

UESB – Universidade Estadual do Sudoeste da Bahia
Bacharelado em Ciência da Computação

WebLab – Aplicação nas nuvens para Laboratório de Análises Clínicas

Elias Adriano Nogueira da Silva

Vitória da Conquista

2010

Elias Adriano Nogueira da Silva

WebLab – Aplicação nas nuvens para Laboratório de Análises Clínicas

Monografia apresentada à
Universidade Estadual do
Sudoeste da Bahia como
requisito para obtenção do
título de bacharel em Ciência
da Computação.
Orientador: Prof. Dr. Fábio
Moura Pereira

Vitória da Conquista

2010

AGRADECIMENTOS

Agradeço muito a ...

Deus por ter me dado força para continuar na graduação durante os anos mais difíceis da minha vida.

Minha Família, em especial minha mãe Abma pelo apoio incondicional e minhas tias Maria Helena e Maria do Socorro pela força durante muitos momentos difíceis.

Valdirando e Paulo Paixão por grandes lições que carregarei por toda vida.

Minha namorada Jéssica por estar sempre do meu lado.

Meu orientador Fábio Moura por ter compreendido minhas dificuldades e ajudado a realizar este trabalho.

Todos os professores que ajudaram na minha formação.

Todas as pessoas que fizeram parte da minha vida e me apoiaram na conquista de mais esta vitória.

Dedico este trabalho a todos que acreditaram em meu potencial, em especial Abma Nogueira Lima.

RESUMO

A computação em Nuvem surge como um novo paradigma da computação e traz benefícios tanto para os desenvolvedores de sistema quanto para usuários das soluções que utilizam esta tecnologia, a nuvem é uma metáfora para a infraestrutura necessária às aplicações. O trabalho aqui proposto apresenta o desenvolvimento de um caso de uso de uma aplicação nas nuvens para laboratório de análises clínicas.

O WebLab é um sistema para gerenciamento de laboratório que facilita o controle de exames, clientes e a entrega de resultados. A aplicação desenvolvida neste trabalho utiliza a arquitetura orientada a serviços e está hospedada em um servidor virtualizado.

Palavras-chave: Computação em nuvem, Sistemas Web, Sistemas Laboratoriais.

ABSTRACT

The Cloud computing has emerged as a new computing paradigm and brings benefits to both system developers and for users of the solutions with this technology, the cloud is a metaphor for the necessary infrastructure to applications. The work proposed here presents the development of a use case an application in the clouds for clinical analysis laboratory.

The WebLab is a system for managing laboratory that facilitates the control tests, customers and delivering results. The application developed in this paper uses a service oriented architecture and is hosted in a virtualized server.

Keywords: Cloud computing, Web Systems, Systems Laboratory.

LISTA DE FIGURAS

Figura 1: Usuários e provedores de computação em nuvem.....	14
Figura 2: Labol8 versão 98d de 2010.....	24
Figura 03: Sistema LABSYS Web.....	25
Figura 4: Estrutura do Sistema.....	26
Figura 05: Diagrama de caso de uso do módulo laboratório.....	27
Figura 06: Diagrama de classes de entidade do sistema WebLab.....	30
Figura 07: Diagrama de Atividade: solicitação de análise clínica.....	31
Figura 08: Diagrama de Sequência.....	32
Figura 9: Atividades da Implementação do Sistema.....	32
Figura 09: Listagem e Cadastro de Médicos.....	34
Figura 10: Listagem e Cadastro de Exames.....	35
Figura 11: Tratamento de erros do sistema.....	35
Figura 12: Mensagens do Sistema.....	36
Figura 13: Modelo lógico do Banco de Dados do Sistema WebLab.....	41

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous Javascript And XML
APIs	Application Programming Interface
BCE	Boundary Controll Entity
CRUD	Create, Retrieve, Update and Delete
DAO	Data Access Object
DOM	Document Object Model
FTP	File Transfer Protocol
GPL	General Public License
IaaS	Interface as a Service
JSP	Java Server Pages
JVM	Java Virtual Machine
MIT	Massachusetts Institute of Technology
PaaS	Platform as a service
POJO	Plain Old Java Objects
SaaS	Software as a service
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WebApp	Web Application
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO.....	11
2 COMPUTAÇÃO EM NUVEM.....	13
2.1 Conceito e Arquitetura.....	13
2.1.1 <i>Software como Serviço – SaaS</i>	13
2.1.2 <i>Plataforma como Serviço – PaaS</i>	15
2.1.3 <i>Infra-estrutura como Serviço – IaaS</i>	15
2.2 Vantagens.....	15
2.3 Desvantagens.....	16
2.4 Plataformas.....	17
2.4.1 <i>Google App Engine</i>	17
2.4.2 <i>Emc</i>	17
2.4.3 <i>NetApp</i>	18
2.4.4 <i>Microsoft Azure</i>	18
2.4.5 <i>Amazon</i>	18
2.4.6 <i>Cloud Computing LocaWeb</i>	19
2.4.7 <i>A WebHost</i>	19
3 TECNOLOGIAS UTILIZADAS NO DESENVOLVIMENTO.....	20
3.1 A Linguagem Java.....	20
3.2 Java Server Pages.....	20
3.3 JavaScript.....	21
3.4 AJAX.....	21
3.5 DOM.....	21
3.6 JQuery.....	22
3.7 VRaptor.....	22
3.8 Hibernate.....	22
3.9 PostgreSQL.....	23
4 APLICAÇÕES SIMILARES.....	24
4.1 Labol 8.....	24
4.2 LABSYSWEB.....	25

5 ESTUDO DE CASO.....	26
5.1 Análise de Requisitos.....	26
5.1.1 Estrutura do Sistema.....	26
5.1.2 Diagrama de Casos de Uso.....	27
5.2 Projeto do Sistema.....	29
5.2.1 Modelo Estrutural	29
5.2.2 Modelagem Comportamental.....	30
5.3 Implementação.....	33
5.3.1 Programação das Classes de Entidade do Sistema.....	33
5.3.2 Mapeamento Objeto Relacional com Hibernate Annotations.....	33
5.3.3 Geração do Modelo lógico com Hibernate.....	33
5.3.4 Criação do DAO Genérico	34
5.3.5 Criação da Camada de Negócios.....	34
5.3.6 Interface Gráfica.....	34
6 CONCLUSÃO E TRABALHOS FUTUROS.....	37
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	38

1 INTRODUÇÃO

Nos primeiros anos da Internet, Web sites não passavam de conjuntos de documentos virtuais interligados. Tudo começou com o surgimento da ARPANET, esta rede permitia que professores de diversos institutos trabalhassem juntos e compartilhassem dados, para fazer parte era necessário ter um contrato de pesquisa com o departamento de defesa e este privilégio eram poucos que tinham.

Logo após, a NFS (National Science Foundation) pensou em uma sucessora para a ARPANET que seria de livre acesso a todos os grupos de pesquisa universitários, surgiu então a NFSNET. O número de hosts conectados a ARPANET se tornou exponencial depois que foi adotada a pilha de protocolos TCP/IP logo após a interconexão com a NFSNET (Tanenbaum, 2006), com este crescimento surge então a Internet.

Desde o lançamento do primeiro navegador de linha de comando houve uma imensa revolução com relação à Web e ao desenvolvimento de software, surgiram novas tecnologias no mercado que permitiram o aumento da capacidade de computação tanto do lado cliente como do lado servidor, adicionando assim uma dose ainda maior de complexidade às aplicações Web e configurando grandes problemas de gerenciamento no seu desenvolvimento.

Neste contexto, percebemos que novos paradigmas de software, processos e aplicações vêm surgindo frequentemente, além disso, tais tecnologias se tornam cada vez mais importantes no dia a dia de usuários comuns até empresas de grande porte. Hoje as WebApps se tornaram sistemas muito sofisticados ligadas a banco de dados, aplicações corporativas e com milhares de usuários das diversas categorias que utilizam tal tecnologia.

Devido uma nova tendência chamada computação em nuvem os aplicativos e arquivos não precisam ser armazenados localmente, todo o conteúdo do usuário fica armazenado nas nuvens. Segundo Mirzarei (2008) a nuvem é uma metáfora para Internet, é uma abstração para uma infra-estrutura complexa onde um conjunto de servidores está interligado com o objetivo de efetuar todo o armazenamento e computação necessária, uma evolução no uso da rede de computadores.

