Fundação Educacional do Município de Assis
Instituto Municípal de Ensino Superior de Assis - IMESA

Joel Rodrigues Alvares Leal

Desenvolvimento de Aplicações em Nuvem usando conceitos de Tecnologia Adaptativa

2014

Assis - SP

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

Desenvolvimento de Aplicações em Nuvem usando conceitos de Tecnologia Adaptativa

Trabalho de Conclusão do Programa de Iniciação Científica (PIC) do Instituto Municipal de Ensino Superior de Assis - Imesa.

Aluno: Joel Rodrigues Alvares Leal
Orientador: Dr. Almir Rogério Camolesi
Linha de Pesquisa: Informática

2014
Assis - SP

FICHA CATALOGRÁFICA

LEAL, Joel Rodrigues Alvares

Desenvolvimento de Aplicações em Nuvem usando conceitos de Tecnologia Adaptativa/Joel Rodrigues Alvares Leal. Fundação Educacional do Município de Assis – Fema: Assis, 2014

43p.

Orientador: Dr. Almir Rogério Camolesi
Projeto de Iniciação Científica (PIC) – Ciência da Computação – Instituto Municipal de Ensino Superior de Assis

1. Java 2. Nuvem 3. Tecnologia Adaptativa

CDD: 001.6
Biblioteca da FEMA

RESUMO

Neste projeto foram desenvolvidas várias aplicações em nuvem, sendo aplicado nelas os conceitos de tecnologia adaptativa.

Dentro deste contexto, o desenvolvimento dessas aplicações visam criar uma possibilidade de aplicações que se adaptam em tempo de execução conforme seu dados de entrada são informados, sem que ninguém precise informar o que ela precisa fazer.

A implementação destas aplicações foi feita utilizando a tecnologia Java com o intuito de adquirir conhecimento sobre esta área, tendo em vista que o mercado profissional é promissor.

Palavras chave: Java. Nuvem. Tecnologia Adaptativa.

"Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível."

Charles Chaplin

Agradecimentos

Agradeço primeiramente e Deus por ter me dado muita força e por sempre estar presente em minha vida, caminhando ao meu lado todos os dias, me fazendo mais forte e motivado.

A FEMA pela bolsa de estudo que me concedeu durante o decorrer desse ano letivo de 2014.

Ao professor Almir Rogério Camolesi, além de um excelente orientador, um amigo que me apoiou e me ajudou no decorrer do projeto transmitindo todo o conhecimento possível. Saiba que com você aprendi muito, existem muitas pessoas que podem fazer o mesmo, porém, poucas com grande dedicação e excelência.

A minha namorada Juliana Barroso Lomiler pelo apoio, ajuda, compreensão, incentivo ao decorrer do projeto.

A minha família Paulo Cesar, Nivea e João Pedro pelo apoio e incentivo. Obrigado pelas orações.

LISTA DE FIGURAS

SUMÁRIO

1 - INTRODUÇÃO

Nos últimos anos, com o crescimento na área da Tecnologia da Informação, a internet vem sendo um meio bastante utilizado para enviar e ter acesso as informações, tanto para um usuário simples como empresas de grande porte, o que fez desenvolvedores de sistemas e fornecedores de equipamentos investirem em soluções para simplificar o acesso a estas informações. Entre todas estas tecnologias, uma que vem ganhando destaque é a Computação em nuvem (*Cloud Computing*)-.

Acredita-se que futuramente aplicativos e dados não serão armazenados em nossos computadores pessoais. Caberá a empresas fornecerem estes serviços através da internet, seguindo a ideia de Arquitetura Orientada a Serviço (*Service Oriented Architecture – SOA).* Ou seja, por meio de um site o usuário cria uma conta, utiliza o aplicativo em rede e pode salvar todo o trabalho que for feito e acessá-lo em qualquer outro lugar desde que haja uma conexão com a internet.

É comum encontrar no meio comercial, sistemas que se modificam, em comportamento e estrutura, para solucionar um problema ou em situações inesperadas. Independente de seus estímulos de entrada, seu comportamento deve ser modificado para atender as novas situações apresentadas. Neste contexto é que situa-se este trabalho que tem como foco o estudo e o uso de conceitos de Tecnologia Adaptativa no desenvolvimento de aplicações em nuvem.

As aplicações desenvolvidas em nuvem, geralmente, são complexas e possuem comportamento que pode ser modificável. Uma tecnologia que vem sendo empregado no desenvolvimento de aplicações com comportamento modificável é a Tecnologia Adaptativa (NETO, 1993). A tecnologia adaptativa envolve um dispositivo não-adaptativo (subjacente) já existente em uma camada adaptativa que permite realizar mudanças no comportamento da aplicação definida (Pistori, 2003). A Tecnologia Adaptativa se caracteriza por ser um sistema com estrutura dinâmica, ou seja, sua estrutura pode ser alterada conforme ocorre a interação com o ambiente, interno (virtual) ou externo (real), e por esta característica tão interessante, acaba por se tornar algo que facilita a construção de aplicações em nuvem.

1.1 - Objetivos

Este projeto tem como finalidade à aplicação dos conceitos de tecnologia adaptativa no desenvolvimento de aplicações em nuvem. Para o desenvolvimento de tais aplicações foi necessário conhecer as tecnologias que compõem o modelo da arquitetura do problema a ser abordado, com o intuito que as aplicações sejam rápidas e eficientes para ter o menor gasto possível. O objetivo é encontrar onde se faz necessário o uso dos conceitos de tecnologia adaptativa, e logo após, verificar se também é necessário que tais aplicações fiquem em nuvem. A linguagem de programação utilizada foi Java[1], o banco de dados utilizado foi PostgreSQL[2] e as ferramentas utilizadas para o desenvolvimento em nuvem foi as disponibilizadas pela Amazon Web Service (AWS)[3].

1.2 - Justificativas

A partir das pesquisas realizadas foi possível perceber o quão escassos é o desenvolvimento de softwares utilizando os conceitos de tecnologia adaptativa, o quão complexo são as aplicações em nuvem e como cresce a cada dia essas duas tecnologias.

Sendo assim, a justificativa é de que o desenvolvimento desse projeto tenha contribuição para o desenvolvimento de aplicações em nuvem e à aplicabilidade dos conceitos de tecnologia adaptativa em softwares desenvolvidos para que possamos ter melhor desempenho.

1.3 - Motivação

A motivação para o desenvolvimento desse projeto é fundamentada em problemas que ainda não foram solucionados tanto com computação em nuvem, quanto na utilização dos conceitos de tecnologia adaptativa.

Sendo assim, verificamos onde podem ser aplicados esses conceitos em aplicações comerciais, onde eles se encaixam. Para provar que isso se faz necessário, foi aplicado esses conceitos em aplicação desenvolvidas para nuvem e apresentados quais são os pros e os contras no uso destas tecnologias.

1.4 – Estrutura do Trabalho

A estrutura do trabalho foi dividida em capítulos:

- Capítulo 1: Introdução
- Capítulo 2: Fundamentação Teórica
- Capítulo 3: Desenvolvimento do Projeto
- Capítulo 4: Conclusão

Referências Bibliográficas

2 - FUNDAMENTAÇÕES TEÓRICAS

Neste capítulo serão apresentados os conceitos que fundamentam esse projeto de pesquisa. Primeiramente, serão feitas considerações sobre as tecnologias utilizadas para o desenvolvimento do projeto. A teoria utilizada para o desenvolvimento do projeto foi os conceitos de computação em nuvem, tecnologia adaptativa e a tecnologia Java.

2.1. Computação em Nuvem

Computação em nuvem é uma tendência recente de tecnologia cujo objetivo é proporcionar serviços de Tecnologia da Informação (TI) sob demanda com pagamento baseado no uso. Tendências anteriores à computação em nuvem foram limitadas a uma determinada classe de usuários ou focadas em tornar disponível uma demanda específica de recursos de TI, principalmente de informática. O que torna hoje a computação em nuvem uma tecnologia muito interessante é o avanço tecnológico na velocidade das conexões disponíveis e o ilimitado armazenamento de dados.

MILLER (2008) destaca que, por se tratar de um novo paradigma, existem muitas contradições. Entretanto, a maioria dos pesquisadores considera que essa nova abordagem deva proporcionar economia de escala, uma vez que possibilitará que usuários domésticos, a partir de um computador com capacidades reduzidas ou até mesmo um televisor de alta definição, possa utilizar serviços especializados, oferecidos por companhias.

Basicamente há três partes que compõe os elementos de uma computação em nuvem: Clientes, *Data Center* e Servidores Distribuídos. Cada elemento desenvolve um papel especifico na solução.

Data Center é um conjunto de servidores onde são armazenados os aplicativos que são acessados pela internet. Uma técnica que vem crescendo na área de TI é a virtualização de servidores. O que permite que vários servidores sejam instalados em apenas uma maquina física, vale lembrar que o número de servidores instalados em uma única máquina dependerá do tamanho e da

velocidade do servidor e quais aplicações estão funcionando no servidor virtual. Nesta técnica o software pode ser instalado permitindo que vários servidores virtuais sejam utilizados. VELTE (2011, p.07).

