Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

Sistema de Gerenciamento de Help Desk

Autor:

Victor Gadelha Boscolo

Orientador:

Prof. Antonio Claudio Gomez de Sousa, M.Sc.

Examinador:

Prof. Aloyzio de Castro Pinto Pedroza, Dr.Ing.

Examinador:

Prof. Sergio Barbosa Villas Boas, Ph.D.

DEL

Setembro de 2009

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica – Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro – RJ CEP 21949-900

DEDICATÓRIA

Dedico este trabalho ao povo brasileiro que contribuiu de forma significativa à minha formação e estada nesta Universidade. Este projeto é uma pequena forma de retribuir o investimento e confiança em mim depositados.

AGRADECIMENTO

Agradeço a todos aqueles que estiveram ao meu lado, principalmente a Deus, a minha família, namorada e amigos, e que me ajudaram a atingir esta grande meta na minha vida.

RESUMO

Este trabalho prevê a criação de uma solução sistemática para o helpdesk de empresas que não possuem a área de TI como sua área fim. Esta solução se baseia no conceito de CRM, onde o relacionamento com o cliente é vital para o desenvolvimento dos negócios.

O desenvolvimento do projeto será guiado através da metodologia MEGA

A solução criada será responsável pelo controle de chamados dos usuários do sistema para os diversos problemas existentes dentro de uma empresa. Com isso, um panorama detalhado sobre as atividades relacionadas pode ser mostrado, servindo de apoio para tomada de decisões e melhor planejamento.

A solução batizada de Sistema de Gerenciamento de Help Desk será construindo utilizando a linguagem de programação JAVA e utilizará banco de dados Mysql.

Palavras-Chave: CRM, Help Desk, Chamados, Java, MEGA.

ABSTRACT

This project involves the creation of a systematic solution to the help desk of companies that lack the IT area in order. This solution is based on the concept of CRM, where the customer relationship is vital to the development of business.

The development project will be guided through the MEGA methodology.

The solution created will be responsible for control of requests from users of the system to the various issues within a company. Thus, a detailed overview on related activities can be shown, serving as support for decision making and better planning.

The solution named “Sistema de Gerenciamento de Help Desk” will be built using the JAVA programming language and uses Mysql database.

Key-words: CRM, Help Desk, Requests, Java, MEGA.

SIGLAS

UFRJ – Universidade Federal do Rio de Janeiro

CRM – Customer Relationship Management (Gerenciamento de Relacionamento com o Cliente)

Help Desk - Termo inglês que designa o serviço de apoio a usuários para suporte e resolução de problemas técnicos em informática, telefonia e tecnologias de informação. Este apoio pode ser tanto dentro de uma empresa (profissionais que cuidam da manutenção de equipamentos e instalações dentro da empresa), quanto externamente (prestação de serviços aos usuários).

TI – Tecnologia de Informação.

SQL - Linguagem de Consulta Estruturada, do inglês *Structured Query Language*

ROI – Retorno do Investimento, do inglês *Retunrn of Investment*

Basel II – Segundo acordo de Basel que faz recomendações à leis bancárias e regulatórias.

Solvency II – Conjunto de requisitos regulamentares para empresas de seguros que operam na União Europeia.

ORM – Gerenciamento de Riscos Operacionais, do inglês *Operational Risk Management.*

Smalltalk – Linguagem de Programação Orientada a Objeto

Simula 67 – Linguagem de Programação, projetada para apoiar a simulação de eventos discretos.

MVC – Padrão de Arquitetura de Software, do inglês *Model-view-controller.*

XML – É uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais, do inglês *eXtensible Markup Language*.

UI – Interface do Usuário, do inglês *User Interface.*

HTTP – Protocolo de comunicação via web, do inglês *Hypertext Transfer Protocol.*

API – Interface de Programação de Aplicativos, do inglês *Application Program Interface.*

GPL – Licença para Software Livre, do inglês *General Public License.*

OLAP – Aplicações com capacidade de manipular e analisar grande volume de dados sobre múltiplas perspectivas, do inglês *On-line Analytical Processing.*

GIS – Sistema de Infomações Geográficas, do inglês *Geographic Information Systems.*

TCP/IP – Conjunto de protocolos de comunicação entre computadores em rede.