A Computação em Nuvem traz benefícios tanto para desenvolvedores quanto para usuários, além disso, os processos e plataformas de desenvolvimento de aplicações nas nuvens foram pouco explorados. Este trabalho teve como objetivo

principal analisar as plataformas de desenvolvimento de aplicações nas nuvens e desenvolver um caso de uso de uma aplicação para gerenciamento de laboratório de análises clínicas.

Muitos laboratórios possuem diversos postos de coleta espalhados por bairros e até cidades diferentes. Através da solução apresentada aqui – O Weblab, os laboratórios poderão enviar resultados rapidamente para diversas localidades de difícil acesso, contribuindo com rapidez e agilidade nos diagnósticos médicos. O WebLab serviu como caso de uso para desenvolvimento de sistemas nas nuvens utilizando arquitetura orientada a serviços com aplicação de técnicas de engenharia de software para desenvolvimento de sistemas nas nuvens.

A metodologia utilizada no desenvolvimento deste trabalho foi a pesquisa-ação. A pesquisa-ação caracteriza-se pela definição de uma ação a ser desenvolvida, identificação dos seus agentes, objetivos e obstáculos e experimentação em situação real (THIOLLENT, 1986).

O restante deste trabalho está organizado da seguinte maneira: o capítulo 2 apresenta o conceito de computação em nuvem e tecnologias relacionadas, bem como as vantagens e desvantagens de sua utilização. O capítulo 3 apresenta as tecnologias utilizadas no desenvolvimento deste trabalho. O capítulo 4 apresenta aplicações similares a solução aqui apresentada e o capítulo 5 apresenta o desenvolvimento do caso de uso.

2 COMPUTAÇÃO EM NUVEM

Neste capítulo veremos o conceito de computação em nuvem e tecnologias relacionadas, bem como as vantagens e desvantagens de sua utilização e algumas das diversas plataformas oferecidas pelas empresas que provêm a nuvem.

2.1 Conceito e arquitetura

A computação em nuvem é uma das mais importantes evoluções tecnológicas que ainda estão amadurecendo, em breve a mesma se estabelecerá e fará parte de vida de cada um de nós. A nuvem é um conceito, uma metáfora para Internet, uma abstração para uma infra-estrutura complexa (MIRZAREI, 2008) formada por um conjunto de servidores interligados com o objetivo de efetuar todo o armazenamento e computação que o usuário necessita, restando o lado cliente das aplicações somente a visualização. Todas as funções de um sistema nas nuvens são vistas como serviços de software independentes e autocontidos (MACKENZIE et. al., 2006).

O paradigma da computação em nuvem é dividido em tipos de serviços, de modo que diversas interfaces de programação de aplicações (APIs) podem ser executadas, entretanto, por se tratar de um paradigma existem muitas contradições nas definições (MILLER, 2008). Vejamos entre os serviços os principais:

- SaaS: Software como Serviço;
- IaaS: Infra-estrutura como Serviço;
- PaaS: Plataforma como Serviço;

2.1.1 *Software Como Serviço - SaaS*

Software como serviço - SaaS é uma aplicação em um servidor remoto que permite acesso pela Internet. Desta forma os sistemas nas nuvens nada mais são do que SaaS onde o cliente paga apenas o que ele usa (Velte, 2010) . Vejamos como os serviços podem ser utilizados:



Figura 1: Usuários e provedores de computação em nuvem, ARMBRUST et. Al., 2009.

Na figura 1 vemos os serviços que compõem a nuvem, composta pelo provedor da nuvem (IaaS), a plataforma (PaaS) e o serviço (SaaS). O usuário utiliza uma aplicação Web e requisita do provedor de serviço que é a empresa que disponibilizará a infra-estrutura necessária. Um serviço é um componente de software com as seguintes propriedades (HEWITT, 2009):

- Está definido por uma interface independente de plataforma.
- Está disponível na rede.
- As operações definidas na interface levam em conta funções de negócio operando em objetos de negócio.
- Podem ser arquitetadas interfaces que podem ser estendidas com efeito de tempo real.

As vantagens de software como serviço (SaaS) para usuários finais e provedores de serviço são bem compreendidas. Os provedores de aplicação fazem a instalação do software em sua infra-estrutura e os clientes permitidos têm acesso a tais serviços de acordo com suas necessidades específicas e o que é disponibilizado pelo provedor gratuitamente ou mediante pagamento. Isto simplifica muito a manutenção, aumenta a escalabilidade e usuários podem ter acesso às aplicações a qualquer hora e de qualquer lugar (ARMBRUST et. Al., 2009). A computação em nuvem não muda estes argumentos, mas dá aos provedores de aplicação a escolha de negociar o produto deles como SaaS. Além disso, do lado cliente não é necessária a aquisição de uma grande e cara estrutura, até computadores com poder de computação pequeno podem ser utilizados como clientes (NOGUEIRA; PEZZI, 2009).

2.1.2 Plataforma Como Serviço – PaaS

A plataforma como serviço é disponibilização de um ambiente de desenvolvimento com recursos de codificação, debug, compilação e testes (IMASTERS, 2010). Neste conceito podemos tomar como exemplo o Google App Engine¹.

2.1.3 Infra-estrutura Como Serviço - IaaS

A infra-estrutura como serviço é a disponibilização de máquinas virtuais para o cliente, com recursos de processamento, armazenamento de dados, servidores e componentes de rede (SUN, 2009). Podemos adquirir um servidor virtualizado para disponibilizar aplicações, desta forma os dados dos usuários ficam armazenados na nuvem e poderão ser acessados a qualquer hora de lugares ou até dispositivos diferentes. Questões como poder de processamento, memória virtual e largura de banda podem ser negociadas com o provedor.

2.2 Vantagens

Geralmente novas tecnologias trazem consigo grandes discussões ao seu entorno, com a computação em nuvem não seria diferente, quando falamos nela podemos perceber claramente os benefícios evidentes que esta nova tecnologia nos trará e em contrapartida as grandes questões e críticas que surgiram sobre o tema. No momento vejamos alguns benefícios:

- Flexibilidade: com a idéia de SaaS temos a flexibilidade de uso de tais aplicações, uma vez que é possível fazer uso desde computadores comuns até sistemas embarcados e celulares.
- Escalabilidade: com computação em nuvem é possível que organizações possam processar grandes quantidades de dados sem requerer servidores pesados na própria instituição. Seria mais fácil e barato alugar o serviço somente durante o tempo em que se necessita processar, que adquirir e manter um grande mainframe em sua estrutura, semelhante a isto muitas organizações podem fazer uso de computação poderosa através da rede sem muito esforço e trabalhar com muitos clientes a um custo baixo.

¹ Google App Engine: <http://code.google.com/intl/pt-BR/appengine/>

- Portabilidade: O ideal é que as pessoas possam usar os serviços, independente de sistema operacional e plataforma de hardware, a computação nas nuvens poderá suportar o uso de clientes magros¹ (TANENBAUM; STEEN, 2007) e incorporar o uso de dispositivos móveis, além de hardware específico que porventura venha a surgir, para isto, basta que eles estejam conectados à Internet.

2.3 Desvantagens

Em contradição aos benefícios temos questões como segurança e posse dos dados, ética e utilização. Vejamos algumas desvantagens:

- Posse dos dados: aplicativos e até mesmo informações confidenciais do cliente estarão em posse dos provedores da nuvem, existem diversos questionamentos a respeito do que os eles podem fazer com estas informações. As empresas privadas teriam acesso de forma privilegiada a perfis de usuários podendo redirecionar propagandas e obter lucro.
- Segurança: é possível que o sistema seja atacado por invasores e informações confidenciais caiam em mãos erradas. Órgãos e usuários dos serviços que não se convencerem da segurança podem negociar condições adicionais para o armazenamento e recuperação de suas informações.
- Desempenho: é possível que determinadas aplicações fiquem lentas por estarem muito distantes do local onde o usuário está acessando, desta forma o tempo de resposta das requisições poderia diminuir o desempenho, uma vez que o meio de comunicação é uma rede de computadores que está mais propícia a erros e lentidão que um barramento físico de uma máquina local.

