**ANDRESSA FARIA**

**EDILSON JUSTINIANO**

**BUSCA POR MÃO DE OBRA QUALIFICADA, UTILIZANDO BANCO DE DADOS ORIENTADO A GRAFOS**

**UNIVERSIDADE DO VALE DO SAPUCAÍ**

**POUSO ALEGRE**

**2015**

**ANDRESSA FARIA**

**EDILSON JUSTINIANO**

**BUSCA POR MÃO DE OBRA QUALIFICADA, UTILIZANDO BANCO DE DADOS ORIENTADO A GRAFOS**

Projeto de pesquisa apresentado à banca de qualificação do curso de Sistemas de Informação como requisito para o desenvolvimento do Trabalho de Conclusão de Curso sob a orientação do prof. Márcio Emílio Cruz Vono de Azevedo.

**UNIVERSIDADE DO VALE DO SAPUCAÍ**

**POUSO ALEGRE**

**2015**

SUMÁRIO

INTRODUÇÃO 3

2 OBJETIVOS 7

2.1 Objetivo geral 7

2.2 Objetivos específicos 7

3 JUSTIFICATIVA 8

4 QUADRO TEÓRICO 9

4.1 Iconix 9

4.2 Tecnologias 11

4.2.1 Java 11

4.2.2 Tomcat 7 12

4.2.3 Neo4j 14

4.2.4 Cypher Query Language 16

4.2.5 JavaServer Faces 17

4.2.6 Primefaces 18

5 QUADRO METODOLÓGICO 20

5.1 Tipo de pesquisa 20

5.2 Contexto 20

5.3 Participantes 21

5.4 Instrumentos 22

5.5 Formulários de pesquisa 22

5.6 Reuniões 22

5.7 Análise documental 22

5.8 Procedimentos 23

5.9 Cronograma 24

5.10 Orçamentos 25

6 REFERÊNCIAS 26

# INTRODUÇÃO

De acordo com Sanches (2002) os primeiros bancos de dados relacionais foram desenvolvidos nas décadas de 60 e 70 dentro dos laboratórios da IBM. As empresas daquela época perceberam que, manter vários funcionários para realizar algumas tarefas como armazenamento e organização de documentos estava se tornando altamente custoso para elas. Por isto a IBM decidiu investir em pesquisas e desenvolvimentos nesta área a fim de gerar uma forma mais simples, mais barata e mecânica de realizar tais tarefas.

Ainda segundo Sanches, desde suas criações os bancos de dados veem sofrendo constantes melhorias. Isto graças aos feedbacks de usuários que os utilizam.

Com o advento da internet, e mais tarde sua popularização, uma série de aplicações para diferentes fins foram desenvolvidas. Grande parte destas aplicações utilizam bancos de dados para armazenar suas respectivas informações. Aplicações como: serviços de música, serviços de busca (este não se restringindo apenas ao Google), serviços de geolocalização, entre outros. Porém um dos mais populares entre todos os serviços são sem dúvida as redes sociais. Segundo a pesquisa de Goes (2013) que apresenta a lista de sites mais acessados no ano de 2013 em todo o mundo, o facebook (rede social mais famosa do mundo) aparece no topo da lista sendo o site mais acessado em todo o mundo.

Este sucesso das redes sociais está diretamente ligado ao modelo de vida que seguimos hoje em dia. Hoje, sabemos que tudo e todos estão conectados direto ou indiretamente e que as informações são trocadas em uma velocidade surpreendente. Precisamos obter informações de forma mais clara possível, além é claro do quanto antes. E como as redes sociais possuem facilidades para manter conectados os usuários com seus familiares e amigos, inclusive fazer novas amizades e facilitar a comunicação entre todos estes usuários, além é claro de possuir entretenimento, promoções de produtos, entre outros fatores fizeram das redes sociais o sistema mais utilizado em todo o mundo como mostra a pesquisa citada anteriormente. (Barbosa, 2011).

Segundo Daniel Negreiros (diretor de Criação da agência de comunicação Plan B), “o crescimento do uso da internet móvel é o principal responsável pela expansão da presença dos brasileiros nas redes porque esses aparelhos funcionam como se fossem um computador à mão em qualquer hora e em quase todo lugar.”. Podemos afirmar então que um dos principais motivos para tamanho popularização deste serviço foi a criação dos smartphones, pois os mesmos somado a evolução das redes de dados 3G e 4G possuem a capacidade de conectar as pessoas em qualquer lugar e a qualquer momento bastando alguns toques, tornando assim ainda mais fácil localizar pessoas e se comunicar com elas.

Segundo a pesquisa realizada pela consultoria IDC (empresa que realiza pesquisas na área de TI e Telecom), no primeiro semestre de 2014, 4,2 milhões de tablets foram comercializados no Brasil, contra 3,4 milhões em igual período de 2013. Já a venda de smartphones bateu recorde no segundo trimestre de 2014 em comparação com abril, maio e junho de 2013, avançando para 13,3 milhões de unidades um avanço de 22% no período.

De acordo com esta pesquisa é possível perceber o quanto o mercado móvel no brasil veem crescendo nos últimos anos ultrapassando até mesmo o número de vendas de notebooks no pais e este aumento se reflete diretamente no aumento de usuários cadastrados em redes sociais.

Com este grande aumento da popularização das redes sociais, foi necessário repensar alguns pontos do sistema, o que é natural ao longo da vida de um software uma vez que o mesmo inicialmente não fora projetado para tamanho quantidade de dados. Alguns pontos que são necessários revisar são: otimização, escalabilidade, manutenção, entre outros pontos.

Este aumento somado a evolução dos *smartphones* em conjunto com aumento do acesso da população mundial a internet se refletiu na quantidade de conteúdos publicados na *web.* Isto fez com que uma nova era para os bancos de dados surgisse. Uma nova geração foi criada, denominada banco de dados NOSQL[[1]](#footnote-1). Esta nova geração foi criada devido a frustração pelas limitações desempenho e de modelagem que o bancos de dados relacionais possuíam. Limitações que levam a perda de desempenho em algumas operações devido a complexidade e o volume de dados, dificultando assim o gerenciamento e manipulação dos dados, o que acaba interferindo diretamente nas informações geradas pelo s*oftware*,impossibilitando assim em alguns casos o sucesso do *software*.