FTP – Protocolo de Transferência de Arquivos, do inglês *File Transfer Protocol.*

LGPL – Licença de Software Livre, do inglês *Lesser General Public License.*

HQL – Dialeto SQL para o *framework* Hibernate, do inglês *Hibernate Query Language.*

SGBD – Sistema Gerenciador de Banco de Dados.

JTA – API pertencente à plataforma Java EE, do inglês *Java Transaction API.*

JDBC – Conjunto de classes e interfaces (API) escritas em Java que faz o envio de instruções SQL para qualquer banco de dados relacional, do inglês Java Database Conectivity.

Sumário

Lista de Figuras

Capítulo 1

Introdução

1.1 – Tema

O Help desk é um serviço que visa o atendimento a reclamações de clientes. A central de Help desk atua como elo entre a empresa e seus clientes, deve possuir um sistema ágil de registro e auxílio à solução de problemas, roteamento para especialistas, registros da solução e correlação que permitem ações pró-ativas de caráter preventivo ou corretivo.

O tema do trabalho é o estudo de uma solução sistemática para o helpdesk de empresas que não possuem a área de TI como área fim.

1.2 – Delimitação

O conceito de help desk é bastante amplo e abrange funções como:

- Gerenciar e administrar as solicitações de atendimento, problemas e as ordens de serviço (tarefas para solucionadores);
- Fazer o efetivo acompanhamento dos reportes de problemas e o andamento de suas soluções e o tempo de execução das tarefas;
- Responder a questões e coordenar a solução de problemas dos clientes, controlando os processos de atendimento de forma organizada;
- Mensurar o nível de satisfação dos clientes com relação à Empresa e aos seus serviços;
- Emitir relatórios gerenciais para acompanhamento de performances por atendimento ou por responsável, indicadores operacionais, reclamações por serviço, região, etc.;
- Registrar, acompanhar e solucionar reclamações de Clientes e Usuários internos;
- Criar um banco de conhecimento de problemas e soluções conhecidas;

- Registrar, acompanhar e solucionar reclamações de Clientes e Usuários internos;
- Estabelecer uma comunicação única e personalizada com os clientes, independente da procedência e do assunto que gerou o contato.

Dentro deste projeto não estarão incluídos os parâmetros que têm como função mensurar a satisfação dos clientes com relação à Empresa e aos seus serviços. O foco deste projeto é a abertura de chamados de manutenção e pedidos de suprimentos.

1.3 – Justificativa

O desenvolvimento do sistema justifica-se pela necessidade de uma instituição de possuir uma ferramenta controladora de chamados. Desta forma, o usuário terá um atendimento mais rápido e personalizado e a instituição terá um controle maior sobre quais os problemas mais comuns e a melhor forma de solucioná-los.

1.4 – Objetivos

O Help desk tem com objetivo receber os problemas a serem resolvidos por um determinado setor. É um sistema facilitador de informações. A proposta deste trabalho é a modelagem e o desenvolvimento de um Sistema de Gerenciamento de Help Desk para ser utilizado por uma instituição qualquer, que não possua a área de TI como área fim.

1.5 – Metodologia

O Sistema de Gerenciamento de Help Desk irá informatizar o processo de abertura de chamado para resolução de problemas, feitos ao setor responsável. Além de ser informatizado também o processo de solicitação de suprimentos feito a um fornecedor através do setor de compras.

Para tal, será utilizada a Metodologia MEGA, uma metodologia voltada para o suporte aos processos de negócio. Nela serão descritas todas as informações necessárias à construção do sistema, seja a parte das regras de negócio, descrição de funcionalidades e telas, quanto a parte técnica, com seu modelo de dados, tabelas e definições de cada um dos campos do sistema.

