Local Formação: 0.0
VS Linguagem de programação secundária: 0.0

VS Global: 15%

Nome: Funcionario 05 CPF: 123456789

VS Local Linguagem de Programação: 0.6600000262260437
VS Local Banco de dados: 1.0
VS Local Framework: 0.0
VS Local Servidor de Aplicação e Web Container: 0.0

VS Local Relatórios: 0.0
VS Local Níveis de Programador: 0.0
VS Local Linguas: 0.0
VS Local Certificação: 0.6600000262260437
VS Local Sistema Operacional: 0.0
VS Local Utilitários: 0.0
VS Local Facilidade de negociação: 1.0
VS Local Capacidade de liderança: 0.6600000262260437
VS Local Facilidade de Comunicação: 1.0
VS Local Áreas de conhecimento: 1.0
VS Local Capacidade de lidar com mudanças: 0.6600000262260437
VS Local Facilidade de trabalhar em equipe: 1.0
VS Local Área preferencial de atuação: 0.0
VS Local Formação: 0.0
VS Linguagem de programação secundária: 0.0

VS Global: 42%

O usuários que possuem características 1.0 é porque estão de acordo com as características que foram propostas a realizar o trabalho, outros tendem a estar parcialmente similar ao que foi proposto, já as características que apresentam 0.0 é porque não possuem similaridade nenhuma com o que o usuário deseja.

4 Conclusão e Trabalhos Futuros

De acordo com o que foi desenvolvido o sistema pode se comportar de acordo com o domínio apresentado, essa nova técnica pode ser encontrada em outros domínios como na área médica, florestal entre outras. Contudo nesse projeto foi testada e validada trazendo resultados de acordo com as habilidades dos membros, ao qual pode ser aprimorada para se transformar em uma ferramenta gerencial de empresas no que diz respeito ao controle de qualidade tanto da equipe e consequentemente do produto que irá ser construído.

Levando em conta o projeto apresentado se pode salientar que o mesmo está estático não podendo ser alterado, estando fixado em um conjunto de características pré-determinada, o que ainda pode ser evoluído dinamicamente sendo a critério do cliente a realização dessa coleta e transformação da informação, além disso, pode-se também expandir o que já foi realizado coletando novas características para novas funções dentro da organização em que o sistema está.

Referências

ABES; Mercado Brasileiro de Software. Disponível em <

<http://www.abessoftware.com.br/dados-do-setor/dados-2014> > Acesso 26 out. 2016.

AAMODT, Agnar; PLAZA, Enric. Case-based reasoning: foundational issues, methodological variations, and system approaches. *Ai Communications*, London, p. 39-59. 01 mar. 1994.

ALENCAR, Aretha Barbosa. Visualização da evolução temporal de coleções de artigos científicos. 2013. 157 f. Tese (Doutorado) - Icmc Usp, São Carlos, 2013.

ALMEIDA, Bruno. Carreira de TI em outras áreas do conhecimento. Disponível em <https://www.profissionaisiti.com.br/2014/02/carreira-de-ti-em-outras-areas-do-conhecimento/>, 2014. Acesso 26 Jul. 2017.

AVOYAN, Hovhannes; As melhores plataformas PHP para 2015. Disponível em <<http://imasters.com.br/linguagens/php/as-melhores-plataformas-php-para-2015/?trace=1519021197&source=single>> Acesso 28 out. 2016

ARLOT, Sylvain; CELISSE, Alain. A survey of cross-validation procedures for model selection. Lille: Irma, 2010.

BARION, Eliana Cristina Nogueira; LAGO, Decio. Mineração de textos: text mining. *Revista de Ciências Exatas e Tecnologia*, Valinhos, n. , p.123-140, 08 dez. 2008.

BARRETO, Ailton Silva; DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO. Rio de Janeiro; 2005.

BAUER, C; King G. (2007) “Java Persistence com Hibernate” 1º Edição, Editora Ciência Moderna Ltda. Rio De Janeiro, Brasil.

BELL, Timothy C.; MOFFAT, Alistair; WITTEN, Ian H.. *Managing gigabytes: compressing and indexing documents and images*. San Francisco: Morgan Kaufmann, 1999.

BIASOLI, Flavio (2016) 5 problemas graves nas equipes de trabalho Disponível em< <http://blog.prolancer.com.br/problemas-graves-nas-equipes-trabalho/>> Acesso em: 08 Dez. 2016

BUENO, KJ O que é JSF (Java Server Faces). Disponível em:

< <http://fabrica.ms.senac.br/2013/06/o-que-e-jsf-java-server-faces/>>. Acesso em: 27 out. 2016.