¹Cliente magro é um computador cliente em uma rede cliente-servidor de duas camadas o qual tem poucos ou nenhum aplicativo instalado, de modo que depende de um servidor central para o processamento de atividades

2.4 Plataformas

Uma plataforma para aplicações nas nuvens é a forma como um ambiente é disponibilizado, ou seja, quais tecnologias, padrões e aplicações que devem ser utilizadas pelos desenvolvedores para prover os serviços aos clientes. Seria uma surpresa falar em uma nova tecnologia sem que os grandes nomes do mundo da computação não estivessem incluídos, a seguir veremos algumas das plataformas de desenvolvimento de aplicações nas nuvens.

2.4.1 Google App Engine

O Google App Engine¹(PaaS) permite que executemos nossos serviços na infra-estrutura da Google, ele suporta aplicações escritas em várias linguagens de programação (GOOGLE APP ENGINE, 2010). Com a plataforma do Google App Engine, podemos criar aplicativos usando tecnologias padrão Java, incluindo a Java Virtual Machine², Java Servlets³, e a linguagem de programação Java ou qualquer outra linguagem usando um interpretador JVM baseada ou compilador, como JavaScript⁴ e Ruby⁵. App Engine também dispõe de um ambiente de execução dedicado Python⁶, que inclui um interpretador e a biblioteca padrão do Python.

Não é necessário nenhum investimento inicial para começar a desenvolver aplicações utilizando o App Engine, todos os aplicativos podem usar até 500 MB de armazenamento e largura de banda suficiente para suportar uma aplicação eficiente que serve até 5 milhões de visualizações de página por mês. A partir daí é necessário pagamento para a utilização dos recursos.

2.4.2 Emc

A EMC Corporation (EMC, 2010) é uma das grandes empresas em produtos, serviços e soluções de informação, armazenamento e gerenciamento. Eles dão suporte à computação em nuvem e virtualização. Segundo a empresa, a sua Symmetrix V-Max system, é o primeiro sistema de gestão de apoio de alta qualidade centros de dados virtuais. Este sistema permite que centros de dados múltiplos

¹ <http://code.google.com/intl/pt-BR/appengine/>

² JVM(Java Virtual Machine): http://www.java.com/pt_BR/download/faq/whatis_java.xml

³ Java Servlets: <http://www.oracle.com/technetwork/java/index-jsp-135475.html>

⁴ JavaScript: <http://www.w3schools.com/js/default.asp>

⁵ Ruby: <http://www.rubyonrails.pro.br/>

⁶ Python: <http://www.python.org/>

sejam executados como se fossem únicos, tornando a sua gestão muito mais fácil e mais eficiente. Ao contrário do Goglee App Engine, a V-Max utiliza somente virtualização (IaaS) de servidores ao invés de PaaS e SaaS.

2.4.3 NetApp

A NetApp oferece centros de dados e serviços de armazenamento, bem como virtualização (NETAPP, 2010). Seus produtos incluem uma plataforma de sistema operacional, armazenamento seguro, gerenciamento e segurança de software.

2.4.4 Microsoft Azure

A Plataforma da Microsoft para computação em nuvem é a Windows Azure (WINDOWS AZURE, 2010). O Windows Azure é um sistema operacional baseado na nuvem que atua como ambiente de desenvolvimento, hospedagem de serviços e gerenciamento. Ele fornece aos desenvolvedores, computação sob demanda e armazenamento para hospedar, escalar e gerenciar aplicações Web, nos centros de dados da Microsoft.

A Azure é uma plataforma flexível que dá suporte a várias linguagens e se integra facilmente com um ambiente local já existente. Para criar aplicações e serviços, os desenvolvedores podem usar o conhecimento já adquirido no Microsoft Visual Studio. Além disso, são suportados protocolos e padrões populares, como o protocolo simples de acesso a objetos - SOAP¹, à Transferência de Estado Representacional - REST², eXtensible Markup Language - XML³ e PHP⁴. Ela combina recursos de desenvolvimento baseados na nuvem com serviços de infraestrutura de rede, computação e armazenamento.

2.4.5 Amazon

O Amazon Elastic Compute Cloud (AMAZON, 2010) é um serviço Web que oferece capacidade computacional expansível na nuvem (IaaS), e é projetado para tornar mais escaláveis as WebApps, ele fornece uma interface Web simples que permite obter e configurar o ambiente com facilidade, além do controle de recursos.

2.4.6 Cloud Computing LocaWeb

¹ SOAP: <http://www.w3.org/TR/soap/>

² REST: <http://www.w3.org/Submission/SA-REST/>

³ XML: <http://www.w3.org/XML/>

⁴ <http://www.w3schools.com/php/default.asp>

O Cloud Computing LocaWeb (LOCAWEB, 2010) é o ambiente da LocaWeb, uma das empresas que oferecem em computação em nuvem no Brasil. Formado por uma nuvem de grande poder de processamento e armazenamento, garante segundo a empresa, escalabilidade, processamento isolado para cada cliente e um desempenho muito bom. O cliente pode contratar o serviço, e a medida da sua necessidade fazer um upgrade sem nenhuma dificuldade e nem necessidade de se preocupar com suas aplicações.

2.4.7 A WebHost

A Webhost (WebHOST, 2010) é uma empresa brasileira que opera no mercado de Internet desde 1997. Atualmente, oferece serviços que abrange hospedagens de sites, registro de domínios e servidores virtualizados (IaaS). A empresa disponibiliza dois planos:

- Linha Linux: para hospedar aplicações desenvolvidas com linguagem PHP, suporte aos Bancos MySql¹ e Postgres², serviço FTP³ e controle de versão com Subversion.
- Linha Java: possui todos os recursos da linha Linux e tem suporte a aplicações desenvolvidas em Java. Utiliza de virtualização de servidor com máquina virtual Java e o Webcontainer Apache Tomcat⁴.

Para o desenvolvimento do estudo de caso, foram utilizadas tecnologias compatíveis com a estrutura disponibilizada pela WebHost. Detalhes serão apresentados nos próximos capítulos.

¹MySql: <http://www.mysql.com/>

²Postgres:[http:// www.postgresql.org/](http://www.postgresql.org/)

³ FTP: File Transfer Protocol – Protocolo de transferência de arquivos

⁴ Tomcat: <http://tomcat.apache.org/>

3 TECNOLOGIAS UTILIZADAS NO DESENVOLVIMENTO

Neste capítulo veremos as tecnologias utilizadas no desenvolvimento do caso de uso proposto.

3.1 A linguagem Java

A linguagem de programação Java¹ foi desenvolvida na década de 90 pela Sun Microsystems². Diferente das outras linguagens que são compiladas, Java é considerada um linguagem híbrida, ou seja, passa por um processo de compilação e interpretação (SEBESTA, 2003). O processo de compilação de Java ao invés de gerar diretamente código de máquina, gera os chamados bytecodes (forma intermediária de código) que posteriormente são interpretados pela máquina virtual (DEITEL, P. J; DEITEL, H. M., 2005), assim a portabilidade é uma das características mais marcantes desta linguagem.

Características como sintaxe fácil, orientação a objetos, segurança, vasto conjunto de bibliotecas, coleta de lixo automática e suporte a criação de aplicações distribuídas e multitarefa, fazem com que Java seja uma das linguagens mais utilizadas na atualidade, razão pela qual a escolhemos para desenvolvimento do projeto.

3.2 Java Server Pages

Com a sofisticação dos serviços disponibilizados via Web, surgiu à necessidade de disponibilizar informações com natureza dinâmica (lista de preços atualizados, compras on-line, etc). Java Server Pages (JSP) consiste numa linguagem de script baseada em Java para criação de sites com conteúdos dinâmicos. Pelo fato de termos utilizado Java para camada de negócio e dos diversos benefícios do JSP, escolhemos esta tecnologia para camada de visualização deste projeto, entretanto, poderíamos utilizar qualquer tecnologia que permite o uso de Asynchronous Javascript And XML³ (AJAX).

¹Java: <http://www.java.com/>

² Sun: <http://www.sun.com>

³ AJAX: <http://www.w3schools.com/ajax/default.asp>

3.3 JavaScript

JavaScript¹ é uma linguagem de programação criada pela Netscape na década de 90 com o objetivo de tornar a interação do usuário com as páginas Web mais dinâmicas. Após o sucesso, a linguagem recebeu uma grande colaboração da Sun. Atualmente JavaScript é utilizada na grande maioria das aplicações Web, pelo fato de conferir aos usuários uma experiência mais agradável em relação a sua interação com a página, esta linguagem foi escolhida para ser utilizada neste projeto.