Uma descrição detalhada sobre a Metodologia MEGA será dada no apítulo 2 deste projeto.

1.6 – Descrição

No capítulo 2 será descrita a metodologia de projeto utilizada, baseada na Metodologia MEGA.

O capítulo 3 apresenta as tecnologias utilizadas no projeto, mostrando como foram utilizadas e sua presença no mercado de software na atualidade.

Uma avaliação de como foi o andamento do projeto e as conclusões sobre o mesmo são mostradas no capítulo 4.

Os próximos passos com relação ao projeto serão descritos no capítulo 5.

Os Apêndices conterão os documentos do projeto de acordo com os modelos propostos pela metodologia MEGA.

Capítulo 2

Metodologia MEGA

2.1 - Introdução

A metodologia MEGA é uma metodologia de suporte aos processos de negócio de uma empresa ou de um projeto, por ser altamente flexível ela pode ser aplicada em qualquer área de uma empresa necessitando apenas de poucos ajustes.

O formato utilizado para esta metodologia foi criado pela empresa TalentWork Services tomando como base as diversas metodologias existentes no mercado, focando neste caso a implantação de sistemas em geral.

A metodologia MEGA propõe uma estratégia de implementação da gestão do risco operacional. À medida que aumentam os requisitos de conformidade, existe cada vez maior pressão sobre os executivos para a gestão dos riscos operacionais de forma rápida e eficiente. Num ambiente em mudança, os executivos precisam provar que conseguem gerir os riscos de forma eficiente e controlar as suas operações.

A metodologia tem como objetivo ajudar os gestores de controle interno e de controle do risco a implementarem uma solução flexível para a gestão do risco operacional, a fim de facilitar a mudança e a melhorar o desempenho da empresa.

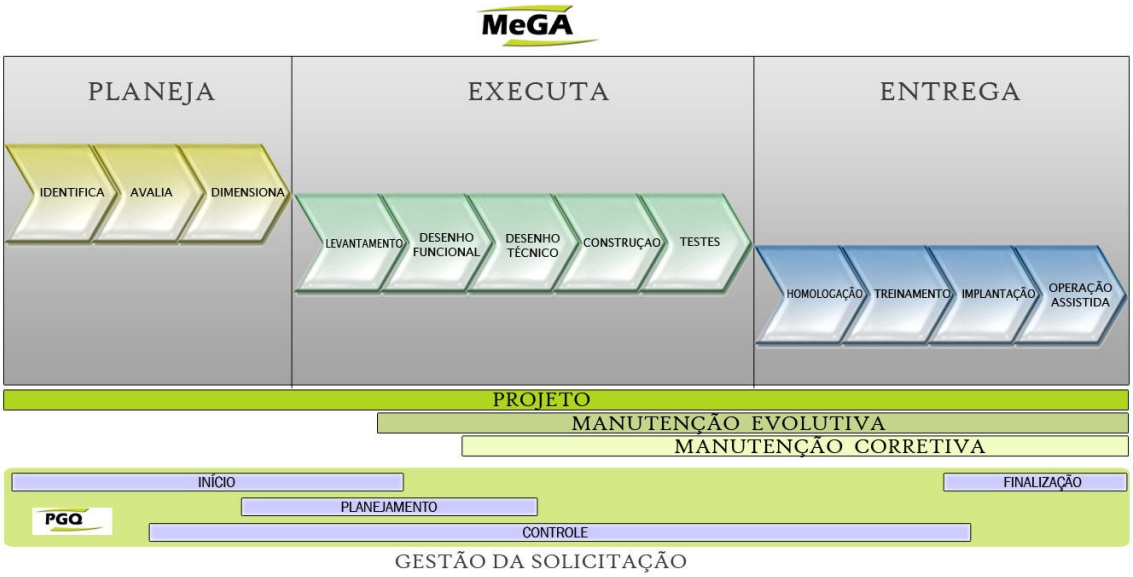

Figura 1 – Metodologia MEGA

A metodologia MEGA permite uma abordagem orientada ao ROI, através:

- De uma abordagem flexível, que permite a integração dos conceitos, vocabulário e metodologias próprias da empresa, assegurando assim o envolvimento de todos os stakeholders, de acordo com as necessidades.
- Do nível certo de controle para acompanhar todas as operações através de auditorias, com acesso automático ao histórico dos registos em qualquer altura.
- Do enfoque na informação e na comunicação, com uma identificação precisa das responsabilidades e das regras de confidencialidade, bem como o suporte para a consciência de toda a empresa relativamente às políticas de gestão do risco.
- De uma plataforma standard e escalável para apoiar o desenvolvimento e implementação da solução na organização.

Seguindo uma abordagem de negócio, a metodologia MEGA permite uma compreensão priorizada dos riscos e do seu impacto no negócio. Fornece assim aos decisores e a todos os stakeholders uma visão partilhada e transparente à política de gestão do risco na organização, através da transferência de informação, portais empresariais, e um fluxo apropriado de relatórios. O resultado é a oportunidade de transformar a atividade de conformidade em valor de negócio para a organização. As vantagens chave desta solução incluem:

- Suporte dos requisitos de conformidade Basel II ou Solvency II, com a possibilidade de reutilizar elementos e de apoiar múltiplas regulamentações;
- Envolvimento dos gestores do negócio no suporte da política de ORM, através de um sistema de avisos e de mensagens, e da definição de perfis de utilizador para proteger o acesso a dados sensíveis;
- Construção de uma ponte entre o negócio e os departamentos de TI, a fim de assegurar que os riscos são geridos de forma consistente, desde a perspectiva do negócio, até a infra-estrutura de sistemas de informação;
- Transformação da conformidade numa oportunidade para aumentar o desempenho da organização, através da integração da informação da gestão do risco operacional numa iniciativa mais ampla de gestão desse risco.

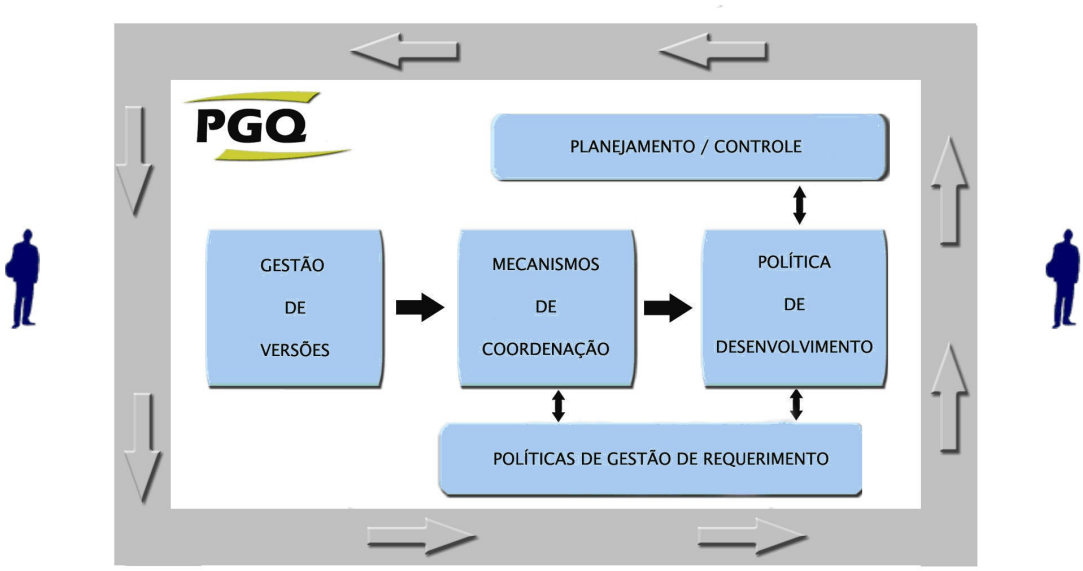

Figura 2 – Política de Gestão de Qualidade MEGA

Para este projeto serão utilizados 3 documentos que servirão como base para a documentação do projeto:

1. Documento Funcional: Este documento tem como objetivo discriminar todos os processos e regras de negócio existentes no projeto, explicando conceitualmente tudo o que será realizado.