Não há a necessidade dos servidores estarem alocados juntos. Normalmente estes servidores estão espalhados pelo mundo, mas para o usuário, estes servidores trabalham como se estivessem no mesmo local. Isto proporciona ao fornecedor mais flexibilidade e opções de segurança. Se acontecer a perda de comunicação de algum servidor, o serviço ainda poderá ser acessado de outro servidor disponível. VELTE (2011, p.08).

Há muito mais coisas acontecendo por trás das nuvens do que simplesmente igualá-la a internet e armazenamento de informações, atualmente a computação em nuvem é classificada em três modelos de serviços. Software como Serviço (SaaS), Plataforma como um serviço (PaaS) e Infraestrutura como um serviço (IaaS).

O Software como Serviço (SaaS) nada mais é do que um aplicativo oferecido como um serviço aos clientes que acessam através da internet. Este tipo de software custa menos que comprar todo o aplicativo, resumindo, quanto menos ele for usado, mais ele será em conta.

Outra característica do Software como Serviço (SaaS) é que o cliente fica livre da obrigação de manter sua infraestrutura atualizada e funcionando, além de fornecerem uma proteção mais eficiente.

A Plataforma como um Serviço (PaaS) bem como os SaaS é outro modelo de aplicação, na Plataforma como um Serviço (PaaS) é fornecido todos os recursos e ferramentas necessárias para que desenvolvedores de software construa novos aplicativos e serviços diretamente da internet sem precisar instalá-las em seu computador pessoal.

O fator de definição que torna PaaS exclusiva é que permite que desenvolvedores desenvolvam e implementem aplicativos da Web em uma infraestrutura hospedada. Ou seja, PaaS permite aproveitar os recursos de computação aparentemente infinitos de uma infraestrutura de nuvem (ORLANDO, 2011).

Enquanto os SaaS e PaaS oferecem aplicações aos clientes, a Infraestrutura como um Serviço (IaaS) fornece apenas recursos de Hardware, geralmente na forma de virtualização, podem ser fornecidos recursos como: Espaço Físico para servidor; Memória; Equipamentos de rede; Ciclos de CPU; Espaço de Armazenamento.

Na forma de virtualização, vários usuários podem utilizar os recursos de uma única máquina simultaneamente. O pagamento para a utilização destes recursos é de acordo com a quantidade utilizada, ou seja, ela é ajustada de acordo com a necessidade do cliente.

2.2. Tecnologia Adaptativa

Um dos primeiros trabalhos relacionado ao estudo da Tecnologia Adaptativa é o apresentado por NETO e MAGALHÃES (1981), que fornece uma visão de métodos de análise sintática e de geração de reconhecedores sintáticos. Este trabalho foi resultado de um primeiro esforço em busca da inclusão dos conceitos de mecanismos adaptativos em um sistema para apoio à construção de compiladores.

Neto (1983) elaborou uma extensão do modelo inicial que exibe a capacidade de incorporar funções de transdução sintática. Como continuação dos trabalhos, foi desenvolvida uma versão mais prática dos algoritmos de conversão de gramáticas em reconhecedores, que foi publicado como um livro introdutório sobre compilação (NETO, 1987).

Neto (1988) apresenta o transdutor adaptativo que teve por objetivo facilitar a representação de tais problemas, que consiste numa classe de máquinas de estados finitos com memória organizada em pilha e que exibe recursos de aprendizado baseados na alteração dinâmica de sua configuração, com base nas transições efetuadas pelo transdutor.

Um dos principais resultados obtidos no desenvolvimento deste transdutor foi um aumento no poder de representação dos modelos matemáticos, empregados por meio da inclusão de recursos de aprendizado. Os transdutores adaptativos, assim idealizados, são capazes de se moldarem a cada sentença particular a ser

reconhecida, proporcionando uma forma extremamente natural para o tratamento de diversos problemas sintáticos usualmente considerados como parte da manipulação semântica da linguagem. Os resultados de uma investigação em que se procura obter, a partir de técnicas clássicas simples, mecanismos capazes de integrar em um único método de análise os elementos necessários à resolução plena e uniforme de grande parte dos problemas de interesse na análise de linguagens, referentes a aspectos puramente sintáticos podem ser encontrados em (NETO, 1993). Neste trabalho, foi apresentado o autômato e o transdutor adaptativo como dispositivos de reconhecimento e transdução sintática.

Com base no trabalho de Neto (1993), foram realizados outros trabalhos nos quais foi aplicada a tecnologia adaptativa no projeto de sistemas reativos. Almeida (1995) apresentou uma evolução da notação de *Statechart* (HAREL et al., 1987), na qual foram acrescentadas características provenientes da teoria de Autômatos Adaptativos.

O *Statechart* Adaptativo tem capacidade de modificar sua configuração em resposta às entradas impostas ao sistema por ele representado. O trabalho realizado por Almeida (1995) permitiu também a implementação de uma ferramenta de análise e desenvolvimento de aplicações valendo-se de *Statechart* Adaptativo. Essa ferramenta, intitulada STAD (STAD, 2005), constitui-se de um editor e de um simulador de sistemas que utiliza a notação desenvolvida. Além de ter propiciado uma visão prática da teoria do uso de tecnologia adaptativa, a ferramenta STAD comprovou a sua viabilidade, proporcionando assim uma forma bastante útil de difusão desses conceitos.

Um estudo que teve por objetivo melhorar a especificação de um conjunto de sistemas reativos complexos e sincronizados entre si pode ser encontrado em (NOVAES, 1997). Um dos resultados desse trabalho foi o desenvolvimento de um formalismo, o Stad-Sinc, que se fundamenta na junção das notações de Rede de Petri, de *Statechart* convencional e de *Statechart* Adaptativo. O novo formalismo criado permite representar por meio de diagramas os mecanismos de sincronização existentes nas aplicações projetadas.

Neto, Almeida e Novaes (1998) apresentaram uma ferramenta para o desenvolvimento e análise, intitulada STAD-S. Tal ferramenta constitui-se de um

editor de *Statechart* e de um simulador de *Statecharts* sincronizados. O Stad-S permite que um sistema seja considerado e diversos pontos de vista provenientes das características dos formalismos subjacentes ao formalismo desenvolvido. Sendo assim, os aspectos de hierarquia e concorrência fundamentam-se nas características do *Statechart*; os aspectos de sincronização são baseados no mecanismo de Rede de Petri; e os aspectos de aprendizagem utilizam-se dos conceitos de automodificação provenientes dos *Statecharts* Adaptativos. Desta forma, o Stad-S permite a representação de cada um desses aspectos isoladamente, bastando omiti-los, se forem desnecessários.

Um Ambiente de Desenvolvimento de Reconhecedores Sintáticos Baseado em Autômatos Adaptativos foi descrito em (PEREIRA, 1997). Este trabalho introduziu uma ferramenta de auxílio ao desenvolvimento de reconhecedores sintáticos denominada Reconhecedor Sintático para Window (RSW) (PEREIRA, NETO, 1999). A ferramenta RSW proporciona aos seus usuários um ambiente integrado, e os reconhecedores sintáticos reconhecidos e gerados pela ferramenta são baseados na teoria de autômato de estados finitos, autômatos de pilha estruturados e nos autômatos adaptativos. A possibilidade de implementação de reconhecedores baseados em Autômatos Adaptativos (NETO, 1993) constitui uma das importantes metas alcançadas pela ferramenta RSW, comprovando sua viabilidade prática.

Um formalismo dual ao autômato, o qual visava a facilitar o desenvolvimento de linguagens complexas ou outras aplicações que necessitassem especificar linguagens dependentes de contexto na forma de gramáticas foi denominado como Gramática Adaptativa (NETO, IWAI, 1998; IWAI 2000). Tal formalismo possui como característica principal a capacidade de se alterar a medida que vai sendo feita a geração da sentença pertencente à linguagem que é representada pela gramática adaptativa. Tal trabalho foi fruto da compilação de alguns trabalhos referentes às gramáticas adaptáveis (SHUTT, 1993, 1995; RUBINSTEIN, SHUTT, 1993, 1994, 1995), bem como de alguns formalismos correlatos dinâmicos que são utilizados na representação de linguagens, como por exemplo, os autômatos.

Rocha (2000) elaborou um estudo que visa o desenvolvimento de um método de construção de modelos e resolução de problemas complexos utilizando-se

dispositivos adaptativos. Para tal, foram realizados estudos comparativos entre autômatos adaptativos, redes neurais, algoritmos genéticos e agentes, extraindo destes dispositivos, características que permitiram a composição do modelo denominado Busca de Soluções por Máquina Adaptável (BSMA) (ROCHA; NETO, 2000).

Com base no método BSMA foi construído um simulador para Autômatos Adaptativos para a escolha de solução de problemas e apresentada uma proposta para o uso da tecnologia adaptativa na simulação de Redes Neurais em ambientes computacionais (ROCHA, 2001; ROCHA, NETO, 2001). Tais estudos serviram para demonstrar a possibilidade de utilização da tecnologia adaptativa em aprendizagem computacional e, de maneira geral, em inteligência artificial.