A informação está em constante mudança. Não possuindo mais o mesmo padrão que anteriormente possuíam. Estas mudanças se refletem diretamente nas bases de dados sendo necessárias constantes mudanças, porém em alguns casos as mudanças se refletem em redundância de dados a fim de obter uma maior agilidade em suas buscas. Segundo Heuser (1998, pág. 14) há dois tipos de redundância de dados: o primeiro é conhecido como redundância controlada de dados e ocorre quando o software possui o conhecimento de tal redundância e garante a sincronia entre elas quando necessário. O segundo tipo é a redundância não controlada de dados que ocorre quando a responsabilidade da sincronização dos dados está com o usuário e não com o software. Quando isto ocorre a base de dados começa a perder sua integridade e isto acarreta uma série de problemas que pode vir se tornar extremamente prejudicial ao sistema uma vez que os dados que o mesmo apresentará não mais serão os dados corretos. Antes que isto ocorra, deve-se repensar toda a base de dados a fim de evitar tais problemas não permitindo que o *software* apresente aos usuários informações inconsistentes e inválidas.

As redes sociais a cada dia desejam extrair o maior número informações de seus usuários a fim de utilizá-las de maneira mais eficiente possível. Isto implica na mudança constante na base de dados uma vez que as informações são recebidas de diferentes fontes como: *smartphones*, *tablets*, relógios inteligentes, computadores pessoais etc. Sendo possível conter diferentes informações com um mesmo fim de acordo com o tipo de acesso do usuário. Por exemplo: através de um computador pessoal é possível informar um endereço cujo, uma foto foi tirada ao contrário de um dispositivo móvel, que é possível informar o endereço ou enviar as coordenadas do lugar cujo, a foto foi tirada. Isto nos mostra o quanto as informações estão em constantes mudanças. Entretanto, tais informações devem estar coesas. Por este motivo muitas redes sociais como twitter, facebook, linkedin optaram pela migração de seus antigos bancos de dados relacionais para banco de dados orientado à grafos a fim de otimizar suas consultas a ele e evitar possíveis problemas com redundância de dados mencionados anteriormente.

Barreto (1996, p. 2) afirma que informação é: "estruturas significantes com a competência de gerar conhecimento no indivíduo, em seu grupo, ou à sociedade”. Podemos então compreender o motivo pelo qual as redes sociais desejam extrair tamanho quantidade de informações de seus usuários.

Para Gonçalves; Gouveia; Petinari (2008, p. 45)

Uma informação solta, não contextualizada, não é um produto. Para que seja reconhecida e compreendida como produto, primeiramente, precisa estar focada em determinada direção e objetivo. Então, depois de direcionada, a informação será um produto e poderá, como qualquer outro nas mesmas condições, alcançar o seu sentido natural, o motivo pelo qual existe, ou seja, aumentar o grau de certeza em uma tomada de decisão.

Diante destas afirmações, certamente podemos dizer que as informações, quando utilizadas de maneira eficiente, se tornam um diferencial para as empresas que as possuem. Por isto as redes sociais desejam obter tantas informações de seus usuários para mais tarde utilizá-las da forma mais inteligente e lucrativa a elas.

# OBJETIVOS

Abaixo serão descritos os objetivos definidos, a fim de obter o resultado proposto por este trabalho.

## Objetivo geral

Desenvolver uma aplicação *web* utilizando uma base de dados orientada a grafos, capaz de realizar buscas por determinadas mãos de obra.

## Objetivos específicos

* Demostrar o uso do banco de dados Neo4j;
* Apresentar o uso do *framework[[2]](#footnote-2)* Cypher cujo o banco de dados Neo4j utiliza para manipular o grafo;

# JUSTIFICATIVA

Este trabalho será desenvolvido utilizando um banco de dados NoSQL, especificamente um banco de dados orientado a grafos. Esta tecnologia não é abordada pela instituição e, como possui um grande potencial de crescimento e popularização além de suas vantagens sobre as demais tais como: a simplicidade para modelá-lo, a maneira como é possível modelar o mundo real utilizando o, além da facilidade de adaptação as mudanças que, nos dias atuais são extremamente comuns, entre outras. Inclusive, grandes empresas como facebook, twitter, netflix, entre outras já utilizam tal tecnologia. Por esta série de motivos decidimos escrever este trabalho a fim de abordar e aprofundar nossos conhecimentos nesta tecnologia.

Foi constatado através de uma pesquisa informal realizada na região de Pouso Alegre que, não há um sistema para a localização de determinados tipos de mãos-de-obra nos quais não existem vínculos empregatícios, tais como: babá, pedreiros, encanadores, faxineira, cuidador(a) de idosos e etc. Devido a esta limitação na região e para acabar com ela, será desenvolvido um sistema cujo principal objetivo é: localizar e apontar os prestadores de serviços com as melhores indicações, levando em consideração vários fatores como: prestadores de serviços que moram próximos ao usuário que está realizando a busca, prestadores de tais serviços que já trabalharam para pessoas que possuem um laço de parceria com aquela que está realizando a busca, entre outras. Isto fará com que o usuário que está realizando a busca obtenha como resultado as melhores opções possíveis a ele.