2. Documento Técnico: Este documento tem como objetivo discriminar todo o desenvolvimento do projeto, descrevendo o modelo de dados, os relacionamentos entre as tabelas, definições de campos, etc.
3. Plano de Testes: Este documento tem como objetivo garantir que o desenvolvimento, especificado no Documento Técnico, atenda as regras de negócio descritas no Documento Funcional, executando testes unitários e integrados no sistema a ser construído.

2.2 - Descrição dos documentos

Neste item serão mostradas as estruturas presentes em cada um dos documentos mencionados anteriormente:

- Documento Funcional
- Documento Técnico
- Plano de Testes

2.2.1 - Documento Funcional

Este documento tem como objetivo discriminar todos os processos e regras de negócio existentes no projeto, explicando conceitualmente tudo o que será realizado.

O Documento Funcional é dividido da seguinte forma:

- Detalhes da Solução: Uma descrição simples da solução a ser construída informando as necessidades de negócio que serão atendidas.
  - Regras de Negócio: Serão informadas todas as regras do negócio, as quais o sistema deverá respeitar.
  - Suposições: Relação de premissas as quais o sistema se baseia.
  - Glossário: Dicionário de siglas e termos específicos que serão necessários para o entendimento do Documento e do negócio como um todo.
- Descrição Funcional: Descrição da solução a ser construída, assim como o procedimento do usuário e das necessidades a serem atendidas.

- Processos a Desenvolver: Relação de processos e funcionalidades presentes no sistema a ser construído (Exemplo: Informa dados do cliente, Botão Salvar Informações, Botão Gerar Relatório, etc.)

- Diagramas de Processos: Descrição dos relacionamentos entre os diversos processos existentes na solução, além do fluxo completo de cada um desses processos.
  - Interface com o Usuário: Descrição de como será feita a interação entre o usuário e o sistema.
  - Fluxo de Navegação: Descrição das alternativas para a utilização das diversas telas do sistema.
  - Telas: Layout das telas do sistema, informando ao usuário a descrição de cada um dos campos de cada uma das telas do sistema.

- Segurança de Acesso aos Dados: Definição dos requisitos de segurança e acesso aos dados para todos os elementos do sistema, sendo eles dados ou telas.

2.2.2 - Documento Técnico

Este documento tem como objetivo discriminar todo o desenvolvimento do projeto, descrevendo o modelo de dados, os relacionamentos entre as tabelas, definições de campos, etc.

O Documento Técnico é dividido da seguinte forma:

- Introdução: Breve comentário sobre o documento a ser escrito
  - Objetivos do Documento: Orientações técnicas para desenvolvimento da solução permitindo um melhor entendimento do que foi descrito no Documento Funcional. Comentários sobre a solução como o número de telas que serão construídas, workflows, etc.
  - Resumo da Solução: Tabela com nome, tipo e descrição de cada um dos objetos utilizados na solução (tabelas, telas, procedures, etc.). Resumo de como será construída a solução.

- Modelo de Dados: Informações sobre o padrão de dados a ser adotado na solução.

- Modelo Físico de Dados: Diagrama das tabelas os respectivos campos e seus relacionamentos.

- Objetos da Base de Dados: Descrição de cada um dos objetos que serão utilizados na solução.
  - Tabelas: Definição de cada uma das tabelas do sistema, informando nome de campos, obrigatoriedade de preenchimento, tipo de campo, etc.