3.4 AJAX (Asynchronous Javascript And XML)

AJAX² é a tecnologia do uso sistemático de JavaScript e XML implementada em navegadores para tornar páginas Web mais iterativas com o usuário utilizando de requisições assíncronas. Grandes aplicações Web como Gmail, Orkut e Google Maps utilizam esta tecnologia. Devido benefícios como aumento na velocidade do site, melhor experiência do usuário e flexibilidade para escolha da linguagem server-side³ esta tecnologia foi utilizada neste projeto.

3.5 DOM (Document Object Model)

O Document Object Model⁴ (DOM) é uma especificação da World Wide Web Consortium (W3C), independente de plataforma e linguagem, onde pode-se dinamicamente alterar e editar a estrutura, conteúdo e estilo de um documento eletrônico, permitindo que o documento seja mais tarde processado e os resultados desse processamento, incorporados de volta no próprio documento. Esta tecnologia foi utilizada no projeto por oferecer uma maneira padrão de se acessar os elementos de um documento, além de trabalhar com cada um desses elementos separadamente, e criar páginas altamente dinâmicas, fornecendo assim uma melhor experiência ao usuário.

¹ JavaScript: <http://www.w3schools.com/js/>

² AJAX: <http://www.w3schools.com/ajax/default.asp> acessado em 12/11/2010

³ Server-side: termo usado para designar operações que, em um contexto cliente-servidor, são feitas no servidor.

⁴ Document Object Model: <http://www.w3.org/DOM/>

3.6 JQuery

Criada por Jonh Resig e disponibilizada como software livre pelo Massachusetts Institute of Technology (MIT) e GPL, JQuery¹ vem sendo cada vez mais utilizada em todo o mundo (SILVA, 2010). Por se tratar de uma biblioteca JavaScript que reduz significativamente a quantidade de linhas de código e trazer uma API de desenvolvimento de interfaces gráficas muito rica de fácil uso, esta tecnologia foi escolhida para ser utilizada neste projeto. Esta biblioteca foi escolhida para ser utilizada neste trabalho por tornar a experiência de uso das WebApps muito mais agradável, simplificar o desenvolvimento e deixar as aplicações crossbrowser².

3.7 Vraptor

O Vraptor (VRAPTOR, 2010) é um framework Model-View-Controller³ (MENDES, 2002) para desenvolvimento Java que teve sua origem na Universidade de São Paulo, iniciativa brasileira livre para desenvolvimento ágil de aplicações Web. Devido características como alta produtividade, testabilidade, flexibilidade, melhores práticas de desenvolvimento e simples aprendizado, esta tecnologia foi escolhida para ser utilizada neste projeto.

3.8 Hibernate

O Hibernate⁴ é um framework para persistência e recuperação de dados escrito em Java. Utiliza mapeamento objeto relacional gerando todos os comandados SQL (Structured Query Language), liberando o desenvolvedor da conversão do retorno da consulta em objetos. Por se tratar de um software livre de código aberto que agiliza o desenvolvimento, de fácil aprendizagem, portátil e que reduz muito a quantidade de linhas de código, esta tecnologia foi escolhida para ser utilizada neste projeto.

¹ JQuery: <http://www.jquery.com/>

² Crossbrowser: Independente de browser

³ Model-view-controller (MVC): padrão de arquitetura de software que visa a separar a lógica de negócio da lógica de apresentação, permitindo o desenvolvimento, teste e manutenção isolado de ambos.

⁴ Hibernate: <http://www.hobernate.org>

3.9 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional que pode ser utilizado para diversas aplicações. Com robustez e segurança este software livre tem se tornado uma das mais utilizadas soluções de armazenamento de pequenas até grandes aplicações (MILANI, 2008). Devido características como suporte a consultas complexas, integridade transacional, controle de concorrência e robustez, este banco de foi escolhido para ser utilizado neste projeto.

A seguir apresentaremos aplicações similares ao WebLab.

4 APLICAÇÕES SIMILARES

Neste capítulo apresentaremos uma análise de algumas soluções similares ao WebLab. Com a evolução da Web e das tecnologias ligadas ao desenvolvimento de WebApps, muitas aplicações encontradas no mercado mostram-se insuficiente para atender as demandas dos laboratórios, uma vez que muitos deles possuem vários postos de coleta espalhados por diversas localidades. Serão mostradas algumas funcionalidades e limitações dos sistemas analisados.

4.1 LABOL 8

O LABOL8 é um sistema de gerenciamento de laboratórios desenvolvido em COBOL pela empresa carioca Marsoft. A Figura 2 apresenta a interface gráfica deste sistema.



Figura 2: Labol8 versão 98d de 2010

A solução analisada possui funcionalidades como cadastro de pacientes, exames, planilhas, lançamento e impressão de resultado. O Labol mostra-se insuficiente para atender as necessidades dos laboratórios. Este software não oferece vantagens proporcionadas pelo uso da Internet e dos bancos de dados relacionais, além de não funcionar em alguns sistemas operacionais e ser desenvolvido utilizando tecnologia que está caindo em desuso.

4.2 LABSYS WEB

O LABSYS WEB foi criado pela empresa MEDSYSTEM¹. A Figura 3 apresenta a interface gráfica desta solução.



Figura 3: Sistema LABSYS Web

Este aplicativo foi desenvolvido em C++ e integra outra aplicação específica para fazer up-load de resultados e disponibilizá-los através da Internet. Possui funcionalidades como controle de estoque, cadastro de pacientes, exames, emissão de resultados e emissão de relatórios. Esta solução é mais moderna e completa se comparada com a apresentada anteriormente, entretanto, não oferece as vantagens proporcionadas por uma aplicação Web.

O capítulo seguinte apresenta uma solução que busca resolver as limitações apresentadas pelos sistemas analisados.

¹ <http://www.labsysweb.com.br/>

5. ESTUDO DE CASO

O WebLab é uma solução nas nuvens para gerenciamento de laboratório de análises clínicas que busca facilitar o controle de exames, clientes e a entrega de resultados. O sistema incorpora todos os benefícios das WebApps bem como da computação em nuvem. Neste capítulo apresentamos os requisitos principais da aplicação WebLab e descrevemos o desenvolvimento do sistema.

5.1 Análise de Requisitos

Apresentamos agora os requisitos principais do WebLab. O sistema foi modelado segundo a notação UML utilizando-se os seguintes diagramas: diagrama de caso de uso, diagrama de classe, diagrama de sequência e diagrama de atividades.

5.1.1 Estrutura do Sistema

O WebLab possui vários módulos, os mesmos serão desenvolvidos de forma incremental e em sequência. A Figura 04 abaixo mostra todos os pacotes que farão parte do sistema assim que os trabalhos futuros a este estiverem concluídos.

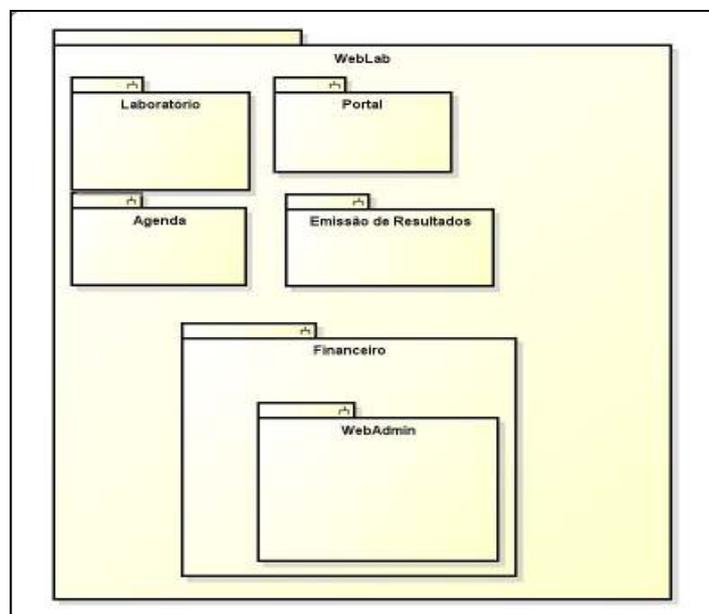


Figura 4: Estrutura do Sistema

O pacote Laboratório conterá funcionalidades referentes às atividades necessárias para emissão dos resultados dos exames.