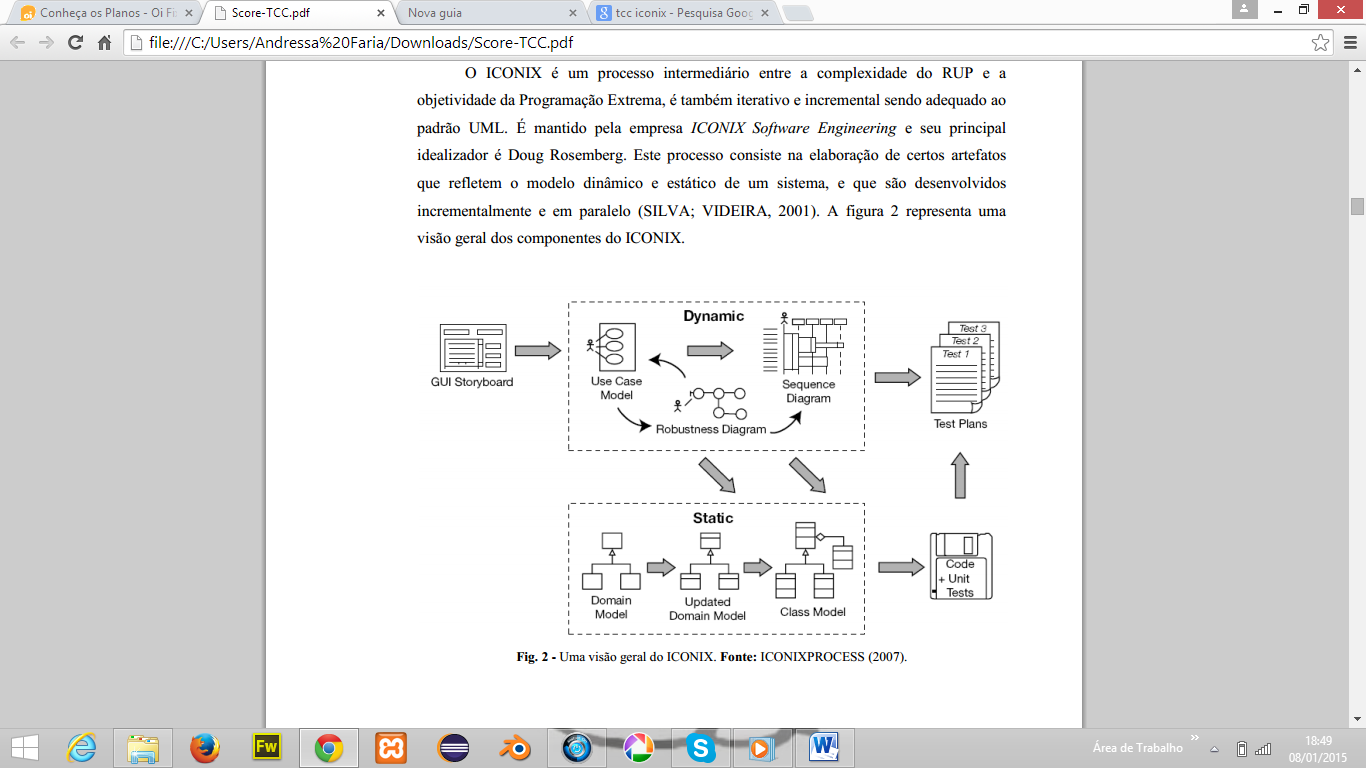
# QUADRO TEÓRICO

Neste capítulo serão discutidas as técnicas, metodologias e tecnologias que serão utilizadas no desenvolvimento deste trabalho.

## Iconix

Existem diversas metodologias de desenvolvimento de *software* que tem como função auxiliar o desenvolvedor durante o projeto. Sua principal função é guiar o desenvolvedor, através de passos que descrevem as melhores práticas para a conclusão do desenvolvimento de um *software*.

Para auxiliar no processo de desenvolvimento deste trabalho usaremos a metodologia ICONIX, que pode ser considerado um processo intermediário entre a complexidade do RUP (*Rational Unified Process*) e a objetividade do XP (*Extreme* *Programming*). O ICONIX segundo Rosenberg (2005) foi criado a partir de um resumo das melhores técnicas de desenvolvimento de software originando assim a UML (*Unified Modeling Language*). Esta metodologia possui como característica ser iterativo e incremental sendo adequado ao padrão UML ele é mantido pela empresa ICONIX *Software Engineering* e seu principal idealizador e Doug Rosenberg. Este processo consiste na elaboração de certos artefatos que refletem o modelo dinâmico e estático de um sistema que são desenvolvidos de maneira incremental e em paralelo (Silva; Videira, 2001). A figura 1 representa uma visão geral dos componentes do ICONIX.



**Fig.1:** Uma visão geral do ICONIX. **Fonte:** ICONIXPROCESS (2007).

O desenvolvimento do projeto é norteado por casos de uso, suas principais fases são: análise de requisitos, análise e projeto preliminar, projeto e implementação. Abaixo temos uma breve descrição de cada fase do ICONIX seguindo ideias de Bona (2002):

* **Análise de requisitos**

Identificar os objetos do problema real e como eles serão abstraídos para um objeto de software através dos modelos de domínio, apresentar um protótipo das possíveis interfaces gráficas, e por fim descrever os casos de uso. Se possível, elaborar também os diagramas de navegação para que cada cliente possa entender melhor o funcionamento do sistema.

* **Análise e Projeto Preliminar**

Detalhamento dos casos de uso com fluxos de ações alternativos e de erros, e a partir disso elaborar a análise de robustez (diagrama de robustez). Ao final, atualiza o diagrama de classes.

* **Projeto**

Elaborar o diagrama de sequência fundamentando-se nos diagramas de casos de uso, e verificar com a equipe se o projeto atende a todos os requisitos.

* **Implementação**

Desenvolvimento do código fonte e testes. O ICONIX não define os passos a serem seguidos nesta fase, ficando a cargo da equipe de desenvolvimento.

Ao término de cada fase um marco é gerado, sendo respectivamente: revisão dos requisitos, revisão do projeto preliminar, revisão detalhada e entrega.

O ICONIX é considerado um processo prático de desenvolvimento de software, pois a partir das iterações que ocorrem na análise de requisitos e na construção dos modelos de domínios (parte dinâmica), os diagramas de classes (parte estática) são incrementados e a partir destes, o sistema poderá ser codificado.

Justamente por essa praticidade disponibilizada pelo ICONIX é que empregamos este processo para o desenvolvimento deste projeto, com o ICONIX podemos ter produtividade no desenvolvimento do *software* ao mesmo tempo em que alguns artefatos são gerados, unindo o aspecto de abrangência e agilidade.

## Tecnologias

Nesta seção serão abordadas as linguagens de programação e as tecnologias que serão utilizadas para o desenvolvimento deste trabalho.

## Java

A *Sun Mycrosystems* lança, em 1995, a tecnologia Java, que em pouco tempo e com a grande ascensão da internet cresceu e se mantém em constante evolução até os dias atuais.

De acordo com a *Oracle* (2010), a tecnologia Java além de ser uma linguagem de programação, é também uma plataforma, que teve como modelo uma outra linguagem, o C++. Ambas possui em comum o conceito de orientação a objeto.

O modelo de desenvolvimento orientado a objeto tem como objetivo trazer o mundo real para dentro do desenvolvimento do *software*, tornando o mesmo mais fácil de entender e reutilizar códigos.

Todo programa que utiliza Java necessita passar por algumas etapas essenciais. Conforme descrito na figura 2, o código é escrito em arquivo de texto com extensão .java, após isto ele será compilado e convertido para um arquivo com extensão *.class*, cujo o texto é transformado em *bytecodes*. Este arquivo com extensão .*class* é interpretado pela JVM[[3]](#footnote-3) que é responsável por executar todo o código do programa.



**Fig.2:** Uma visão geral do processo de desenvolvimento de software. **Fonte:** Oracle (2012).

A tecnologia Java apresenta como grande vantagem a sua portabilidade, o mesmo código que foi escrito e compilado anteriormente para uma determinada plataforma pode ser executado, sem nenhuma dificuldade em outras plataformas. Isso faz com que o desenvolvedor não precise se preocupar com as particularidades de cada plataforma que executará a aplicação.