- Interface com o Usuário: Descrição técnica da navegação entre as telas.
  - Tela: Descrição de acesso à tela.
  - Regra de Abertura: Validações realizadas ao acessar a tela.
  - Layout: Referência ao layout informado no Documento Funcional.
  - Campos: Descrição de cada um dos campos de cada uma das telas, informando seu tipo, obrigatoriedade, forma de visualização, dependências com outros campos e telas, além da informação de onde será armazenada a informação no banco de dados.
  - Eventos: Descrição de cada um dos eventos presentes na tela, como validações após preenchimentos de campos, informações para salvar dados, descrição do funcionamento dos botões, etc.

- Desenho Técnico VS Desenho Funcional: Correlação entre as funcionalidades descritas no Desenho Funcional com os objetos do Desenho Técnico.

2.2.3 - Plano de Testes

Este documento tem como objetivo garantir que o desenvolvimento, especificado no Documento Técnico, atenda as regras de negócio descritas no Documento Funcional, executando testes unitários e integrados no sistema a ser construído.

O Plano de Testes é dividido da seguinte forma:

- Pré-Requisitos: Listagem de tudo que é necessário para a perfeita execução do plano de testes.

- Funcionalidades: Relação de todos os cenários e passos de cenários a serem executados. Neste ponto definimos se o desenvolvimento está de acordo com o que foi especificado.

2.3 – Utilização da Metodologia

Os documentos descritos neste capítulo estão presentes nos Apêndices A, B e C deste projeto e lá serão encontradas informações detalhadas sobre o projeto.

Capítulo 3

Tecnologias e Ferramentas Utilizadas

3.1 – Introdução

Neste capítulo serão descritas as tecnologias e ferramentas utilizadas para a construção desta solução. Será dado um histórico, as características e as maneiras de utilização as mesmas.

3.2 – JAVA

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "*bytecode*" que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem.

3.2.1 – História

Em 1991, na Sun Microsystems, foi iniciado o *Green Project*, o berço do Java, uma linguagem de programação orientada a objetos. Os mentores do projeto eram Patrick Naughton, Mike Sheridan, e James Gosling. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a "próxima onda" do mundo digital. Eles acreditavam que, em algum tempo, haveria uma convergência dos computadores com os equipamentos e eletrodomésticos comumente usados pelas pessoas no seu dia-a-dia.

Para provar a viabilidade desta idéia, 13 pessoas trabalharam arduamente durante 18 meses. No verão de 1992 eles emergiram de um escritório de Sand Hill Road no Menlo Park com uma demonstração funcional da idéia inicial. O protótipo se chamava *7 (leia-se "*StarSeven*"), um controle remoto com uma interface gráfica

touchscreen. Para o *7, foi criado um mascote, hoje amplamente conhecido no mundo Java, o Duke. O trabalho do Duke no *7 era ser um guia virtual ajudando e ensinando o usuário a utilizar o equipamento. O *7 tinha a habilidade de controlar diversos dispositivos e aplicações. James Gosling especificou uma nova linguagem de programação para o *7. Gosling decidiu batizá-la de "Oak", que quer dizer carvalho, uma árvore que ele podia observar quando olhava pela sua janela.

O próximo passo era encontrar um mercado para o *7. A equipe achava que uma boa idéia seria controlar televisões e vídeo por demanda com o equipamento. Eles construíram um demo chamado *MovieWood*, mas infelizmente era muito cedo para que o vídeo por demanda bem como as empresas de TV a cabo pudessem viabilizar o negócio. A idéia que o *7 tentava vender, hoje já é realidade em programas interativos e também na televisão digital. Permitir ao telespectador interagir com a emissora e com a programação em uma grande rede de cabos, era algo muito visionário e estava muito longe do que as empresas de TV a cabo tinham capacidade de entender e comprar. A idéia certa, na época errada.