O pacote agenda será utilizado para efetuar agendamentos e auxiliar os funcionários do laboratório no exercício das suas atividades.

O pacote Portal será um subsistema que se integrará ao portal do laboratório somente com a finalidade de emitir resultados através da Internet, esta aplicação deverá conter uma segurança mais apurada, sendo utilizada criptografia de senhas e protocolos de segurança.

O pacote Emissão de Resultados trará as funcionalidades referentes à emissão dos resultados lançados bem como relatórios dinâmicos necessários.

Destacamos que neste trabalho foram desenvolvidas apenas as funcionalidades previstas no pacote laboratório, por se tratar de um sistema muito complexo para ser totalmente desenvolvido em um semestre.

5.1.2 Diagrama de Casos de Uso

O diagrama de caso de uso é a representação das características externamente observáveis do sistema (BEZERRA, 2007). A figura 5 abaixo mostra o diagrama de caso de uso do sistema WebLab.

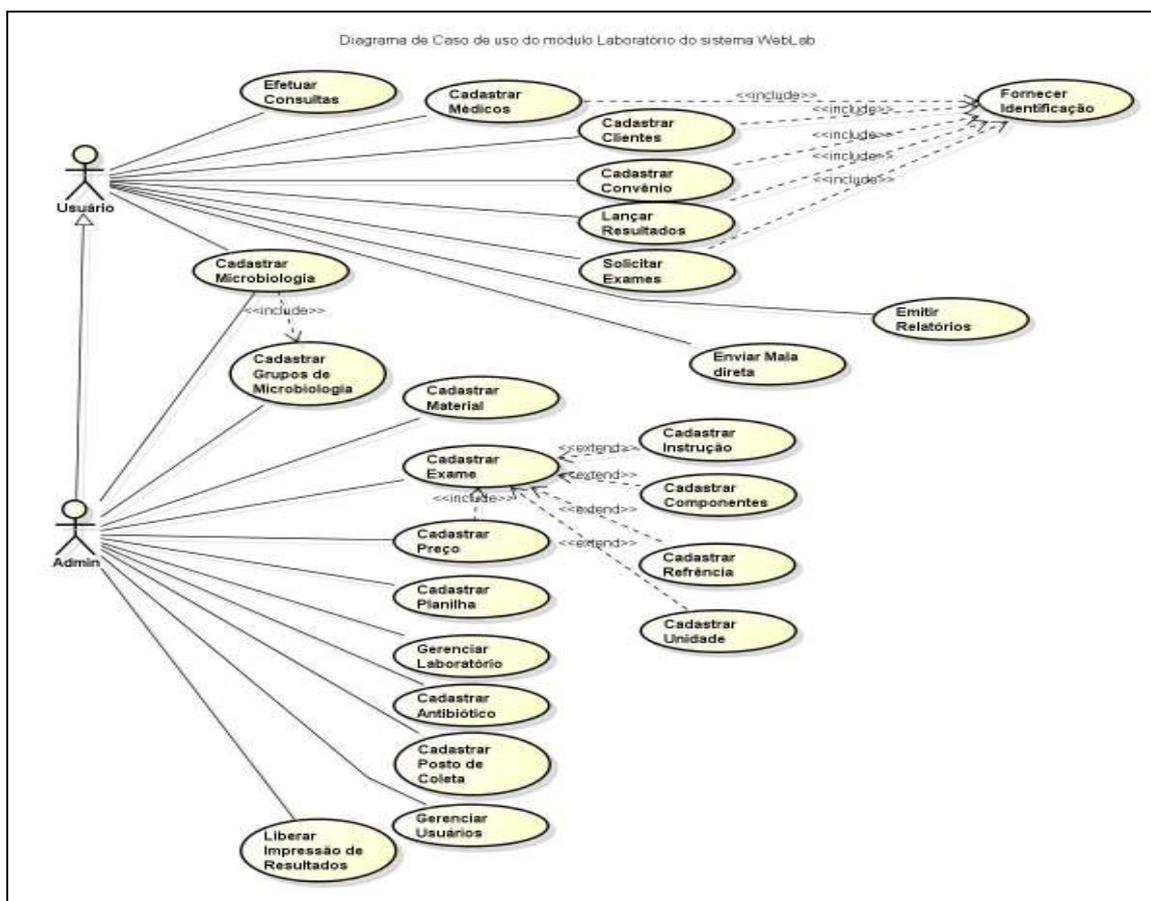


Figura 5: Diagrama de Caso de Uso do Módulo Laboratório

A seguir apresentamos a descrição textual dos casos de uso mais complexos do sistema.

Caso de Uso Fornecer Identificação

Neste caso de uso, o ator Usuário deverá fornecer identificação, serão solicitados seu login e uma senha. Estes dados serão utilizados pela rotina de auditoria que vai armazenar dados das requisições dos usuários.

Caso de Uso Cadastrar Médico

Neste caso de uso o ator Usuário cadastrará os médicos que solicitaram a realização da análise clínica, estas informações serão necessárias ao caso uso Solicitar Exames bem como aos relatórios que serão gerados pelo sistema.

Caso de Uso Cadastrar Clientes

Neste caso de uso o ator Usuário cadastrará o cliente que deseja realizar a consulta.

Caso de Uso Cadastrar Convênio

Neste Caso de uso o ator Usuário deverá cadastrar os convênios feitos junto a planos de saúde e o laboratório. Caso a solicitação do exame seja através de um convênio específico deverá ser impresso nas fichas fornecidas por cada plano todos os dados do exame solicitados pelos mesmos.

Caso de Uso Solicitar exames

Neste caso de uso o ator Usuário solicitará os exames necessários aos clientes.

Caso de Uso Lançar Resultados

Neste caso de uso o ator Usuário lançará os resultados dos exames solicitados.

Caso de Uso Cadastrar exame

Cada exame possui vários componentes, cada componente poderá ser de vários tipos, além de possuir referências específicas de cada faixa etária que poderá ser medida em dias, meses e anos. Existem Tipos de componente que poderão ser multivalorados ou um subconjunto de microbiologia. Mais detalhes poderão ser observados no diagrama de classes ou no modelo entidade relacional, este caso de uso poderá ser acessado somente pelo ator Admin.

Caso de Uso Efetuar Consultas

Cada cadastro no sistema possui uma listagem dos itens já cadastrados, através deste caso de uso o ator Usuário pode fazer buscas e consultas. Como

alguns usuários não podem ter acesso a todas as informações, é necessário fazer este controle de acordo com cada grupo de usuários.

Caso de Uso Gerenciar Laboratório

Neste caso de uso o ator Admin gerenciará os dados dos laboratórios, bem como cadastrar bioquímico responsáveis além dos dados referentes aos exames.

A seguir apresentaremos o projeto do sistema, bem como o modelo estrutural e a modelagem comportamental.

5.2 Projeto do Sistema

Apresentamos agora o projeto do sistema WebLab. Foram utilizados diagramas de classe, atividade e sequência.

5.2.1 Modelo Estrutural

A Figura 6 apresenta o diagrama de classes da aplicação WebLab. É possível geramos o código fonte das classes a partir do modelo do diagrama de classes utilizando ferramentas de UML como astah, entretanto, quando especificamos no modelo as operações que as classes poderão executar, a ferramenta gera somente a escritura do método, não gerando o código referente ao que este método deverá executar, sendo assim não foram especificados os métodos get e set das classes no modelo para evitar ter que programar as funcionalidades dos métodos manualmente, uma vez que eles podem ser gerados por completo pelas interfaces de desenvolvimento (IDE).

O diagrama apresenta 27 entidades e os relacionamentos entre as mesmas, foi utilizado o relacionamento de composição por ser necessário ao mapeamento objeto relacional com hibernate annotations¹.

¹ Hibernate anotations: anotações do framework hibernate para fazer mapeamento objeto relacional sem o uso de arquivos XML.