Por todas as vantagens descritas a cima, além do banco de dados Neo4j utilizar como linguagem de programação o Java, neste projeto será empregado o uso desta tecnologia.

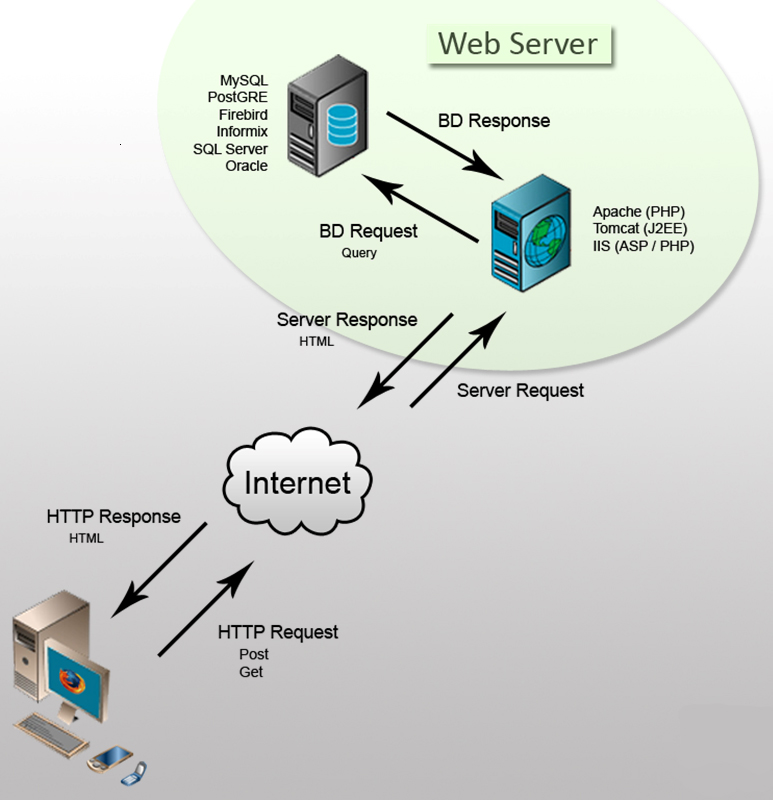
## Tomcat 7

Segundo a Apache (2015) o tomcat é um *software* que possui seu código fonte aberto e é disponibilizado sob a *Apache Lincese version 2*. Isto o tornou um dos *Containers* mais utilizados por desenvolvedores.

Ele foi desenvolvido utilizando a linguagem de programação Java de acordo com Brittain e Darwin (2007, pág.1). Por este motivo é necessário possuir um JRE (*Java SE Runtime Environment)* instalado e atualizado para executá-lo.

O Tomcat é um *container* muito poderoso capaz de executar sem dificuldade alguma aplicações de alto risco e com um alto volume de requisições como é o caso do site da empresa Walmart que é disponibilizado em um servidor *web* executando o Tomcat.

*Containers* são aplicações que são executadas em servidores e possuem a capacidade de hospedar aplicações desenvolvidas em Java *web*. O servidor ao receber uma requisição do cliente entrega esta, não ao *servlet[[4]](#footnote-4)* e sim ao *Container* no qual o *servlet* é distribuído. O *Container* por sua vez entrega ao *servlet* as requisições e respostas HTTP e inicializam os métodos do servlet de acordo com o tipo de requisição realizada pelo cliente. Basham; Sierra; Bates (2010, pág. 39).



**Fig.3:** Uma visão simples da arquitetura web.

Assim como outros *containers* o Tomcatoferece gerenciamento de conexões sockets, suporta *multithreads*, ou seja, ele cria uma nova thread para cada requisição realizada pelo cliente e gerencia o acesso a recursos do servidor, além de outras tarefas.

o Tomcat em especial foi escolhido para ser utilizado neste trabalho, pois o objetivo deste é desenvolver uma aplicação *web* e para hospedá-la em um servidor, uma aplicação container será necessária. Por este motivo e somado a sua facilidade de configuração, além das vantagens acima descritas tal decisão foi tomada.

## Neo4j

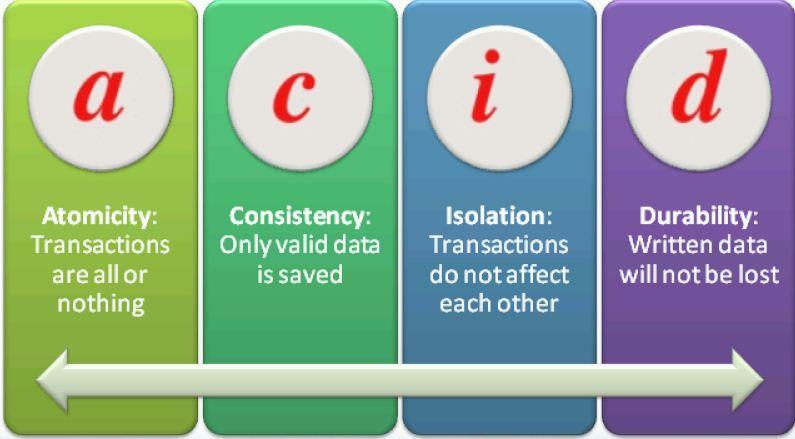
Segundo Bruggen (2014) o Neo4j foi criado no início do século 21 por desenvolvedores que queriam resolver um problema em uma empresa de mídias. Porém os desenvolvedores não obtiveram êxito ao tentar resolver tal problema utilizando as tecnologias tradicionais, portanto, eles decidiram arriscar e criar algo novo. A princípio, o Neo4j não era um sistema de gerenciamento de banco de dados orientado a grafos como é conhecido nos dias atuais. Ele era mais parecido com uma *graph library* (biblioteca de grafo) que as pessoas poderiam usar em seus códigos.

Inicialmente ele foi desenvolvido para ser utilizado em conjunto com alguns bancos de dados relacionais como MySQL e outros, com a intenção de criar uma de abstração dos dados em grafos. Mas com o passar dos anos os desenvolvedores decidiram tirar o Neo4j da estrutura dos bancos relacionais e criar sua própria estrutura de armazenamento em grafos.

Neo4j, como vários outros, também é um projeto de sistema de gerenciamento de banco de dados NoSQL *open source.* De acordo com Rocha (2013, pág. 27) NoSQL é um acrônimo para Not Only SQL, isto significa que estes tipos de banco de dados utilizam não somente os recursos de Structure Query Language (SQL), mas também outros recursos a fim de obter a melhor performance possível.