Freitas (2000) realizou um estudo referente à aplicação da tecnologia adaptativa no desenvolvimento de ambientes que suportem multilinguagens de programação. Este trabalho apresentou uma proposta de implementação de um ambiente que viabilize o emprego da programação multilinguagem por meio do oferecimento de primitivas que facilitem a interface entre os diversos segmentos de linguagens que compõem a aplicação. Tais primitivas estabelecem um mecanismo de gerenciamento de nomes relativos às diferentes variáveis que são importadas ou exportadas entre os módulos componentes da aplicação multilinguagem. Desta forma, ocorre a necessidade de um mecanismo que gerencie o espaço de nomes do ambiente. Neste contexto, Freitas e Neto (2000a) empregaram o autômato adaptativo como coletor de nomes. Embora simples, esta estrutura representa uma alternativa eficiente e elegante para representar as estruturas de dados de armazenamento e busca de cadeias. Para validar a proposta de implementação do ambiente multilinguagem, foi desenvolvido um Ambiente Multilinguagem (AML), no qual as linguagens *C++*, *Prolog*, *Lisp* e *Java* podem ser utilizadas concomitantemente (FREITAS; NETO, 2000b, 2001).

Em relação ao projeto de linguagens, Neto e Silva (2005) apresentaram uma estrutura adaptativa para design de linguagens de especificação de software. Neste trabalho foram apresentadas algumas características adaptativas presentes em linguagens de especificação, e também foi descrita uma estratégia para estender a especificação de linguagens de programação. Um exemplo simples também foi apresentado para demonstrar a estrutura proposta. Neto (2001) busca

características comuns presentes nos dispositivos adaptativos dirigidos por regras, o que permite a um especialista estender um dispositivo dirigido por regras para suportar tecnologia adaptativa. Com base nisso Camolesi e Neto (2003) apresentaram uma proposta de extensão do dispositivo *Interaction System Design Language (ISDL)* (QUARTEL, 1997) definindo então o dispositivo $ISDL_{Adp}$. O estudo também demonstra a aplicação da tecnologia adaptativa na modelagem de aplicações hipermídia. Em estudo posterior (CAMOLESI; NETO, 2004a), foi apresentada uma formulação completa do $ISDL_{Adp}$ e a aplicação de sua linguagem na modelagem de aplicações complexas

No trabalho desenvolvido por Camolesi e Neto (2004b) foi descrita a extensão de Rede de Petri Adaptativa (RP_{Adp}) conforme a formulação de (NETO, 2001). Neste estudo, propunha-se a definição de uma estrutura de dados para a representação da Rede de Petri Adaptativa. As etapas de extensão de um dispositivo adaptativo e uma estrutura de dados geral para a representação de dispositivos adaptativos dirigidos por regras pode ser encontrado em (CAMOLESI, 2005).

Pistori e Neto (2002) apresentaram o AdapTree - um algoritmo para indução de árvores de decisão que usa técnicas adaptativas - e os primeiros resultados da aplicação de técnicas adaptativas na produção de algoritmos de aprendizagem eficientes. Um protótipo de um sistema cuja interface com o usuário era feita pela direção do olhar utiliza técnicas de baixo custo, fundamentadas em aprendizado computacional e que usa tecnologia adaptativa pode ser encontrado em (PISTORI et al., 2003).

Pistori (2003) apresentou o Autômato de Estados Finitos Adaptativo e descreveu uma complementação para a representação de funções e ações adaptativas, além de uma integração de dispositivos adaptativos, basicamente discretos, com mecanismos que manipulam informações não discretas. Neste trabalho, também foram desenvolvidos alguns exemplos que demonstram a utilização de tecnologia adaptativa no desenvolvimento de aplicações. Um outro estudo, que fundiu conceitos de Autômato de Estados Finitos Adaptativo e de algoritmos genéticos, criando um ambiente propício para explorar o impacto de adaptação individual durante toda a vida em evoluções de populações foi apresentado em (PISTORI et al., 2005).

Um estudo focalizando a aplicação de tecnologia adaptativa na otimização de código em compiladores foi apresentado em (LUZ; NETO, 2003). Luz (2004) introduziu o uso de uma ação adaptativa de forma que o algoritmo de otimização automodificasse o seu comportamento em resposta a uma condição de entrada específica e procurasse a seqüência de regras de otimização que melhor se aplicasse ao código-objeto entre as muitas seqüências possíveis resultantes da superposição de duas ou mais regras de otimização igualmente aplicáveis.

A utilização de autômatos adaptativos para o mapeamento de movimentos de robôs móveis autônomos pode ser apreciado em (SOUSA; HIRAKAWA; NETO, 2004a e 2004b). Inicialmente, o robô possuía um pequeno mapa do ambiente que era ampliado por meio dos caminhos percorridos por ele mesmo. Para tal, foi construído um algoritmo que utilizava técnicas adaptativas que inicialmente adicionavam algumas marcas livres que eram modificadas com informações obtidas pelos sensores. Sousa e Hirakawa (2005) demonstram o funcionamento da navegação do robô utilizando esta estrutura.

O trabalho proposto neste projeto tem como base principal a Tecnologia Adaptativa, sendo esta, uma tecnologia com a característica de auto-adaptação. No trabalho desenvolvido por Pistori (2003) foi empregada a Tecnologia Adaptativa para facilitar a interação de jogadores que possuem deficiência motora. Foi desenvolvido um jogo tradicional, o “Jogo da Velha”, que permite ao jogador escolher o local onde jogará utilizando para isto a direção do olhar por meio de uma câmera. Outros trabalhos relacionados ao uso da Tecnologia Adaptativa empregada a jogos também foram realizados, porém estes foram apenas exemplos em sala de aula e não foram realizadas publicações sobre os mesmos. Neste sentido espera que o desenvolvimento desta pesquisa possa contribuir com publicações para esta área. O principal diferencial na Tecnologia Adaptativa é a forma razoavelmente simples que podemos transformar teorias já existentes, bem como fazer um reaproveitamento e estruturação destas para melhorar sua capacidade de respostas e interação.

O desenvolvimento níveis de dificuldades com Tecnologia Adaptativas faz com que seja facilitada a interação com o usuário, sem a necessidade de uma base de dados contemplando todas as possíveis reações de um jogador, porque cada regra inicial pode ser modificada de acordo com o ambiente ao qual está sendo

trabalhado de forma que não necessita de alguém programando isso a cada nova ocorrência encontrada.

Este conceito de auto-modificação da Tecnologia Adaptativa, torna a Inteligência Artificial (muito encontrada em jogos que exigem determinados tipos de respostas ao usuário), possa ser trabalhada de forma mais simples, e eficiente desde que seja feito de forma correta o seu desenvolvimento, seguindo passos que por mais simples que pareçam, mostrem uma complexidade no que diz a atenção dispensada para não ter um sistema com falhas futuras.

2.3. Tecnologia JAVA

A tecnologia Java começou a ser criada em 1991 com o nome de Green Project, nele trabalhavam James Gosling, Mike Sheridan e Patrik Naughton, e tinham como objetivo principal criar um interpretador para dispositivos eletrônicos (WESTER, 2013). No verão de 1992 eles emergiram de um escritório de Sand Hill Road no Menlo Park com uma demonstração funcional da ideia inicial. O protótipo se chamava *7 (StarSeven), um controle remoto com uma interface gráfica touchscreen, que tinha a habilidade de controlar diversos dispositivos e aplicações. James Gosling especificou uma nova linguagem de programação para o *7 e decidiu batizá-la de Oak, que quer dizer carvalho, uma árvore que ele podia observar quando olhava pela sua janela (WESTER, 2013).

Java é uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros. O Java é executado em mais de 850 milhões de computadores pessoais e em bilhões de dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão.

As vantagens dessa linguagem de programação é decorrente do fato delas ser orientada a objeto, portável entre diferentes plataformas e sistemas operacionais. Os funcionamentos dessa linguagem obedecem aos seguintes princípios:

- Todos os programas Java são compilados e interpretados;
- O compilador transforma o programa em bytecodes independentes de plataforma;
- O interpretador testa e executa os bytecodes;
- Cada interpretador é uma implementação da JVM – Java Virtual Machine.

A figura 1 mostra a sequencia do funcionamento da linguagem.

Figura 1 – Funcionamento da linguagem Java.

Uma plataforma é o ambiente de hardware e software onde um programa é executado. A plataforma Java é um ambiente somente de software.

De acordo com WESTER (2013), as suas principais características da linguagem são:

- Orientação a objetos: programas em Java são baseados na composição e interação entre diversas unidades de software chamadas de objetos.
- Independente de plataforma: O byte-code gerado pelo compilador para uma aplicação específica pode ser transportado entre plataformas distintas que suportam Java (Solaris, Windows, Mac OS, Linux etc). Não é necessário recompilar um programa para que ele execute em máquinas com sistemas operacionais diferentes.
- Segura: A presença de coleta automática de lixo (*garbage collection*) evita erros comuns que os programadores comentem quando são obrigados a gerenciar

diretamente a memória. Os mecanismos de tratamento de exceções tornam as aplicações mais robustas.