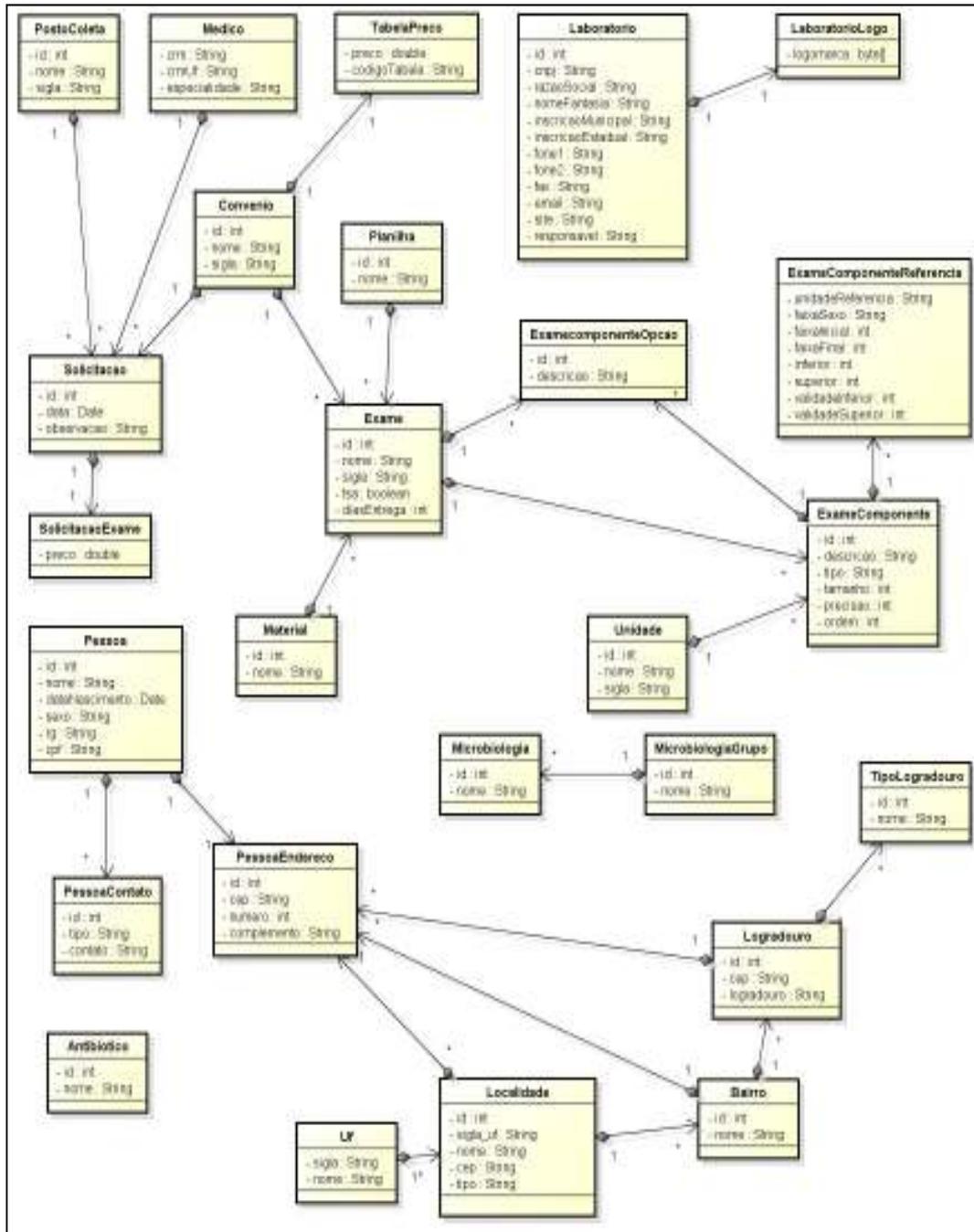


Figura 6: Diagrama de classes de entidade do sistema WebLab.

5.2.2 Modelagem Comportamental

Para modelar o comportamento do sistema foram utilizados os diagramas de atividade e sequência. A Figura 7 apresenta o processo das atividades que serão executadas em uma solicitação de análise clínica.

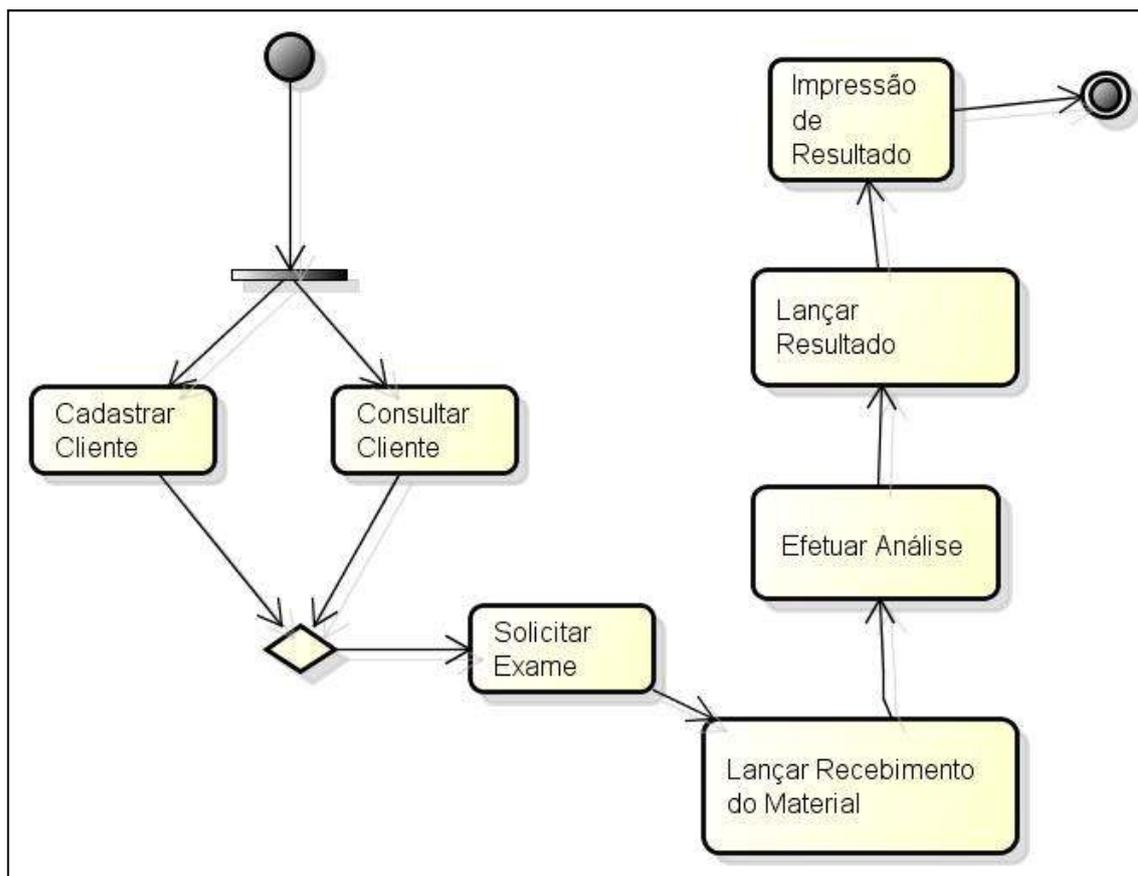


Figura 7: Diagrama de Atividade: solicitação de análise clínica

Foi modelada a atividade em que o cliente solicita uma análise clínica. Inicialmente é feito o cadastro ou consulta do cliente, logo após é feita a solicitação do exame e recolhido o material para análise. Depois de analisar o material recolhido é feito lançamento dos resultados obtidos e emitido um relatório com estes resultados.

O diagrama de sequência apresentado na Figura 8 representa a sequência de mensagens passadas entre os objetos. O diagrama foi representado segundo a categorização BoundaryControllEntity - BCE e os objetos de fronteira (Boundary) como Interface, os de controle (Controll) como Controller e os de entidade (Entity) como Model. Para Bezerra (2007) a categorização BCE serve como arcabouço que podemos tomar como ponto de partida para identificação de classes em cada caso de uso do sistema. Esta notação foi utilizada para simplificar o entendimento e a modelagem da comunicação entre os objetos.