Neo4j é um banco de dados robusto, escalável e de alta performance. Possuindo as seguintes características segundo Neo4j team (2013, pág. 2):

* *true* *ACID transactions;*
* *High availability;*
* *Scales to billions of nodes and relationships;*
* *High speed querying through traversals*



**Fig.4:** Propriedades ACID.

O banco de dados Neo4j é composto por alguns componentes que serão apresentados abaixo.

* **Nós**

Os nós e os relacionamentos são as unidades fundamentais no grafo. Podendo ambos possuir ou não propriedades. Estas propriedades não possuem nenhum tipo limitação a respeito da quantidade de propriedades por nó ou relacionamento.

* **Relacionamentos**

Os relacionamentos são os responsáveis por conectar os nós entre si e organizar o grafo permitindo a localização de informações relacionadas.

* **Propriedades**

Propriedades são pares de *key-value* onde a chave é uma *string* e o valor pode ser um tipo primitivo de valor ou um *array* de *tipo primitivo.*

* **Paths**

Basicamente é um conjunto de nós ou apenas um com conectados entre si e que geralmente são recebidos como resultados de uma query ou um traversal.

* **Traversal**

É uma API que permite ao usuário definir algumas regras de navegação no grafo. As escolhas mais simples são entre navegar por amplitude ou em profundidade.

De acordo com Neo4j *team* (2013, pág. 6) a aplicação que utilizar o Neo4j poderá extrair todo o potencial do grafo além de, manter toda a confiança e segurança que um banco de dados lhe oferece.

Por possuir seu código fonte aberto e ser um banco de dados orientado a grafo bastante robusto, seguro e possuir uma documentação de fácil entendimento, além é claro de possuir um baixo custo de implantação devido a sua licença *open source* este banco de dados foi escolhido para ser utilizado neste trabalho.

## Cypher Query Language

O Cypher Query Language é uma *graph database query language* específica para o banco Neo4j. Ele foi criado devido a necessidade de manipular os dados e realizar buscas em grafos de uma forma mais simples.

Robinson; Webber; Eifrem (2013, pág. 27) afirmam que:

*Cypher is designed to be easily read and understood by developers, database professionals, and business stakeholders. Its ease of use derives from the fact it accords with the way we intuitively describe graphs using diagrams*.[[5]](#footnote-5)

Com esta afirmação é possível compreender o quanto este framework pode auxiliar a todos os profissionais envolvidos no projeto. Isto ocorre devido ao nosso hábito de representar grafos usando diagramas, desta forma, utilizando este framework é possível descrever os grafos programaticamente de uma forma mais precisa.

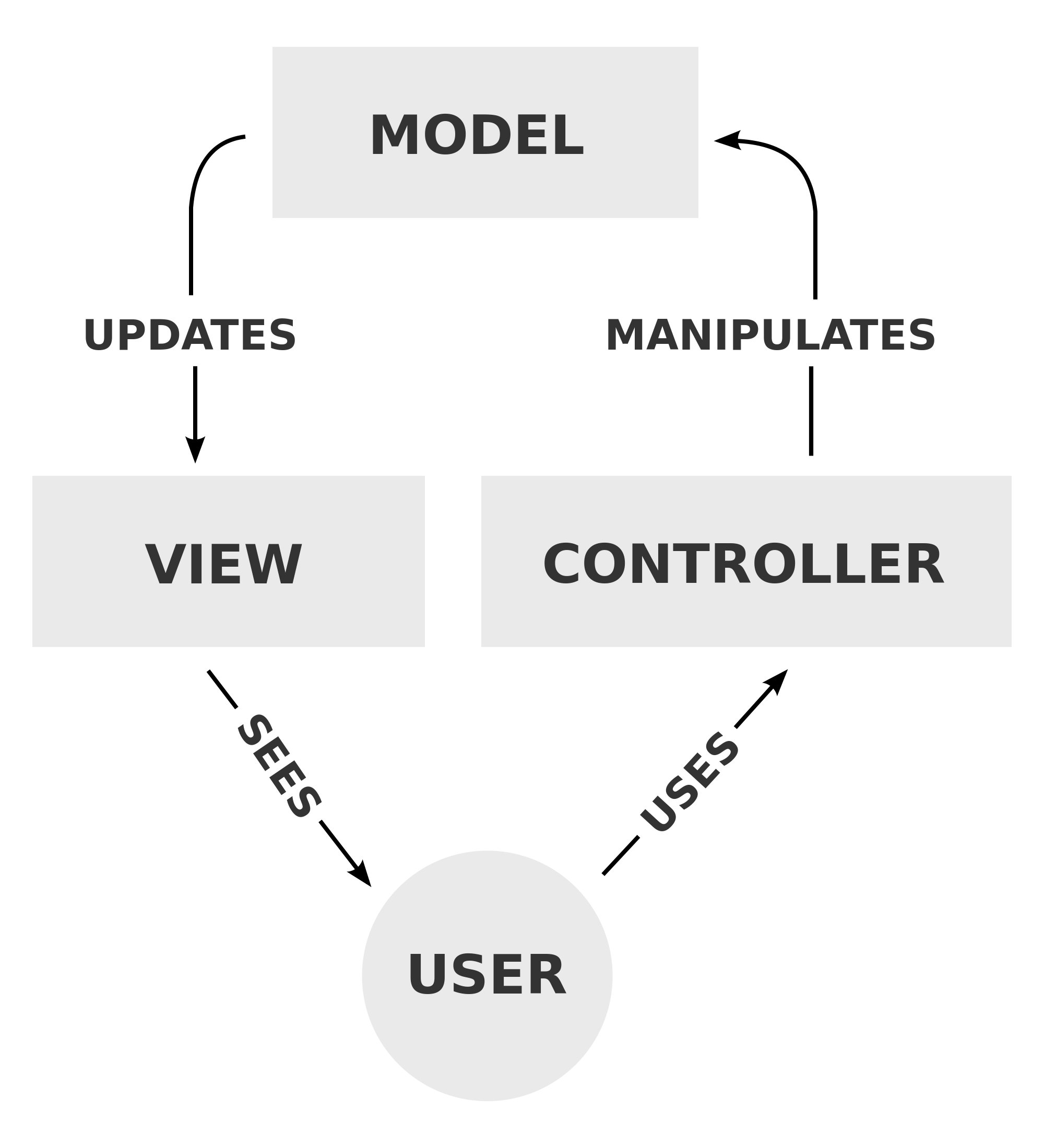
Outras *Query Languages* existem, inclusive com suporte ao Neo4j (banco de dados utilizado para o desenvolvimento deste trabalho), porém devido as vantagens apresentadas acima somada ao fato que o *Cypher Query Language* possui uma curva de aprendizado menor e é excelente para lhe oferecer uma base a respeito de grafos, neste projeto será utilizado o framework *Cypher Query Language* para realizar as tarefas necessárias no banco de dados.