- Suporte à concorrência: Permite de maneira fácil a criação de vários threads de execução amplamente usados em animações e processamento paralelo.
- Suporte para programação de sistemas distribuídos: Java fornece facilidades para programação de *sockets*, TCP/IP, chamadas de métodos remotamente, entre outras.

Com o tempo, o Java foi amadurecendo e vislumbrando possibilidades em outros setores da indústria além da Internet e, reconhecendo a impossibilidade de criar uma plataforma única capaz de abranger completamente as demais áreas de mercado, a Sun2 dividiu a tecnologia em três edições, cada uma visando segmentos específicos de negócio (ORACLE, 2013). Essas três edições são:

- **JSE (Java Standard Edition):** projetada para executar em computadores pessoais (desktops) e estações de trabalho.
- **JME (Java Micro Edition):** especializada em pequenos dispositivos com memória, tela e poder de processamentos limitados.
- **JEE (Java Enterprise Edition):** projetada com foco em aplicações para serem executadas no servidor.

A figura 2 apresenta um diagrama com uma visão geral da plataforma Java:

Figura 2 - Plataforma Java e suas edições (RIZZI e ROSA, 2010)

2.3.1 – Máquina Virtual

Em uma linguagem de programação como C e Pascal, pode se verificar a seguinte situação ao compilar um programa, conforme mostra a figura 3 (CAELUM, 2013).

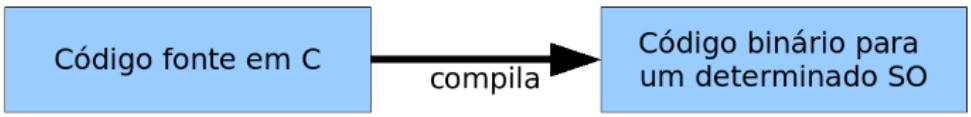

Figura 3 – Compilação de um programa em C e Pascal.

O código fonte é compilado para código de máquina específico de uma plataforma e sistema operacional. Muitas vezes o próprio código fonte é desenvolvido visando uma única plataforma. Esse código executável (binário) resultante será executado pelo sistema operacional e, por esse motivo, ele deve saber conversar com o sistema operacional em questão, conforme mostra a figura 4 (CAELUM, 2013).

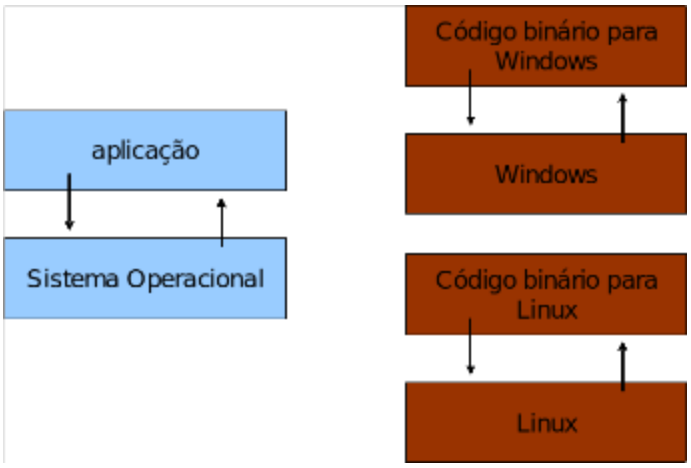

Figura 4 – Arquitetura de funcionamento do código executável.

Isto significa que é obtido um código executável para cada sistema operacional. E é necessário compilar uma vez para Windows, outra para o Linux, e assim por diante, caso o usuário queira que esse o software possa ser utilizado em várias plataformas. Esse é o caso de aplicativos como o OpenOffice, Firefox e outros.

O que se pode verificar é que na maioria das vezes, a sua aplicação se utiliza das bibliotecas do sistema operacional, como, por exemplo, a de interface gráfica para desenhar as "telas". A biblioteca de interface gráfica do Windows é bem diferente das do Linux. Para criar uma aplicação que execute de forma parecida nos dois sistemas operacionais é necessário reescrever um mesmo pedaço da aplicação para diferentes sistemas operacionais, já que eles não são compatíveis.

Já a linguagem Java utiliza do conceito de máquina virtual, onde existe, entre o sistema operacional e a aplicação, uma camada extra que é responsável por "traduzir", mas não apenas isso, o que sua aplicação deseja fazer para as respectivas chamadas do sistema operacional onde ela está executando no momento. A figura 5 mostra a arquitetura dessa camada.

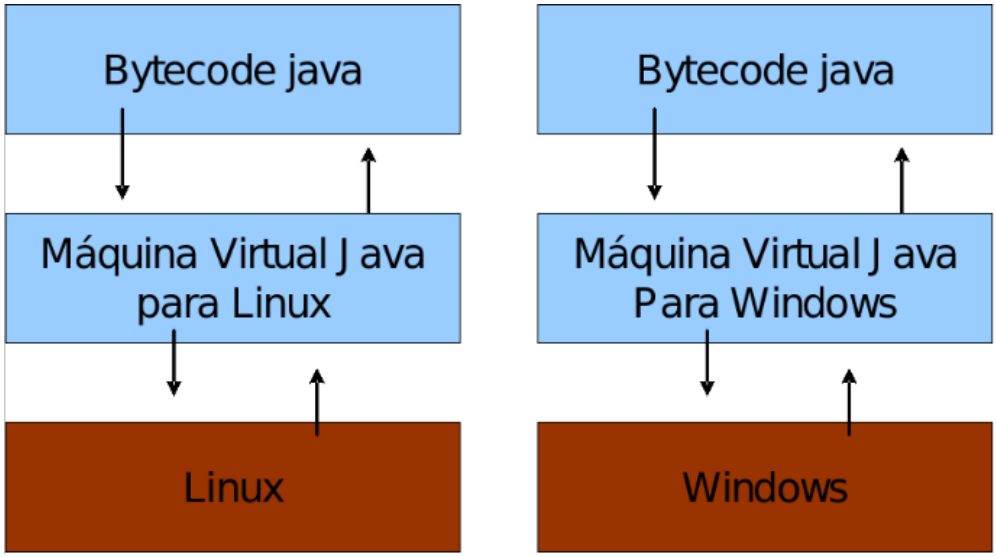

Figura 5 - Arquitetura da Camada da máquina virtual (CAELUM, 2013)

Dessa forma, a maneira com a qual se abre uma janela no Linux ou no Windows é a mesma, e com isso se ganha independência de sistema operacional. Ou, melhor ainda, independência de plataforma em geral já que não é preciso se preocupar em qual sistema operacional sua aplicação está executando, nem em que tipo de máquina, configurações, etc.

O conceito de uma máquina virtual é bem mais amplo que o de um interpretador. Como o próprio nome diz, uma máquina virtual tem tudo que um computador tem. Em outras palavras, ela é responsável por gerenciar memória, threads, a pilha de execução, etc. Sua aplicação executa sem nenhum envolvimento com o sistema operacional, onde conversa apenas com a *Java Virtual Machine* (JVM).

Essa característica é interessante já que tudo passa pela JVM, onde ela pode tirar métricas, decidir onde é melhor alocar a memória, entre outros. Uma JVM isola totalmente a aplicação do sistema operacional. Se uma JVM termina abruptamente, só as aplicações que estavam executando nela irão terminar e isso não afetará as outras JVMs que estejam executando no mesmo computador, nem afetará o sistema operacional. Essa camada de isolamento também é interessante quando se pensa em um servidor que não pode se sujeitar a executar código que possa interferir na boa execução de outras aplicações (DEITEL, 2003). Essa camada, a máquina virtual, não entende código java, ela

entende um código de máquina específico. Esse código de máquina é gerado por um compilador java, como o javac, e é conhecido por *bytecode,* pois existem menos de 256 códigos de operação dessa linguagem, e cada *opcode* gasta um *byte*. O compilador Java gera esse *bytecode* que, diferente das linguagens sem máquina virtual, vai servir para diferentes sistemas operacionais, já que ele vai ser "traduzido" pela JVM (DEITEL, 2010).

3 – Desenvolvimento do Projeto

Neste capítulo serão descritos todos os detalhes do desenvolvimento do projeto. Serão feitos uma descrição do problema e a de sua modelagem, além de detalhar a implementação de uma aplicação em nuvem.

3.1 - Descrição do Problema

Neste projeto foi desenvolvido várias aplicações em nuvem e em algumas delas aplicado os conceitos de tecnologia adaptativa para verificar se tais problemas tem a necessidade de serem em nuvem com conceitos de tecnologia adaptativa.

Um dos software desenvolvidos foi uma agenda que quando for cadastrado uma pessoa, essa pessoa poderia ter um, dois, três ou dez telefones que a aplicação funciona normalmente. Caso essa pessoa necessite de outro telefone basta buscar a pessoa e cadastrar novos telefones que a agenda é atualizada na hora. Essa aplicação foi desenvolvida na linguagem Java utilizando banco de dados PostgreSQL.