CASANOVA, Marco Antonio; MOURA, Arnaldo Vieira. Princípios de Sistemas de Gerenciamento de Banco de Dados Distribuídos. Edição Revisada. 1999. Disponível em<<http://www.inf.puc-rio.br/~casanova/Publications/Books/1985-BDD.pdf>> Acesso em 28 out. 2016

CRUZ, C. D. Aplicação de algumas técnicas multivariadas no melhoramento de plantas. 1990. Tese (Doutorado) – ESALQ, Piracicaba, 1990.

DB-Enginner; DB-Enginner Ranking. Disponível em:<db-engines.com/en/ranking> Acesso 28/10/2016

FELDMAN, Ronen; HIRSH, Haym. Exploiting Background Information in Knowledge Discovery from Text. Journal Of Intelligent Information Systems, v. 9, n. 1, p.83-97, 01 jul. 1997.

FELDMAN, R.; SANGER, J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge, MA: Cambridge University Press, 2007.

FELDMAN, Ronen; SANGER, James. The Text Mining Handbook:Advanced Approaches in Analyzing Unstructured Data. New York: Cambridge University Press, 2007.

GASPARINI, Claudia. Estes são os salários no Brasil para 8 áreas de TI: Estudo da Robert Half mostra a remuneração para áreas como desenvolvimento de software, big data e telecomunicações.. Exame, São Paulo, 27 out. 2016. Disponível em: <<http://exame.abril.com.br/carreira/estes-sao-os-salarios-no-brasil-para-8-areas-de-ti/>>. Acesso em: 27 out. 2016.

GRILLO, Marlene Corroero; GESSINGER, Rosana Maria. Por que falar ainda em avaliação? Porto Alegre: Universitária Pucrs, 2010.

GOMES, Fabio Naveca; QUESADO, Eduardo Figueiras. Estudo Comparativo Entre Ferramentas Gratuitas Versus Proprietárias em um Processo de Desenvolvimento de Software Orientado a Objetos - Centro Universitário Luterano de Manaus, 2003.

HALMENSCHLAGER, Carine. Um algoritmo para indução de árvores e regras de decisão. 2002. 112 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

HINRICHS, T.R Problem solving in open worlds; a case study in design. Northvale, NJ: Erbaum, 1992;

IWEB; Servidores de Aplicação. Disponível em: iweb.com.br/iweb/pdfs/20031008-ap Acesso em: 28 out. 2016

JAX ENTER EDITORIAL; Why Java skills are still Golden for developers Disponível em: <<https://jaxenter.com/why-java-skills-are-still-golden-for-developers-107382.html>> Acesso em: 28 out. 2016

JOYANES, AGUILAR Luiz La Cloud Computing (Computação em Nuvem): O novo paradigma tecnológico para as empresas e organizações na Sociedade do Conhecimento

KEARNS, Michael; RON, Dana. Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. New York: Acm, 1997.

KENGI, Igor Shiohara Conhecendo as principais certificações Java. Disponível em: <<http://www.matera.com.br/2015/06/26/conhecendo-as-principais-certificacoes-java/>>. Acesso em: 27 out. 2016.

KOHAVI, Ron. A Study of CrossValidation and Bootstrap for Accuracy Estimation and Model Selection. Stanford: Ijcai, 1995.

K.J. Hammond, R. Burker, K.Schimitt. A Case-Based Approach to Knowledge Navigation. In D. Leake (ed.), Case-Based Reasoning Experiences: Lessons Learned & Future Direction 1996.

Kolodner, Janet (1993). *Case-Based Reasoning* (em inglês) (San Mateo, CA: Morgan Kaufmann). p. 8.

KOSLOSKY, Marco Antônio Neiva. Aprendizagem baseada em casos: um ambiente para ensino de lógica de programação. 1999. 2 v. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1999.

MAUKY, Manuel. JavaFX vs Swing vs HTML5: Where does JavaFX stand in relation to other UI toolkits. Disponível em < <https://jaxenter.com/javafx-compared-to-other-ui-toolkits-like-swing-html5-and-swt-a-resume-124484.html>> Acesso: 28 out. 2016

MENGATO, Ronaldo Cesar Junior. Modelo de Apoio de equipes em Projetos de desenvolvimento de software, 2014.