Entretanto, o estouro da Internet aconteceu e rapidamente uma grande rede interativa estava se estabelecendo. Era este tipo de rede interativa que a equipe do *7 estava tentando vender para as empresas de TV a cabo. E, da noite para o dia, não era mais necessário construir a infra-estrutura para a rede, ela simplesmente estava lá. Gosling foi incumbido de adaptar o Oak para a Internet e em janeiro 1995 foi lançada uma nova versão do Oak que foi rebatizada para Java. A tecnologia Java tinha sido projetada para se mover por meio das redes de dispositivos heterogêneos, redes como a Internet. Agora aplicações poderiam ser executadas dentro dos navegadores nos *Applets* Java e tudo seria disponibilizado pela Internet instantaneamente. Foi o estático HTML dos navegadores que promoveu a rápida disseminação da dinâmica tecnologia Java. A velocidade dos acontecimentos seguintes foi assustadora, o número de usuários cresceu rapidamente, grandes fornecedores de tecnologia, como a IBM anunciaram suporte para a tecnologia Java.

Desde seu lançamento, em maio de 1995, a plataforma Java foi adotada mais rapidamente do que qualquer outra linguagem de programação na história da computação. Em 2004, Java atingiu a marca de 3 milhões de desenvolvedores em todo mundo. Java continuou crescendo e hoje é uma referência no mercado de desenvolvimento de software. Java tornou-se popular pelo seu uso na Internet e hoje

possui seu ambiente de execução presente em navegadores *web*, *mainframes*, sistemas operacionais, celulares, *palmtops* e cartões inteligentes, entre outros.

3.2.2 – Características

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

- Orientação a objeto - Baseado no modelo de *Smalltalk* e *Simula67*;
- Portabilidade - Independência de plataforma - "*write once, run anywhere*";
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- Segurança - Pode executar programas via rede com restrições de execução;

Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:

- Sintaxe similar a Linguagem C/C++ e principalmente, a C#.
- Facilidades de Internacionalização - Suporta nativamente caracteres Unicode;
- Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução (JVM);
- É distribuída com um vasto conjunto de bibliotecas (ou APIs);
- Possui facilidades para criação de programas distribuídos e multitarefa (múltiplas linhas de execução num mesmo programa);
- Desalocação de memória automática por processo de coletor de lixo (*garbage collector*);
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

3.3 – Banco de Dados Mysql

O Mysql é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.

Entre os usuários do banco de dados Mysql estão: NASA, Friendster, Banco Bradesco, Dataprev, HP, Nokia, Sony, Lufthansa, U.S. Army, US. Federal Reserve Bank, Associated Press, Alcatel, Slashdot, Cisco Systems, CanaVialis S.A. e outros.

3.3.1 - História

O Mysql foi criado na Suécia por dois suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius, que têm trabalhado juntos desde a década de 1980. Hoje seu desenvolvimento e manutenção empregam aproximadamente 400 profissionais no mundo inteiro, e mais de mil contribuem testando o software, integrando-o a outros produtos, e escrevendo a respeito dele.

No dia 16 de Janeiro de 2008, a Mysql AB, desenvolvedora do Mysql foi adquirida pela Sun Microsystems, por US$ 1 bilhão, um preço jamais visto no setor de licenças livres. No dia 20 de Abril de 2009 a Oracle compra a Sun Microsystems e todos os seus produtos, incluindo o Mysql.

O sucesso do Mysql deve-se em grande medida à fácil integração com o PHP incluído, quase que obrigatoriamente, nos pacotes de hospedagem de *sites* da Internet oferecidos atualmente. Empresas como Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics e Texas Instruments usam o Mysql em aplicações de missão crítica. A Wikipédia é um exemplo de utilização do Mysql em sites de grande audiência.

O Mysql hoje suporta *Unicode*, *Full Text Indexes*, replicação, *Hot Backup*, GIS, OLAP e muitos outros recursos.

3.3.2 - Características

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem *drivers* ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP, ASP e Ruby)

- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre com base na GPL;
- Contempla a utilização de vários *Storage Engines* como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid, etc.;
- Suporta controle transacional;
- Suporta *Triggers*;
- Suporta *Cursors (Non-Scrollable e Non-Updatable);*
- Suporta *Stored Procedures e Functions*;
- Replicação facilmente configurável;
- Interfaces gráficas ([Mysql Toolkit]) de fácil utilização cedidas pela Mysql Inc.