3.2 – Modelagem do Problema

A modelagem do problema é de suma importância, pois ela dá uma visão geral do que se pretende desenvolver. Quando os dados ficam armazenados na nuvem, permite que o usuário acesse seus dados de qualquer lugar do planeta, desde que ele tenha acesso a internet, ele pode acessar seus dados através de um *smartphone,* notebook, tablete, computador ou qualquer outro dispositivo que tenha acesso a internet. A figura 6 mostra a modelagem do problema.

Figura 6 – Modelagem do problema

Para facilitar a implantação do projeto o seu desenvolvimento foi dividido em quatro módulos:

3.3 – Desenvolvimento das aplicações

Nesta seção, será apresentada uma descrição detalhada de todos os módulos do desenvolvimento do ambiente computacional.

3.3.1 – Módulo 1: Estudo da Tecnologia Adaptativa

A tecnologia adaptativa foi escolhida por ser um conceito que resolve vários problemas muito complexos. Uma das suas vantagens é a simplicidade e facilidade, pois conforme os estímulos de entrada o programa pode ser alterado

dinamicamente pela aplicação. Dispositivos adaptativos são empregados em uma grande variedade de situação, principalmente quando os problemas são muito complexos.

Inicialmente foi realizado um estudo sobre teoria da computação para podermos começar a entender como funciona a tecnologia adaptativa, depois foi realizado um estudo sobre autômato, com isso começamos a entender o que podemos chamar de dinâmico, adaptável e adaptativo.

3.3.2 – Módulo 2: Estudo de Computação em Nuvem

Para podermos começar a desenvolver aplicações que poderiam executar na nuvem tivemos que realizar um estudo aprofundado sobre computação em nuvem, para sabermos quando uma aplicação pode estar na nuvem, se a aplicação pode ser local e o banco em nuvem.

A disponibilidade das aplicações em nuvem podemos dividir em três partes, publica, privada e hibrida. A nuvem pública qualquer usuário pode acessar os dados de qualquer lugar do mundo, a nuvem privada, apenas usuários com permissão podem acessar os dados, já a nuvem hibrida alguns usuários podem se conectar e outros não, ou alguns usuários podem verificar qualquer informação que fica armazenada em determinado local e outros não podem acessar nada ou apenas algumas. A figura 7 os tipos de nuvem.

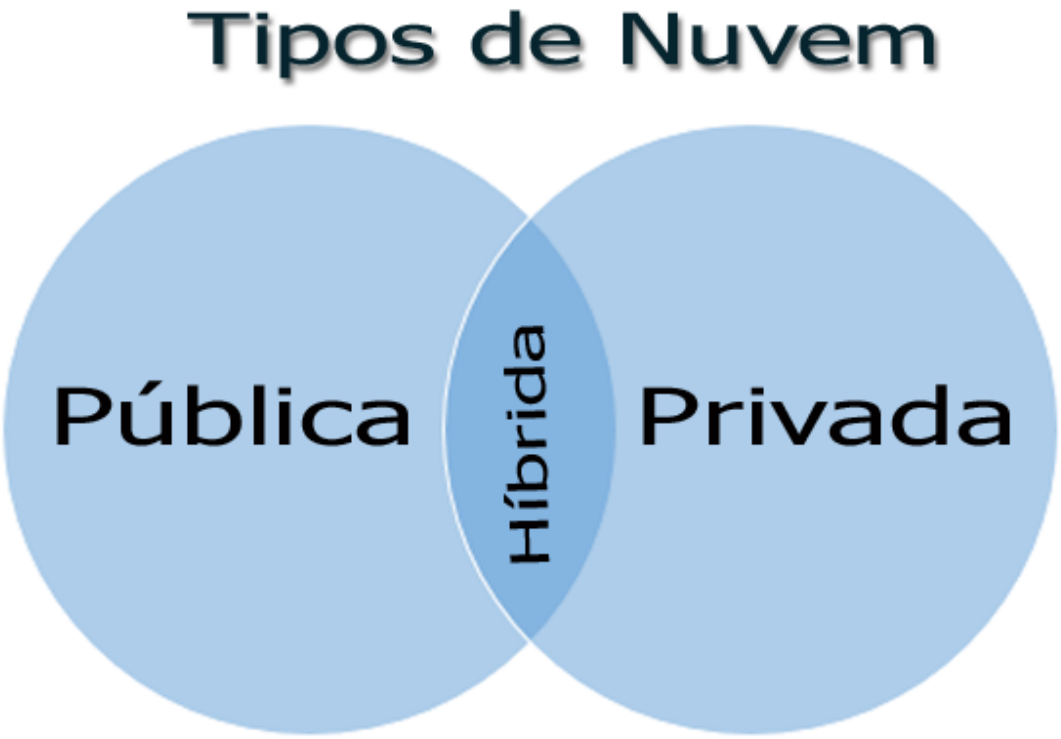

Figura 7 – Tipos de nuvem

3.3.3 – Módulo 3: Desenvolvimentos das Aplicações

Com os conhecimentos adquiridos nos módulos 1 e 2 foram desenvolvidas algumas aplicações web para ser colocadas na nuvem, algumas apenas banco de dados em nuvem, outras apenas aplicação na nuvem e banco de dados local e outras tudo em nuvem, aplicação e banco de dados.

Umas das preocupações era com que essas aplicações ficassem dinâmicas. A aplicação que ficou melhor para demonstrar o dinamismo foi a agenda, cujo as pessoas são cadastras e não importa quanto telefones essas pessoa tem, a aplicação se adapta ao usuário que esta sendo cadastrado. Na figura 8 será apresentado a tela principal do sistema, onde temos duas opções, cadastrar o consultar.

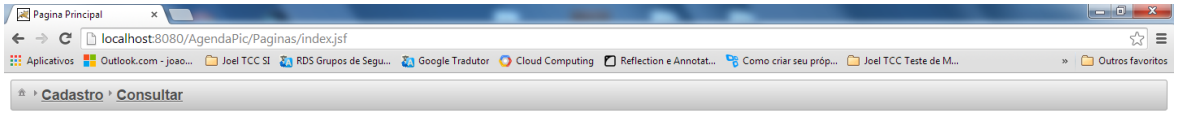

Figura 8 – Interface principal do sistema.

A figura 9 mostra a interface ao selecionar a opção cadastrar. Para cada pessoa cadastrada podemos gravar o um numero infinito de telefones, tanto residencial, celular ou comercial. Caso queira cadastrar um telefone de uma pessoa já cadastrada temos que apenas entrar na aba cadastrar e ir direto em buscar pessoa, para cadastrar novos telefones.

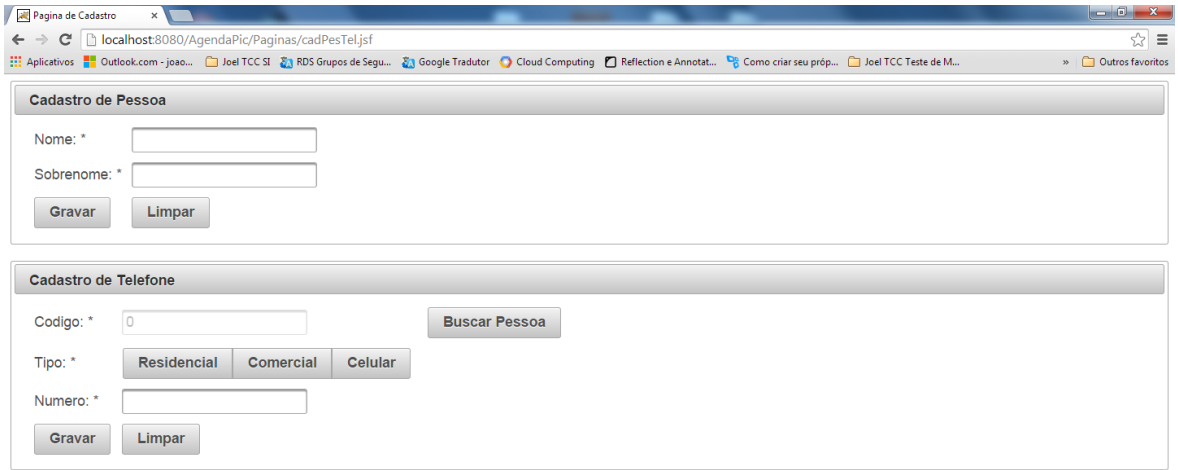

Figura 9 – Interface de cadastro

Após o usuário cadastrar a pessoa que deseja salvar seus contatos ou o usuário queira cadastrar novos telefones para uma pessoa já cadastrada basta ele escolher a opção cadastro novamente e na tela de cadastro escolher a opção buscar pessoa na guia Cadastro de Telefone, quando ele escolher essa opção aparecerá todas as pessoas já cadastradas, basta ele escolher a opção selecionar do lado direito do nome da pessoa, como mostra a figura 10:

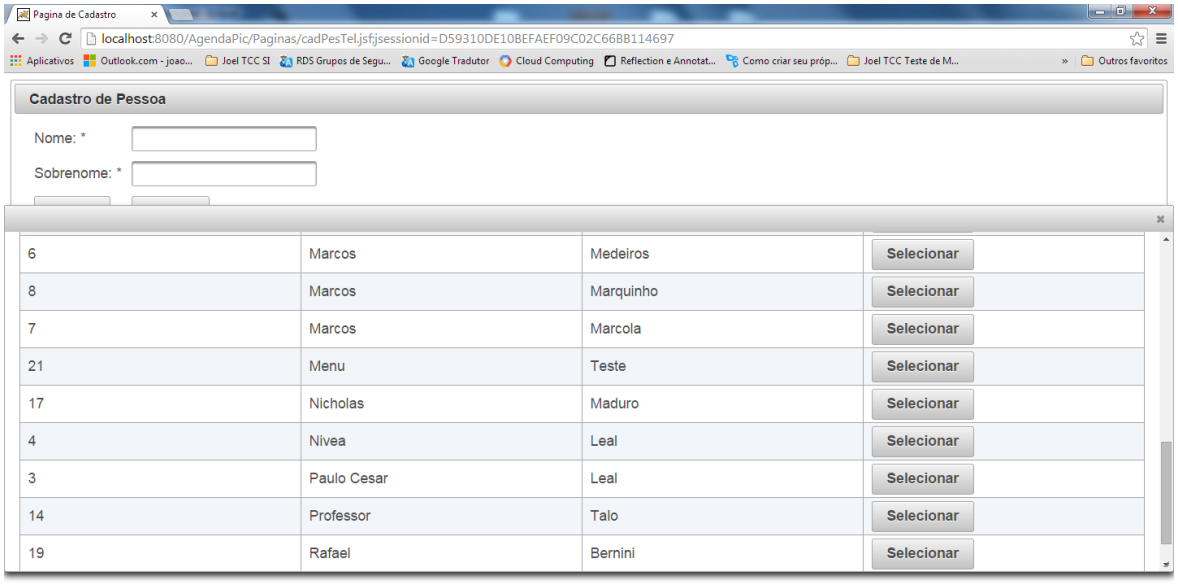

Figura 10 - Selecionando pessoa a qual será atribuído o telefone.

Depois de escolher a pessoa a qual será atribuído número do telefone, o usuário escolher se o telefone é residencial, comercial ou celular, informa qual é o número e clica no botão gravar, como mostra a figura 11:

Figura 11 - Cadastro do número de uma pessoa já selecionada

A figura 12 mostra a interface ao selecionar a opção consultar na tela principal, onde tem todos as pessoas cadastradas e podem ser consultadas por nome e sobrenome, as duas opções de busca são case-sensitive.

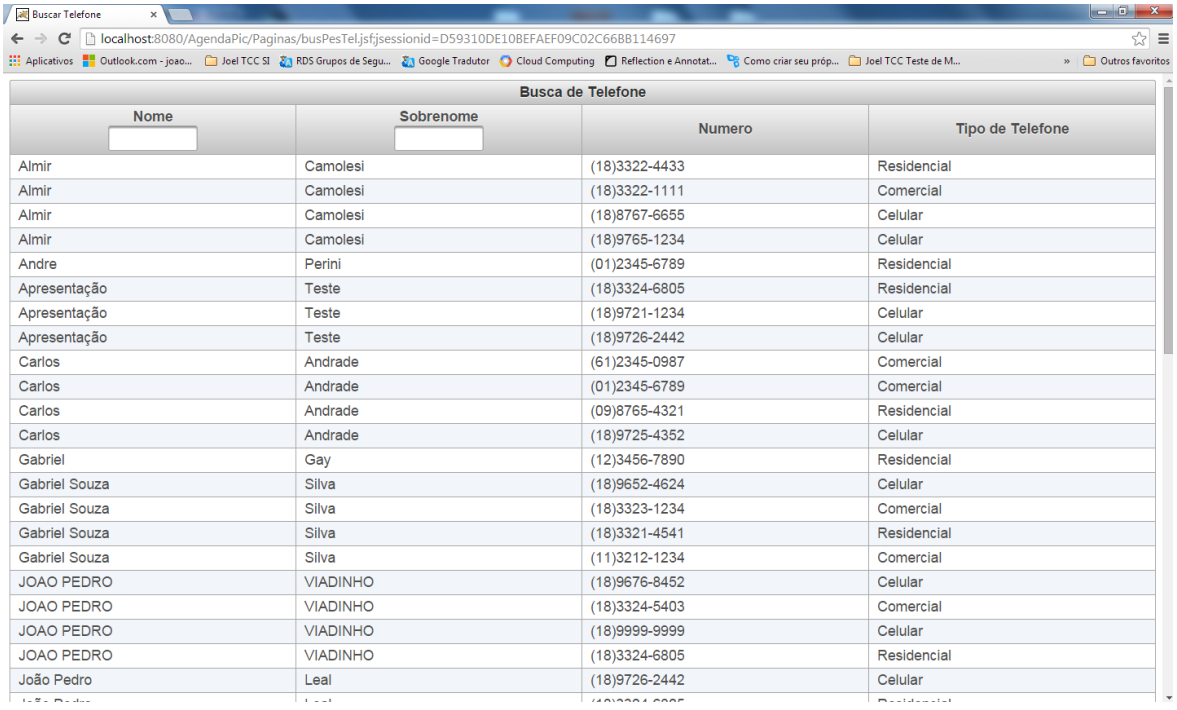

Figura 12 - Busca de todas as pessoas cadastradas.

A figura 13 mostra a busca por nome, na qual todas as pessoas com nome igual aparecem na lista.

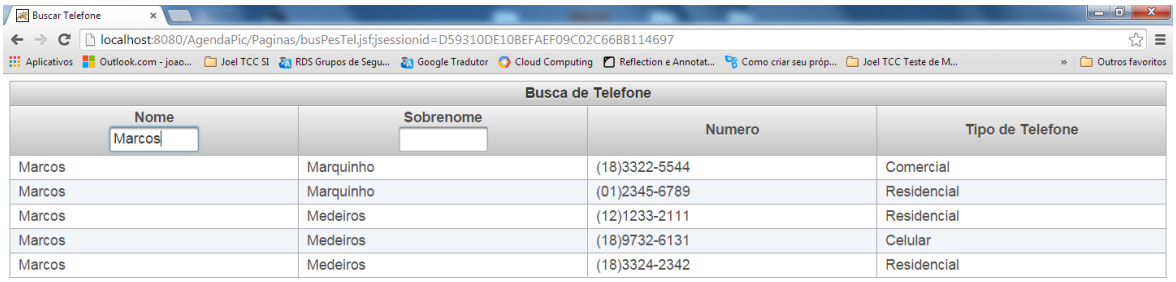

Figura 13 - Busca por nome

A figura 14 mostra a busca por nome e sobrenome, a qual o nome Marcos foi o escolhido para demonstrar porque existe mais de um Marcos cadastrado com sobrenome diferente.

Figura 14 - Busca por nome e sobrenome.

3.3.4 – Módulo 4: Subindo Aplicações na Nuvem

Neste módulo foram colocadas algumas aplicações na nuvem, a aplicação da agenda colocamos apenas banco de dados na nuvem e aplicaçãoo localhost. Na figura 15 será mostrado um trecho do código fonte onde o banco de dados está em nuvem a aplicação localhost. Para utilizar o banco de dados em nuvem basta contratar um serviço na nuvem e alterar apenas uma linha de código na aplicação web, mas isso depois da nuvem está toda configurada para que sua aplicação possa executar.

Conexao.java

```
1  package uteis;
2
3  import java.sql.Connection;
5
6  public class Conexao {
7      public static Connection getConnection(){
8          Connection con = null;
9          try {
10             Class.forName("org.postgresql.Driver");
11             con = DriverManager.getConnection("jdbc:postgresql://down1.warzombie.com.br:5432/cloud","postgres","masterkey");
12             //con = DriverManager.getConnection("jdbc:postgresql://localhost:5432/TestePic","postgres","123456");
13             System.out.println("Conectado com Sucesso");
14         } catch (Exception e) {
15             System.out.println("Erro na conexao "+e.getMessage());
16         }
17         return con;
18     }
19 }
20
```

Figura 15 – Método de conexão com banco de dados.

Nesta seção também foram colocadas aplicações para executar no *Google Cloud Compting e* o banco de dados da *AWS (Amazon Web Services),* essa tentativa não foi executada com êxito pois se utilizar aplicação da google tenho que utilizar o banco de dados deles também, o maior problema é que a aplicação na google não é paga para teste, já o banco de dados é pago. Na AWS o banco de dados é gratuito até uma certa utilização, a figura 16 mostra as ferramentas da AWS que foram utilizadas e a figura 17 as ferramentas da google que também foram utilizadas.