## JavaServer Faces

O JSF (JavaServer Faces) é um *framework server-side* baseado em componentes *web*, cujo sua principal função é abstrair os detalhes de manipulação dos eventos e organização dos componentes na página *web*. Através dele é possível desenvolver páginas *web* mais sofisticadas de forma simples. Abstraindo inclusive, o tratamento de requisições e respostas. Bergsten (2004, pág. 1). Isto permite ao desenvolvedor focar-se no *back-end* da aplicação, ou seja, na lógica, e não se preocupar com detalhes a respeito de requisições e respostas HTTP e como obter as informações recebidas e/ou enviadas através deste protocolo.

De acordo com Oracle (2014) o JSF é de fácil aprendizado e utilização pois, possui sua arquitetura claramente definida, sendo divida entre a lógica da aplicação e apresentação. Esta divisão é possível pois ele utiliza o padrão de projeto MVC (*Model-View-Controller*), tornando o um importante framework para desenvolvimento de aplicações utilizando a plataforma Java *Web* e com alta demanda de procura no mercado.

Segundo Gamma; Helm; Johnson; Vlissides (2009) o padrão de projeto MVC é dividido em três partes. O *Model* é a lógica de sua aplicação, a *View* é a apresentação de sua aplicação e por último o *Controller* que é responsável por definir a interface entre a lógica com a apresentação da aplicação. Portanto, todo tipo de requisição ou resposta deve ser obrigatoriamente enviada ao *Controller*  que por sua vez encaminhará para a camada de visão ou de lógica da aplicação.



**Fig.5:** Estrutura do padrão MVC.

Por possuir as vantagens descritas acima e possuir uma simples configuração, além de ser um framework *cross-browser* (cujo a aparência e funcionalidade dos componentes serão as mesmas independente de dispositivo ou navegador de internet utilizado). Somado ao fato deste projeto utilizar como banco de dados o Neo4j que utiliza como linguagem de programação o Java este framework foi escolhido para auxiliar no desenvolvimento das páginas *web*.

## Primefaces

O Primefaces é uma biblioteca de componentes de código fonte aberto que implementa a especificação do JSF e deve ser utilizada em conjunto com o mesmo a partir da versão 2.0 possuindo diversos componentes para auxiliar o desenvolvimento de interfaces ricas segundo Ross; Borsoi (2012).

Segundo a Juneau (2014), uma das grandes vantagens do Primefaces é a facilidade de integração entre ele e o JSF, bastando apenas, incluir a biblioteca do Primefaces no projeto JSF. Salvo alguns componentes específicos como o caso do *file upload* que necessitam de pequenas configurações adicionais.

Outra vantagem que o Primefaces nos traz segundo Dobnik; Šalej é o suporte a criação de templates para páginas *web* de acordo com a necessidade do desenvolvedor, além da grande quantidade inclusos por padrão.

Por todas as vantagens mencionadas acima, somado ao fato desta biblioteca possuir uma ótima documentação, em conjunto com uma grande comunidade de desenvolvedores que a utilizam e a alta demanda de desenvolvedores que conheçam esta biblioteca no Mercado, esta biblioteca foi escolhida em conjunto com o JSF para desenvolver as páginas *web* deste projeto.

# QUADRO METODOLÓGICO

Será apresentado neste capitulo a metodologia de pesquisa que sera utilizada para realizar este projeto e os passos necessários até a sua conclusão.

## Tipo de pesquisa

Para realizar este projeto, será necessário o uso de uma metodologia de pesquisa para analisar possíveis melhorias no processo de busca por determinadas mãos de obra temporária, para a realização de trabalhos domésticos, cujo um sistema de plataforma web possa oferecer.

Para Padua(2007, p. 31) pesquisa é:

“Toda atividade voltada para a solução de problemas; como atividade de

busca, indagação, investigação, inquirição da realidade, e a atividade que

visa nos permitir, no âmbito da ciência, elaborar um conhecimento, ou um conjunto de conhecimentos, que nos auxilie na compreensão desta

realidade e nos oriente em nossas ações.”.

O método de pesquisa aplicada foi escolhido para a obtenção dos dados, pois conforme Cooper e Schindler (2003, p.32) ela “ tem uma ênfase prática na solução de problemas, embora a solução de problemas nem sempre seja gerada por uma circunstância negativa.”

Esta metodologia será utilizada no desenvolvimento deste projeto, pois a mesma busca analisar o problema e gerar uma solução para o mesmo através de um aplicativo ou serviço. Neste caso, será o desenvolvimento de um sistema web que auxilia na busca por profissionais temporários.

## Contexto

Encontrar um profissional com boas referências que ofereça serviços temporários de qualidade não é uma tarefa muito fácil hoje em dia, este tipo de mão de obra está se tornando cada vez mais escasso. Por este motivo, muitas pessoas ficam desgastadas com a busca.

Esta pesquisa terá como foco todas as pessoas que necessitam de mão de obra temporária para realizar tarefas domésticas e rotineiras, além daquelas que não ocorrem com tanta intensidade. Uma vez que o sistema será desenvolvido em uma plataforma web, todas as pessoas terão fácil acesso ao serviço, sendo necessário apenas possuir uma comunicação com a internet.

## Participantes

Participam deste projeto dois acadêmicos e um professor orientador. Sendo eles:

* Andressa de Faria Giordano: acadêmica do 7º período do curso de bacharelado em Sistemas de Informação pela Universidade do Vale do Sapucaí (Univás). Seu papel neste projeto será o levantamento dos requisitos e a elaboração de todos os diagramas descritos nas fases do ICONIX, o desenvolvimento do banco de dados, a implantação da logica de negócios, comunicação entre os módulos, acesso aos dados pelo sistema e a escrita da documentação.
* Edilson Justiniano: acadêmico do 7º período do curso de bacharelado em Sistemas de Informação pela Universidade do Vale do Sapucaí (Univás). Seu papel neste projeto será o levantamento dos requisitos e a elaboração de todos os diagramas descritos nas fases do ICONIX, o desenvolvimento do banco de dados, a implantação da logica de negócios, comunicação entre os módulos, acesso aos dados pelo sistema e a escrita da documentação.
* Márcio Emílio Cruz Vono de Azevedo: professor orientador, engenheiro em telecomunicações pelo Instituto Nacional de Telecomunicações (INATEL) e mestre em Ciência da Computação pela Universidade Federal de Itajubá (UNIFEI). Professor do INATEL e da Universidade do Vale do Sapucaí (Univás) na área da informação e especialista em sistemas do Inatel.