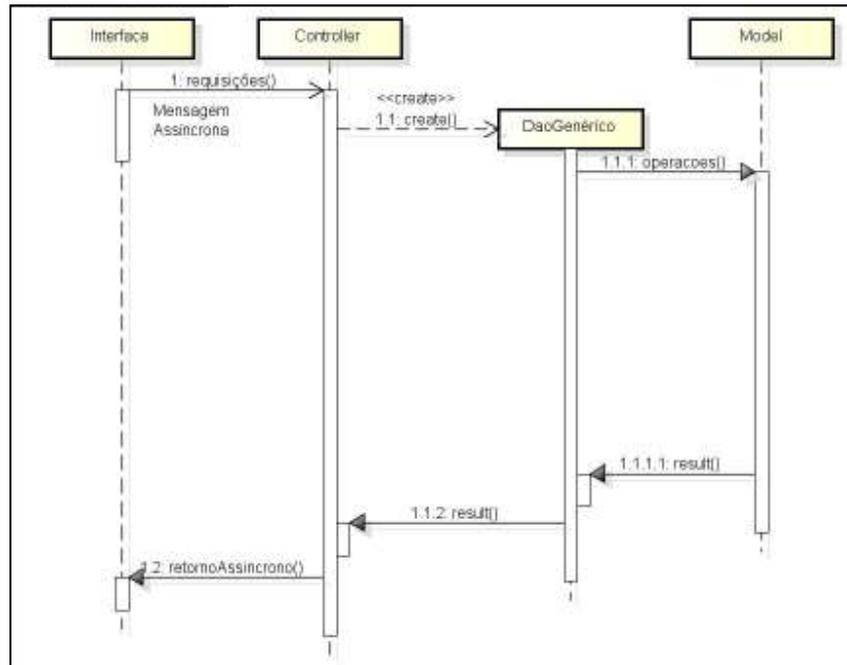


Figura 8: Diagrama de Sequência

O usuário final comunica-se com os objetos de fronteira representados pelo objeto Interface, como podemos analisar, a interface faz requisições assíncronas à camada de negócio representada pelo objeto Controller, que por sua vez trata a requisição e utiliza o objeto representado como DaoGenerico para persistir ou efetuar consultas. O DaoGenérico foi construído para não ser necessária uma classe DAO¹ para cada entidade do sistema. Com ele é possível fazer persistência e consultas complexas sem nenhuma linha de SQL, somente com orientação a objetos.

Assim que o DaoGenerico conclui sua comunicação com o banco de dados (representado pelo objeto Model) e produz uma resposta para o Controller, é gerada uma resposta que é enviada para camada de visualização para ser tratada e emitida ao usuário.

A seguir veremos detalhes da implementação do sistema.

¹ DAO: Data Access Object: Utilizado para resgatar, persistir e atualizar objetos.

5.3 Implementação

A Figura 9 apresenta as atividades que foram realizadas na implementação do módulo laboratório do sistema WebLab.



Figura 9: Atividades da Implementação do Sistema

5.3.1 Programação das Classes de Entidade do Sistema

Através da modelagem do sistema com o diagrama de classes foi possível gerar o código fonte das classes de entidade do sistema. A ferramenta astah UML, gera as classes somente com atributos e escrituras dos métodos especificados no modelo, por isso foi feita uma revisão em todo o código gerado e adicionados os métodos get e set pela IDE Netbeans.

5.3.2 Mapeamento Objeto Relacional Com Hibernate-annotations

Após a programação das entidades ter sido concluída, foi necessário fazer o mapeamento objeto relacional utilizando Hibernate-annotations. O mapeamento objeto relacional funciona com a transformação dos dados de um objeto em uma linha de uma tabela do banco de dados, ou com a transformação de uma linha da tabela em um objeto da aplicação.

Em outras palavras, é possível transformar classes em tabelas no banco de dados, ou tabelas no banco de dados em classes. Como as entidades já existiam, foi possível gerar as relações do banco de dados.

5.3.3 Geração do Modelo Lógico com Hibernate

Este modelo representa a concepção da estrutura necessária ao armazenamento dos dados necessários à aplicação, o mesmo foi implementado após análise rigorosa das necessidades dos futuros usuários. Destacamos que o banco foi gerado pelo framework Hibernate a partir das classes anotadas.

5.3.4 Criação do DAO Genérico

Muitos projetos utilizam uma classe DAO para cada entidade do sistema, esta forma de trabalhar polui o código das aplicações, torna a manutenção e programação mais trabalhosa e diminui a produtividade dos programadores.

Neste trabalho foi construído um DAO genérico utilizando o framework Hibernate e um conceito do Java chamado reflection¹. Este DAO permite persistência e consultas complexas utilizando somente orientação a objetos.

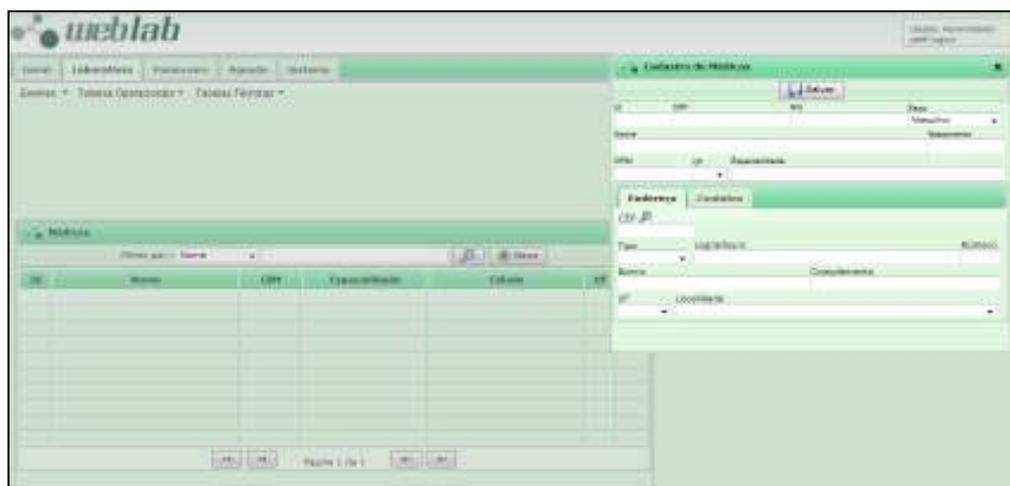
5.3.5 Criação da Camada de Negócio

Após a criação das classes, do banco de dados e do DAO genérico foi criada a camada de negócio da aplicação, cada entidade tem restrições e permissões que deverão ser implementadas na camada de negócio. Nesta camada estão todos os serviços que poderão ser requisitados pela camada de aplicação que pode estar em qualquer tipo de dispositivo, desde que este esteja conectado à Internet.

5.3.6 Interface Gráfica

Baseados em regras de interação humano-computador, a interface gráfica do sistema foi desenvolvida levando em considerações diretrizes para proporcionar uma melhor experiência ao usuário, como utilização cores, agrupamentos, máscaras de campos e tecnologias que deixam as páginas mais amigáveis.

Na figura 9 apresentamos a interface de listagem e cadastro de médicos. Na listagem podemos efetuar buscas utilizando vários critérios, foi criado um sistema de paginação que agrupa as consultas trazendo para visualização poucos itens por vez.



¹ Reflection: Permite um programa Java examinar ou fazer uma introspecção nele mesmo, ou seja, olhar e examinar suas propriedades e estrutura.

Figura 10: Listagem e cadastro de médicos

Nos cadastros foram utilizadas abas para que o usuário possa visualizar melhor os campos que vai preencher e máscara nos campos para auxiliá-lo no preenchimento. A figura 10 apresenta listagem e cadastro de exames.

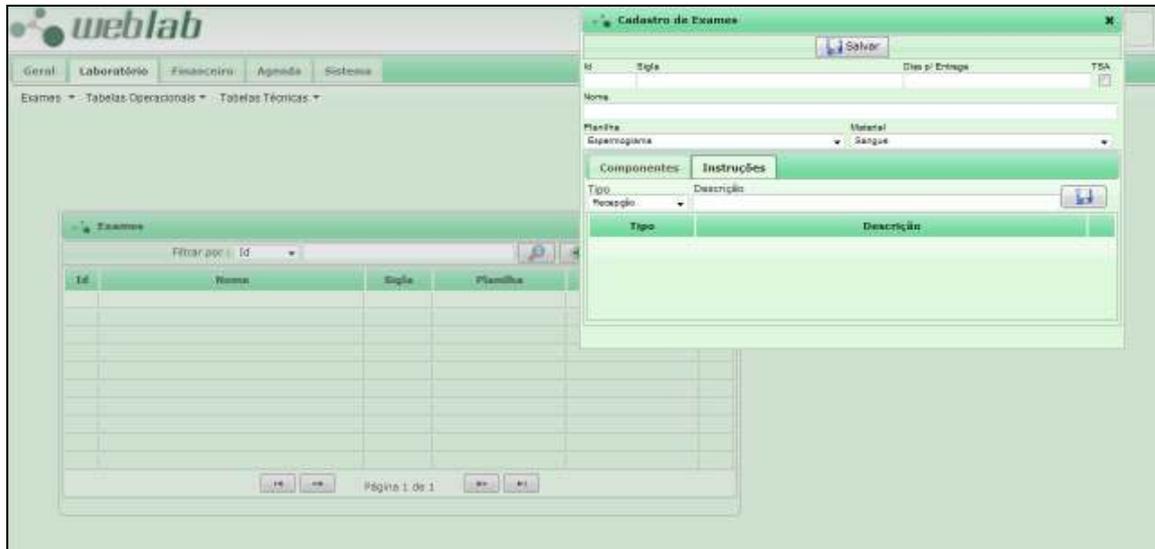


Figura 11: Listagem e cadastro de Exames

Através desta interface podemos efetuar o cadastro dos exames que o laboratório efetuará. Atributos ligados aos exames foram agrupados em abas e estão na mesma tela de cadastro de exames.