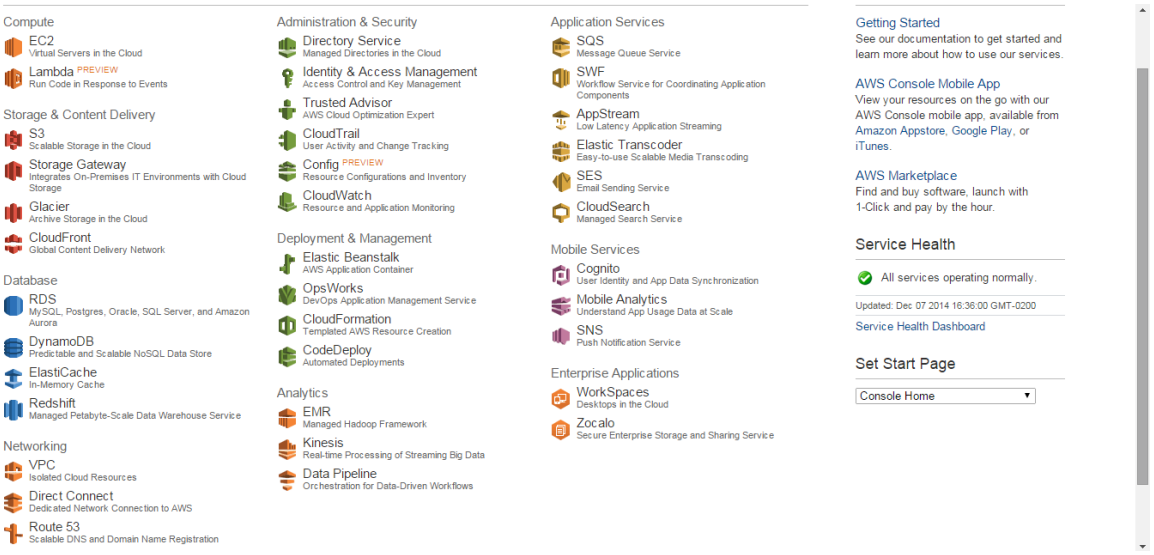

Figura 16 - Ferramentas AWS

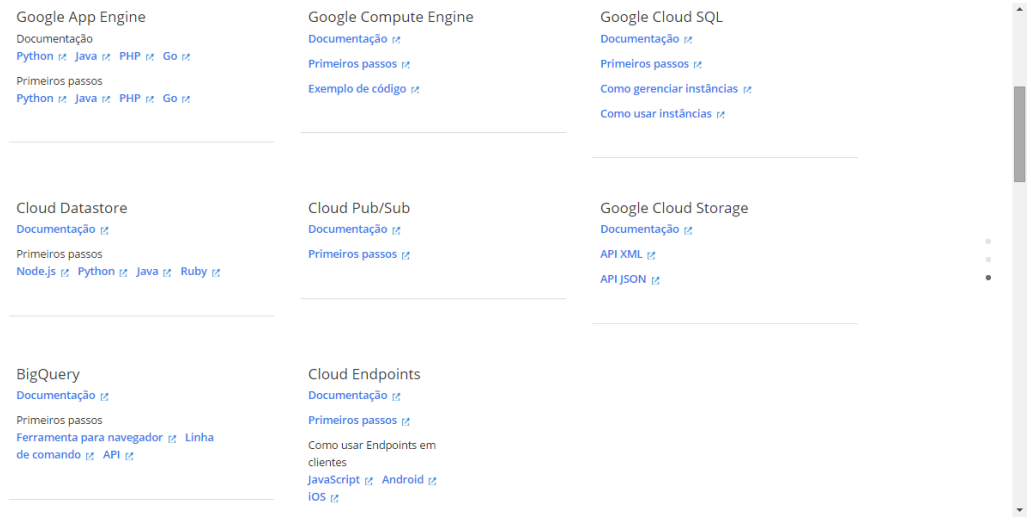

Figura 17 - Ferramentas Google

A figura 18 mostra o banco de dados utilizado na amazon para realizar testes, o mesmo funcionou executando com uma aplicação localhost. A ferramenta utilizada foi Relational Database Service (RDS).

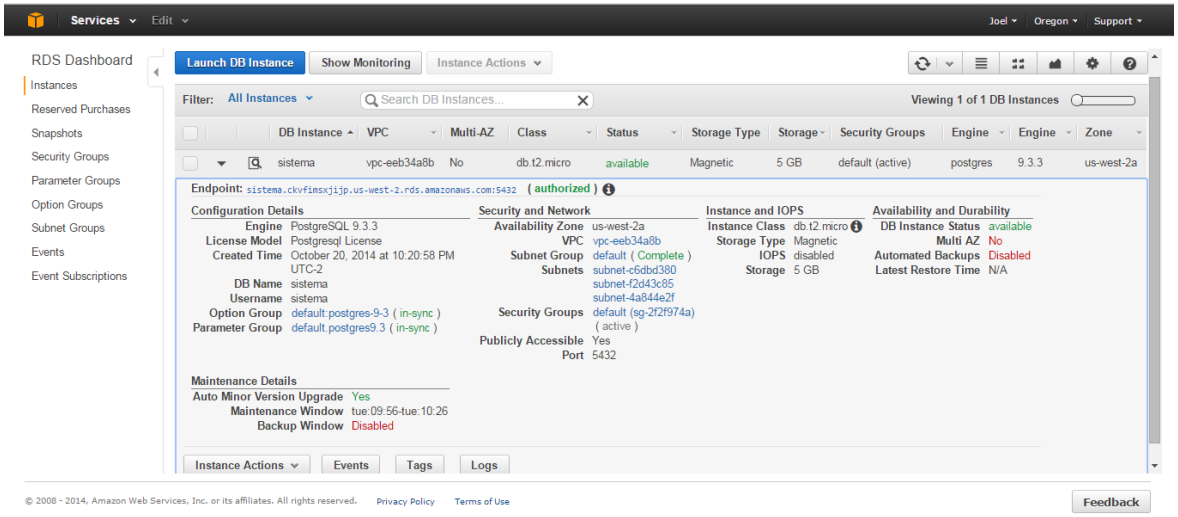

Figura 18 - RDS criado na amazon

A figura 19 mostra algumas aplicações da *Google App Engine* que foram criadas

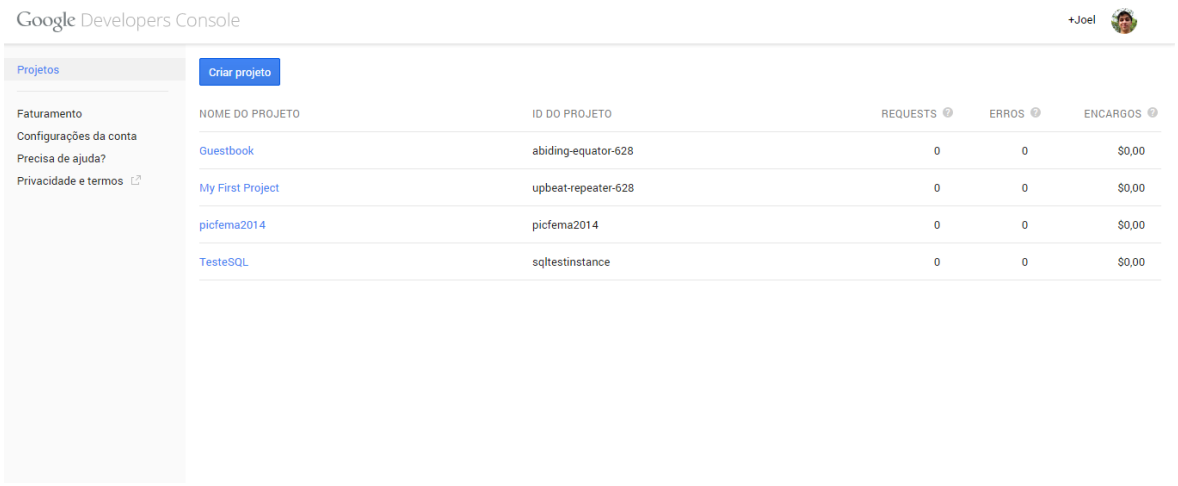

Figura 19 - Aplicação criadas *Google App Engine*

4 – Conclusão

O desenvolvimento deste projeto de pesquisa foi de essencial importância para aquisição de conhecimentos práticos e teóricos. As pesquisar realizadas sobre computação em nuvem e tecnologia adaptativa foram realizadas com sucesso. No final desse projeto temos o conhecimento para dizer quando uma aplicação é dinâmica, adaptável e adaptativa, pois os conceitos de cada uma tem varias semelhanças, também temos um aplicativo de uma agenda dinâmica que podemos cadastrar quantos números de telefone bem entendermos para cada pessoa.

Durante o desenvolvimento do projeto foi possível ter a percepção de como se deve conduzir com responsabilidade um projeto. Além disso, foi possível adquirir experiência com relação às dificuldades encontradas para saber qual a melhor metodologia e o melhor algoritmo para solucionar o problema, dentre tantas existentes. Foram realizados vários testes para saber qual era o melhor, qual o mais rápido, e o mais eficiente, no sentido de atingir satisfatoriamente os objetivos traçados no projeto. Futuramente pretende-se concluir a implementação de um software executando tudo na nuvem (Infraestrutura, Banco de dados e aplicação) com os conceitos de tecnologia adaptativa

Uma das maiores contribuições com o desenvolvimento deste projeto foi o conhecimento adquirido, além do crescimento profissional e acadêmico.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, J.R. **STAD - Uma ferramenta para representação e simulação de sistemas através de statecharts adaptativos.** Tese de Doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, 1995.

CAELUM, **FJ -11 Java e orientação a objetos**. Ensino e Inovação Caelum. 2013.