MEDEIROS, Higor. Introdução ao Padrão MVC Disponível em: <<http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308/>>. Acesso em: 27 out. 2016

MEDEIROS, Higor. Utilizando construtores privados em classes Utilitárias em Java Disponível em < <http://www.devmedia.com.br/utilizando-construtores-privados-em-classes-utilitarias-em-java/27918>>. Acesso 23 Jun. 2017

MENDES, Marcio Almeida. Sistematização da Assistência de Enfermagem usando Raciocínio Baseado em Casos implementado em JAVA. 2009. 69 f. Dissertação (Mestrado) - Icmc Usp, São Paulo, 2009.

MIRANDA, Willian; A importância do inglês para um profissional de Ti Disponível em<<http://aprendaplsql.com/2015/02/importancia-ingles-para-um-profissional-de-ti/>> Acesso em 01 out. 2016

MCKILLIP, Jack, COX, Caryl. Strengthening the criterion-related validity of professional certifications. Evaluation and Programming Planning.

MORAIS, Edison Andrade Martins; AMBRÓSIO, Ana Paula L..Mineração de textos. Goiânia: Universidade Federal de Goiás, 2007.

MRACK, M. (2006) “Hibernate, Uma visão geral sobre o framework padrão de fato para mapeamento objeto-relacional” Disponível em: <<http://www.slideshare.net/mmrack/workshop-hibernate-com-comentarios>> Acesso. 27 out. 2016

MOURA, Nara (2014) Os impactos da insatisfação no trabalho sobre as organizações. Disponível em: <<http://www.administradores.com.br/artigos/negocios/os-impactos-da-insatisfacao-no-trabalho-sobre-as-organizacoes/75033/>> Acesso: 08 Dez. 2016.

PROJECT MANAGEMENT INSTITUTE – PMI. Guide of Project Management Body of Knowledge - PMBoK, 2000.

POZZEBON, Rafaela Qual a diferença entre cargo júnior, pleno, sênior, master e especialista? Disponível em: <https://www.oficinadanet.com.br/artigo/carreira/qual_a_diferenca_entre_cargo_junior_pleno_senior_master_e_especialista>. Acesso em: 27 out. 2016.

RODRIGO Waltenberg (2016) “AngularJS: O que é e porquê utilizar” Disponível em<<http://blog.algaworks.com/o-que-e-angularjs/>>

SIMPOSIO BRASILEIRO DE BANCO DE DADOS; Topics Of Interest Salvador Bahia. Disponível em<sbbd2016.fpc.ufba.com.br/chamadas-de-artigos> Acesso em: 28/10/2016

SILVA, Valdinei Freire da et al. RECONHECIMENTO DE ESCRITA BASEADO EM REDES NEURAIS ARTIFICIAIS UTILIZANDO B-SPLINES E TDF. Bauru: Escola Politécnica da Universidade de São Paulo, 2003.

SOMMER, Robert; SOMMER, Barbara Baker. A practical guide to behavioral research: tools and techniques. New York: Oxford University Press, 2002.

SOUZA, G.S., 2003, Representação da Distribuição do Conhecimento, habilidades e experiências através da estrutura organizacional, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

TAN, Pang-ning; STEINBACH, Michael; KUMAR, Vipin. Introduction to data mining. Minnesota: Addison-wesley, 2006.

TAN, A. Text Mining: The State of the Art and the Challenges. In: Workshop on Knowledge Discovery from Advanced Databases (PKDAD'99), 1999. Beijing. Proceedings. Beijing: 1999, p.71-76.

SHTUB,A. ; BARD, J. F. ; GLOBERSON, S. , 1994Project Management Engineering, Technology and Implementation, Prentice Hall, New Jersey.

TELLES, Viviane Carra et al. Sistema de Raciocínio Baseado em Casos para Recomendação de Programa Alimentar. Pelotas: Resi, 2006.

VICINI , Analise Multivariada da Teoria à Pratica. 2005

VON Wangenheim, Christiane Gresse; VON Wangenheim, Aldo.Raciocínio baseado em casos. Barueri: Manole, 2003.

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, 1999.

Watson, Ian (1997). *Applying Case-Based Reasoning*. Techniques for Enterprise Systems (em inglês) (San Francisco, CA: Morgan Kaufmann). p. 19

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M.C.; REGNELL, B.; WESSLÉN, A., Experimentation in Software Engineering: an Introduction. Kluwer Academic Publishers, Massachusetts, 2000.