## Instrumentos

Os instrumentos de pesquisa são as ferramentas usadas para a coleta de dados. Como afirma Marconi e Lakatos (2009, p.117) os instrumentos de pesquisa abrangem: “desde os tópicos de entrevista, passando pelo questionamento e formulário, até os testes ou escala de medida de opiniões e atitudes”.

Para a realização deste projeto serão utilizados formulários de pesquisa, reuniões e análise documental como instrumentos de pesquisa.

As próximas subseções descreverão como serão utilizados os instrumentos de pesquisa neste projeto.

## Formulários de pesquisa

A observação será feita através de formulários de pesquisa que avaliarão o interesse dos usuários pelo sistema que ofereça o serviço de localização de mão de obra temporária. Os testes serão feitos visando oferecer a melhor interação entre o usuário e o software, levando em conta sua aplicabilidade, desempenho e facilidade de utilização.

Os pontos mais relevantes serão: facilidade de utilização, aprendizagem, aplicabilidade e eficiência do software.

## Reuniões

Serão realizados encontros presenciais e também virtuais com os acadêmicos e com o professor orientador, para o levantamento dos requisitos, da aplicabilidade do software, dos modelos de engenharia de software e codificação do sistema, como também questões teóricas que estejam relacionadas com as melhores práticas para se obter um software aplicável e ágil.

## Análise documental

Para auxiliar no desenvolvimento deste projeto serão utilizados documentos como: manuais, tutoriais, livros, trabalhos e artigos acadêmicos, além de pesquisa web. Estes serão utilizados para o embasamento teórico e também para o prático, que será dividido em duas etapas.

## Procedimentos

Para desenvolver este projeto, usaremos os processos de engenharia de softwares definidos no ICONIX, que consistem em quatro etapas subdivididas em algumas tarefas específicas.

Na primeira etapa, será realizada a análise dos requisitos que serão necessários para o início do projeto. Esses requisitos serão esboçados em telas, onde os autores do projeto farão a abstração dos objetos do mundo real e o software. A partir deste contexto, será desenvolvido o modelo de domínio e, posteriormente, os casos de uso.

Após o término da primeira etapa, será feita a análise preliminar, realizando para cada caso de uso, um diagrama de robustez e atualizando paralelamente o modelo de domínio, com os atributos e métodos. Ao término desta fase será verificado a consistência dos artefatos gerados e, se houver necessidade, será refeita a etapa de análise de requisitos, incluindo as alterações descobertas. Com as informações geradas a partir deste processo, será possível desenvolver a base de dados da aplicação.

Após a segunda etapa concluída será criado, para cada caso de uso, um diagrama de sequencia, contendo detalhes a respeito da futura implementação e novamente, o modelo de domínio gerado na 1ª etapa será atualizado, incluindo a ele os novos métodos que foram coletados neste processo.

Para concluir, o último passo será a implementação, que com base em todos os processos descritos acima, será realizado com segurança, de que este projeto possivelmente irá atender aos requisitos levantados.

## Cronograma

Será exibido aqui um quadro contendo o cronograma a ser seguido por este projeto, desde a sua definição até a sua conclusão.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mês**  **Ação** | Dez | Jan | Fev | Mar | Abri | Mai | Jun | Jul | Ago | Set | Out | Nov | Dez |
| Definição do Pré-projeto |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Aprovação de Pré-projeto |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Levantamento bibliográfico |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Primeira entrega do Pré-projeto |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Orientação sobre Introdução |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entrega da Introdução |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Orientações sobre Objetivos e Justificativas |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entrega dos Objetivos e Justificativas |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Orientações sobre o Quadro Teórico |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entrega do Quadro Teórico |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Orientações sobre o Quadro Metodológico |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entrega do Quadro Metodológico |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Revisão de Referências |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Qualificação do projeto |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Levantamento de requisitos |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desenvolvimento |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Reuniões |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pré- banca |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Banca |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Orçamentos

Abaixo serão apresentadas as despesas de forma geral previstas para a realização deste projeto.

|  |  |
| --- | --- |
| **Despesas** | **Valor Previsto (R$)** |
| Impressão | R$ 70,00 |
| Encadernação | R$ 35,00 |
| Impressão em capa dura | R$ 80,00 |
| Livros | R$ 274,00 |
| Total | R$ 459,00 |

# REFERÊNCIAS

APACHE. **Apache Tomcat**. Disponível em: <http://tomcat.apache.org/index.html> Acesso em: 15 jan. 2015.

BARBOSA K. **Por que as pessoas usam as redes sociais?**. Disponível em: <http://super.abril.com.br/blogs/tendencias/por-que-as-pessoas-usam-as-redes-sociais/> Acesso em 28 dez. 2014.

BARRETO A. **A eficiência técnica e econômica e a viabilidade de produtos e serviços de informação**, ed. [S.l : s.n], v. 25, n. 3, 1996.

BASHAM B.; SIERRA K.; BATES B. **Use a Cabeça! Servlets & JSP**, S.l., Alta Books, n. 2, 2010.

BERGSTEN H. **JavaServer Faces**, ed. [S.l : s.n], abr. 2004.

BONA C. **Avaliação de Processos de Software: um estudo de caso em XP e ICONIX.**

Florianópolis: Universidade Federal de Santa Catarina, 2002.

BRITTAIN J.; DARWIN I. **Tomcat The Definitive Guide**, ed. [S.l], [s.n], n. 2, 2007

BRUGGEN R., **Learning Neo4j**, [S.l], Packt, 2014.

COOPER D.; SCHINDLER P. **Métodos de Pesquisa em Administração**, Bookman, Porto Alegre, n. 7, 2003.

DEITEL H.; DEITEL P. **Java, Como Programar**, Bookman, Porto Alegre, n. 4, 2003.

DOBNIK S.; ŠALEJ M. **Developing Rich Web Applications with Primefaces in Java EE7**. Disponível em: <https://docs.google.com/file/d/0B7kPrfCRft\_qa1ZOdU5kUUpuZ2c> Acesso em: 15 jan. 2015.

FURBINO Z. **Mais de 80% dos brasileiros com smartphone usam redes sociais**. Disponível em: <http://www.em.com.br/app/noticia/tecnologia/2014/09/30/interna\_tecnologia,574196/mais- de-80-dos-brasileiros-com-smartphone-usam-redes-sociais.shtml> Acesso em: 28 dez. 2014.