CAMOLESI, A.R. **Modeling a tool for the generation of programming environments for adaptive formalism.** International Conference on Adaptive and Natural Computing Algorithms (ICANNGA 2005), Springer Verlag editor, pp. 341-344, Coimbra University, Coimbra, Portugal, 21 - 23 march 2005.

CAMOLESI, A.R.; NETO, J.J. **Modelagem Adaptativa de Aplicções Complexas**. XXX Conferencia Latinoamericana de Informática - CLEI'04. Arequipa - Peru, Setiembre 27 - Octubre 1, 2004a.

CAMOLESI, A.R.; NETO, J.J. **Representação Intermediária para Dispositivos Adaptativos Dirigidos por Regras**. 3rd International Information and Telecommunication Technologies Symposium, UFSCar, São Carlos, Brasil, 2004b.

CAMOLESI, A.R.; NETO, J.J. **An adaptive model for specification of distributed systems.** IX Congreso Argentino de Ciencias de la Computación, La Plata, Argentina, 6-10 de Outubro, 2003.

DEITEL, H.M.,: **Java Como Programar**, trad. Carlos Arthur Lang Lisbôa, 4ed. Bookman, Porto Alegre, 2003.

DEITEL, H.M.: **Java Como Programar**, trad. Edson Furmankiewicz, 8ed. Pearson, São Paulo. 2010.

FREITAS, A.V. **Aspectos do projeto e implementação de ambientes multilinguagens de programação.** Dissertação de Mestrado, USP, São Paulo, 2000.

FREITAS, A.V.; NETO, J.J. **Aspectos da Implementação de um Ambiente Multilinguagem de Programação.** Anais do CACIC2000 - VI Congreso Argentino de Ciencias de la Computación. Ushuaia, Argentina, 2000a.

HAREL D. et al. **On the formal semantics of statecharts.** In: Symposyum on logic in Computer Science, 2°, Ithaca, Proceedings, IEEE Press, pp. 54-64, New York, 1987.

LUZ, J.C. **Tecnologia adaptativa aplicada à otimização de código em compiladores**. Dissertação de Mestrado, USP, São Paulo, 2004.

LUZ, J.C.; NETO, J.J. **Tecnologia adaptativa aplicada à otimização de código em compiladores.** IX Congreso Argentino de Ciencias de la Computación, La Plata, Argentina, 6-10 de Outubro, 2003.

MILLER, Michael. **Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online**. Que Editor, ISBN: 0789738031, 2008.

NETO J.J. **Introdução a Compilação**. EDITORA: LTC, Rio de Janeiro, 1987

NETO, J.J. **Adaptive Rule-Driven Devices - General Formulation and Case Study**. Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Springer-Verlag, Vol.2494, pp. 234-250, Pretoria, South Africa, July 23-25, 2001.

NETO, J.J. **Contribuições à metodologia de construção de compiladores**. Tese de Livre Docência, USP, São Paulo, 1993.

NETO, J.J. **Cross-Assemblers para Microprocessadores** - Geração Automática Através do SPD. I CONAI - Congresso Nacional de Automação Industrial, pp. 501-509, São Paulo, 1983.

NETO, J.J. e MAGALHÃES, M.E.S. **Um Gerador Automático de Reconhecedores Sintáticos para o SPD.** VIII SEMISH - Seminário de Software e Hardware, pp. 213-228, Florianópolis, 1981.

NETO, J.J. **Uma Solução Adaptativa para Reconhecedores Sintáticos**. Anais EPUSP - Engenharia de Eletricidade - série B, vol. 1, pp. 645-657, São Paulo, 1988.

NETO, J.J.; ALMEIDA Jr.J.R.; NOVAES, J.M. **Synchronized Statecharts for Reactive Systems**. Proceedings of the IASTED International Conference on Applied Modelling and Simulation, pp.246-251, Honolulu, Hawai, 1998.

NETO, J.J.; IWAI, M.K. **Adaptive Automata for Syntax Learning**. CLEI 98 - XXIV Conferencia Latinoamericana de Informatica, MEMORIAS. pp. 135-149, Quito, Equador, 1998.

NETO, J.J.; SILVA, P.S.M. **An adaptive framework for the design of software specification language**. Proceedings of the International Conferenceon Adaptive and Natural ComputINing Algorithms (ICANNGA), Springer Verlag editor, pp. 349-352, Coimbra University, Coimbra, Portugal, 21 - 23 march 2005.

NOVAES, J.M. **Um formalismo adaptativo com mecanismo de sincronização para aplicações concorrentes.** Dissertação de Mestrado, USP, São Paulo, 1997.

ORLANDO, Dan. **Modelos de Serviços de Computação em Nuvem, Parte 2: Plataforma como Serviço. Enterprise RIA Consultant, Vision Media Group.** Disponível em**:** <http://www.ibm.com/developerworks/br/cloud/library/cl-cloudservices2paas/>. Acesso em 05 jun. 2012.

PEREIRA, J.C.D.; NETO, J.J. **Um Ambiente de Desenvolvimento de Reconhecedores Sintáticos Baseado em Autômatos Adaptativos**. II Simpósio Brasileiro de Linguagens de Programação - SBLP97, pp. 139-150, Campinas, 1997.

PISTORI H. et al. **Adaptive finite states automata and genetic algorithms: merging individual adaptation and population evolution**. International Conference on Adaptive and Natural Computing Algorithms (ICANNGA), Springer Verlag editor, pp. 333-336, Coimbra University, Coimbra, Portugal, 21 - 23 march 2005.

PISTORI, H. **Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações**. Tese de Doutorado, USP, São Paulo, 2003.

PISTORI, H.; NETO, J.J. AdapTree - **Proposta de um Algoritmo para Indução de Árvores de Decisão Baseado em Técnicas Adaptativas**. Anais Conferência Latino Americana de Informática - CLEI 2002, Montevideo, Uruguai, Novembro, 2002.

PISTORI, H.; NETO, J.J.; COSTA, E.R. **Utilização de Tecnologia Adaptativa na Detecção da Direção do Olhar**. SPC Magazine, v.2., n.2, pp.22-29, Lima, Perú, Maio, 2003.

QUARTEL, D**. Actions Relations – Basic design concepts for behaviour modelling and refinement.** Ph.D Thesis Twente University, Netherlands, 1997.

RIZZI, F.A. e ROSA, W.A.: **Uma ferramenta web para interação com deficientes auditivos**. Trabalho de Conclusão de Curso. Centro Universitário Vila Velha. Vila Velha, 2010.

ROCHA, R.L.A. **Um método de escolha automática de soluções usando tecnologia adaptativa.** Tese de Doutorado, USP, São Paulo, 2000.

ROCHA, R.L.A. **Uma proposta de uso de tecnologia adaptativa para simulação de redes neurais em um dispositivo computacional**. In: IX Encuentro Chileno de Computación 2001, Punta Arenas. Proceedings of the Encuentro Chileno de Computación. Punta Arenas: Universidad de Magallanes, v. CD-ROM, pp. 1-9, 2001.

ROCHA, R.L.A.; NETO, J.J. **Construção e Simulação de Modelos Baseados em Autômatos Adaptativos em Linguagem Funcional**. Proceedings of ICIE 2001 - International Congress on Informatics Engineering, Buenos Aires: Computer Science Department - University of Buenos Aires, v. CD-ROM, pp. 509-521, 2001.

ROCHA, R.L.A.; NETO, J.J**. Uma proposta de método adaptativo para a seleção automática de soluções**. Proceedings of ICIE Y2K - International Congress on Informatics Engineering, Buenos Aires, 2000.

SHUTT, J.N. **Self-Modifying Finite Automata - Power and Limitations.** Worcester, Massachusetts, 1995.

SOUSA, M.A.A.; HIRAKAWA, A.R. **Robotic mapping and navigation in unknown environment using adaptive automata**. International Conference on Adaptive and Natural Computing Algorithms (ICANNGA), Springer Verlag editor, pp 345-348, Coimbra University, Coimbra, Portugal, 21 - 23 march 2005.

SOUSA, M.A.A.; HIRAKAWA, A.R.; NETO, J.J. **Adaptive automata for mobile robotic mapping**. Proceedings of VIII Brazilian Symposium on Neural Networks - SBRN'04. São Luís/MA - Brazil. September 29 - October 1, 2004a.

SOUSA, M.A.A.; HIRAKAWA, A.R.; NETO, J.J. **Adaptive automata for mapping unknown environments by mobile robots**. Lecture Notes in Artificial

Intelligence. C. Lemaître, C. A. Reyes, J. A. González (Eds.): Advances in Artificial Intelligence - IBERAMIA 2004, Springer-Verlag, pp 562-571, 2004b.

STAD. **State-Charts Adaptativos**. em **www.pcs.usp.br/lt**a , visitado em Janeiro de 2009.

VELTE, Anthony T; VELTE, Toby J.; ELSENPETER, Robert. **Computação em Nuvem.** Rio de Janeiro: Alta Books Editora, 2010.

WESTER, I. **Linguagem Java**. Disponível em: <http://www.infowester.com/lingjava.php>. Acessado em junho de 2013.