Existem alguns tipos de erros mais comuns nesta aplicação, eles podem acontecer quando o AJAX faz a requisição para o Controller, durante a validação, caso o objeto a ser inserido viole alguma restrição de integridade ou ainda quando acontece alguma falha na comunicação com o banco de dados. Estes erros foram tratados e foi criada uma lista que é retornada para visualização conforme visto na Figura 11.

The screenshot shows a web form titled 'Cadastro de Médicos'. It contains several input fields for personal and professional information. A red error message box is overlaid on the form, stating 'ERRO: Campo Nascimento é Obrigatório'. The form fields include:

- Id, CPF (589.380.935-87), RG (1450895493), Sexo (Masculino)
- Nome (Elias Adriano Nogueira da Silva), Nascimento (empty)
- CRM (432523), UF (BA), Especialidade (Geral)
- Endereço section with fields for CEP, Tipo (Logradouro), Número, Bairro, Complemento, UF (Bahia), and Localidade.

Figura 12: Tratamento de erros do sistema

Neste caso foi deixado em branco o campo Nascimento, a mensagem foi exibida devido a validação server-side de campos, uma mensagem será exibida caso aconteça outros tipos de erro.

Para que o usuário acompanhe o status da sua requisição foram colocadas mensagens de confirmação. Do mesmo modo das mensagens de erro. Figura 12 apresenta uma mensagem de confirmação no sistema.

The screenshot shows a table titled 'Antibióticos'. It has a search bar and a '+ Novo' button. The table contains two rows of data. A green confirmation message box is displayed in the center of the table, stating 'OK! Excluído com sucesso!'. The table footer shows 'Página 1 de 1' and navigation buttons.

Id	Nome
3	Antibiótico 04
1	Antibiótico 01

Figura 13: Mensagens do sistema

6 CONCLUSÃO E TRABALHO FUTUROS

A Computação em Nuvem surge como um novo paradigma da computação e traz benefícios tanto para os desenvolvedores de sistema quanto para usuários das

soluções que utilizam esta tecnologia. Este trabalho apresentou o desenvolvimento de uma aplicação que tem como objetivo o gerenciamento de laboratórios de análises clínicas. O WebLab está hospedado em um servidor virtualizado na Internet, utiliza conceitos da computação em nuvem e arquitetura orientada a serviços.

Com o desenvolvimento do trabalho, pudemos perceber a importância da Computação em Nuvem e ter a idéia da grande evolução que esta tecnologia poderá trazer para computação.

Dentre as principais dificuldades encontradas para realização do trabalho destacamos o levantamento de requisitos, compreender o mundo e as necessidades dos laboratórios foi um desafio. Consideramos que as tecnologias que foram criadas (DaoGenerico, Scripts da interface e padronização do uso das bibliotecas de desenvolvimento) foram os aspectos mais importantes do trabalho desenvolvido, uma vez que a partir delas será possível o desenvolvimento de outras aplicações que possuem as mesmas características.

Como trabalhos futuros destacamos: adicionar as funcionalidades previstas na modelagem do sistema completo e incrementos como geração de relatórios dinâmicos que colaborariam para o sistema se tornar uma importante solução na área de gerenciamento de laboratórios, possibilitando ao mesmo concorrer com outras soluções existentes no mercado.

7 REFERÊNCIAS BIBLIOGRAFICAS

AMAZON. Disponível em: <<http://www.amazon.com/>>. Acessado em: set 2010.

ARMBRUST, Michael; FOX, Armando; GRIFFITH, Rean; JOSEPH, Anthony .
;KATZ, Randy; KONWINSKI, Andy; LEE, Gunho; PATTERSON. David;
RABKIN Ariel; STOICA Ion; ZAHARIA, Matei. Above the Clouds: **A Berkeley
View of Cloud Computing**, 2009. Disponível
em:<[http://www.eecs.berkeley.edu/
Pubs/TechRpts/2009/EECS-2009-
28.html](http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html)>. Acessado em: set 2010.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**,
Elsevier, 2007.

DEITEL, Paul J.; DEITEL, Harvey M. **Java: Como Programar** - 6ª Edição – BR.
Bookman, 2005.

EMC Corporation. Disponível em: [http://www.emc.com/products/detail/hardware
/symmetrix-vmax.htm](http://www.emc.com/products/detail/hardware/symmetrix-vmax.htm)>. Acessado em: 02 set. 2010

GOOGLE APP ENGINE. Disponível em: <[http://code.google.com/intl/ptBR/appe
ngine/](http://code.google.com/intl/ptBR/appengine/)>. Acessado em: set 2010

HEWITT, Eben. **Java SOA Cookbook**. O`Reilly Media, 2009.

LOCAWeb. Disponível em: <<http://www.locaWeb.com.br>>. Acessado em: set 2010.

MACKENZIE, C. Matthew; LASKEY, Ken; McCabe, Francis; BROWN, Peter;
METZ, Rebekah; HAMILTON, Booz Allen. **Reference Model for Service
Oriented Architecture**. OASIS Standard, 12 October 2006. Disponível
em:<asis-open.org/committees/tc_home.php?wg_abbrev=soa-rm>.
Acessado em: set 2010.

MENDES, Antônio. **Arquitetura de Software – Desenvolvimento Orientado para Arquitetura**. Campus, 2002.

MILANI, André. **PostgreSQL – guia do programador**, Novatec, 2008.

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

MIRZAEI, Nariman. **Cloudcomputing**, 2008. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download> >. Acessado em: out 2010.

NETAPP. Disponível em: <<http://www.netapp.com/br/>>. Acessado em: set 2010.

NOGUEIRA, Matheus Cadori; PEZZI, Daniel da Cunha. A computação agora é nas nuvens. , 2009. Disponível em: <<http://underlinux.org/pl/attachments/f98/8369d1257953490-inwazja-cloud-computing-pdf>>. Acessado em: out 2010.

PaaS, Cloud Computing, Virtualização e o Futuro - Parte 02. In: **IMASTERS** 2010. Disponível em: <http://imasters.com.br/artigo/14228/cloud/paas_cloud_computing_virtualizacao_e_o_futuro_parte_02/> Acessado em: out 2010.

SEBESTA, Robert W. **Conceitos de linguagens de programação**, 5ª Edição- Porto Alegre: Bookman, 2003.

SILVA, Mauricio Sammy. **JQuery – A biblioteca do programador**, 2ª edição, Novatec, 2010.

SUN MICROSYSTEMS - **Introduction to Cloud Computing Architecture**, White Paper, 1st Edition, June 2009. Disponível em: http://Webobjects.cdw.com/Webobjects/media/pdf/Sun_CloudComputing.pdf. Acessado em: set 2010.

THIOLLENT, Michel. **Metodologia da pesquisa-ação**. São Paulo: Cortez: Autores Associados, 1986.

TANENBAUM, ANDREW S.; STEEN, MAARTE VAN. **Sistemas Distribuídos
Princípios e paradigmas.** Prentice-Hall, 2007.

WebHOST. Disponível em: <<http://www.Webhost.com.br>>. Acessado em: set 2010.

WINDOWS AZURE. Disponível em: <<http://msdn.microsoft.com/pt-br/windowsazure/default.aspx>>. Acessado em: set 2010.

VRAPTOR. Disponível em: <<http://vraptor.caelum.com.br/>>. Acessado em: set 2010.