GAMMA E.; HELM R.; JOHNSON R.; VLISSIDES J. **Design Patterns: Elements of Reusable Object-Oriented Software**, [S.l : s.n], 2009.

GOES G. **Os 20 sites mais acessados no mundo em 2013**. Disponível em: <http://www.techtudo.com.br/artigos/noticia/2013/02/os-20-sites-mais-acessados-no-mundo- em-2013.html> Acesso em 28 dez. 2014.

GONÇALVES M.; GOUVEIA S.; PETINARI V., **A Informação como produto de alto valor no mundo dos negócios**, São Paulo: [s.n], v. 1, n. 1, p. 43-54, 2008.

HEUSER C. **Projeto de Banco de Dados**, [S.l : s.n], n. 4, 1998.

JUNEAU J. **Primefaces in the Enterprise**. Disponível em: <http://www.oracle.com/technetwork/articles/java/java-primefaces-2191907.html> Acesso em: 15 jan. 2014.

ORACLE, **About the Java Technology**. Disponível em: < http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html> Acesso em: 28 dez. 2010.

\_\_\_\_\_\_. **JavaServer Faces Technology Overview**. Disponível em: <http://www.oracle.com/technetwork/java/javaee/overview-140548.html> Acesso em: 15 jan. 2014.

PADUA E. **Metodologia da Pesquisa: Abordagem Teorico-Pratica**. ed. Campinas : Papirus Editora, 2007.

ROBINSON I.; WEBBER J.; EIFREM E. **Graph Database**, ed. [S.l], O’REILLY, 2013.

ROCHA R. **Algoritmos de Particionamento e Banco de Dados Orientado a Grafos**, ed.

Itajubá, [s.n], 2013.

ROSENBERG D.; STEPHENS M.; COLLINS-COPE M. **Agile Development with ICONIX Process: People, Process, and Pragmatism**. New York, Estados Unidos : April 2005.

ROSS C.; BORSOI B. **Uso de Primefaces no desenvolvimento de aplicações ricas para web**, ed. Paraná, [s.n], 2012.

SANCHES R. **Fundamentos de Armazenamento e Manipulação de Dados**, ed. [S.l : s.n], 2002.

**Área:** Banco de dados

**Tema:** NoSQL, Banco de dados orientado à grafos

**Objetivos específicos (Na verdade são procedimentos):**

* Aprender sobre banco de dados orientado à grafos;
* Aprender a utilizar o Neo4j;
* Modelar o banco de dados;
* Modelar o sistema;
* Implementar o sistema;

Houve um tempo onde havia mão de obra para serviços temporários em abundância, principalmente serviços, cujo as principais atividades eram realizadas de forma braçal. Serviços como: pedreiros, marceneiros, diaristas etc, porém esta realidade está mudando e o que antes estava escasso possui em abundância e vice-versa. E devido a esta escassez de mãos de obra, encontrar uma pessoa qualificada se tornou uma tarefa difícil de ser realizada. Nem sempre os profissionais que possuem disponibilidade são os mais competentes o que afeta diretamente no resultado de seu trabalho em alguns casos.

Este cenário está diretamente relacionado ao envelhecimento da população e a cultura das novas gerações. Mãos de obra como cuidador(a) de idosos, babá, pedreiros, marceneiros entre vários outros que a algum tempo atrás haviam em abundância pois, a população mais idosa em geral não possui amplo acesso a educação e isto as forçavam a aprender as profissões com seus familiares e exercê-las, produzindo mão de obra em abundância. Diferente do que ocorre hoje, cujo a atual geração possui maior acesso a educação, tornando possível a um jovem cursar boas instituições de ensino, se formar e obter sucesso em sua carreira profissional, deixando o cada vez mais distante destes tipos de mãos de obra. Com esta inversão de cenário, o que anteriormente era simples de se encontrar se tornou complicado.

Hoje em dia a população não dispõe de tempo hábil para realizar buscas por pessoas confiáveis e qualificadas devido ao ritmo de vida que elas levam. E como não há um local ou *software* que concentre estas informações a respeito destes prestadores de serviço, esta busca poderá se tornar uma tarefa desgastante para a população., impedindo aos contratantes em alguns casos .

A fim de facilitar a localização destas mãos de obras qualificadas que hoje estão escassas este projeto será desenvolvido para gerar um *software* capaz de realizar tal busca e encontrar os melhores profissionais ao usuário. Evitando uma série de problemas encontrados para localizar estas pessoas.

Para encontrar as melhores opções de prestadores de serviços, este projeto utilizará um conceito mais atual de banco de dados denominado banco de dados *NoSQL*. Segundo Rocha (2013, pág. 27), estes tipos de bancos de dados são denominados desta forma, pois, utilizam não somente os recursos do *SQL (Structure Query Language)*, mas também outros recursos a fim de obter melhor desempenho.

**Comentar que este trabalho usará o banco Neo4j**

**Comentar que Será usado os bancos em grafos, pois será utilizado uma rede social**

Começamos a falar sobre a dificuldade de se encontrar uma mão de obra qualificada devido grande parte ao envelhecimento da população brasileira e a cultura da população que não está disposto a prestar serviços braçais em alguns casos.

**Podemos citar depois que os bancos de dados foram criados na década tal. E com a evolução deles, seguido do desejo de trazer a computação ao cotidiano do mundo real outra geração de banco de dados foi criada. Sendo elas os bancos NoSQL.**

**Então desta forma este projeto será realizado para integralizar esta tecnologia (Graph Databases) à um sistema que facilite a localização de pessoas.**

1. NOSQL – São bancos de dados que utilizam não somente os recursos de Structure Query Language (SQL), a fim de obter a melhor performance. [↑](#footnote-ref-1)
2. FRAMEWORK – É uma abstração que une códigos comuns entre vários projetos de software, a fim de obter uma funcionalidade genérica. [↑](#footnote-ref-2)
3. JVM – *Java Virtual Machine* é a responsável pela execução dos *Bytecodes* (programa java compilado). [↑](#footnote-ref-3)
4. Servlet – Tecnologia capaz de gerar páginas dinâmicas na plataforma Java. [↑](#footnote-ref-4)
5. *Cypher* é desenvolvido para ser facilmente lido e entendido por desenvolvedores, profissionais de banco de dados e as partes interessadas. Sua facilidade de uso deriva do fato de estar de acordo com a nossa forma intuitive de descrever os grafos usando diagramas. [↑](#footnote-ref-5)