3.4 – *Framework* Struts 2

Apache Struts 2 é um *framework* de aplicações web. Comparado com Apache Struts não é apenas uma nova versão, mas uma aplicação completamente nova. Struts 2 é a segunda geração do *framework* de aplicações web que implementa o *Model-View-Controller (MVC) design pattern*.

Apache Struts 2 é uma estrutura extensível para a criação de aplicações empresariais na Web Java pronto. O quadro é projetado para aperfeiçoar o ciclo de desenvolvimento, de construção, a implantação, a manutenção de aplicativos com o tempo. Depois de trabalhar de forma independente por vários anos, a comunidade *Webwork e Struts* se uniram para criar Struts 2. Esta versão do Struts afirma ser mais simples de usar e mais perto da visão original do Struts.

3.4.1 – História

Struts 2 é construído a partir da base sobre as melhores práticas e comprovadas, a comunidade aceita padrões de design. Isso também foi verdade para a primeira versão do Struts. De fato, um dos principais objetivos do Struts era incorporar o padrão MVC em aplicação *desktop* do mundo em um *framework* de aplicações web. O padrão resultante é, às vezes, chamado de modelo 2 *pattern*. Este foi um passo crucial na evolução das aplicações web bem concebido, uma vez que forneceu a infra-estrutura para atingir facilmente a separação MVC. Este desenvolvimento permitiu com poucos recursos para tais sutilezas de arquitetura para utilização em uma pronta solução de melhores práticas. Struts 1 pôde reivindicar a responsabilidade de muitas das aplicações web melhor desenhado dos últimos 10 anos. Em algum momento, a comunidade de Struts ficou ciente das limitações no primeiro quadro. Com uma comunidade tão ativa, identificando os pontos fracos e inflexíveis no quadro não foi difícil de realizar. Struts 2 tira partido das muitas lições aprendidas para apresentar um aspirador de implementação de MVC. Ao mesmo tempo, introduz várias novas características arquitetônicas que compõem o quadro mais limpo e mais flexíveis.

Esses novos recursos incluem interceptores para camadas transversais preocupações longe da lógica de ação; configuração com o intuito de reduzir ou eliminar a configuração de XML, uma linguagem de expressão potente, *Object-Graph Navigation Language* (OGNL), que atravessa todo o quadro, e um mini *MVC-tag-based API* que suporta componentes modificáveis e reutilizáveis UI. Struts 2 senta em cima de duas importantes tecnologias. No centro de todas as aplicações Struts 2 encontram-se os intercâmbios cliente / servidor do protocolo HTTP. O Java Servlet API expõe esse baixo nível da camada HTTP com a linguagem Java.

3.5 – Hibernate

O Hibernate é um *framework* para o mapeamento objeto-relacional escrito na linguagem Java, mas também é disponível em .Net como o nome NHibernate. Este programa facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) para estabelecer esta relação.

Hibernate é um software livre de código aberto distribuído com a licença LGPL.

3.5.1 – História

A HQL (Hibernate Query Language) é um dialeto SQL para o Hibernate. Ela é uma poderosa linguagem de consulta que se parece muito com a SQL, mas a HQL é totalmente orientada a objeto, incluindo os paradigmas de herança, polimorfismo e encapsulamento.

No Hibernate, você pode escolher tanto usar a SQL quanto a HQL. Escolhendo a HQL, você poderá executar os pedidos SQL sobre as classes de persistência do Java ao invés de tabelas no banco de dados, aumentando, assim, a distância entre o desenvolvimento da regras de negócio e o banco de dados.

3.5.2 – Características

O objetivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional (presente na maioria dos SGDBs). Em especial, no desenvolvimento de consultas e atualizações dos dados.

Sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL). O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual da conversão dos dados resultante, mantendo o programa portável para quaisquer bancos de dados SQL, porém causando um pequeno aumento no tempo de execução.

Nas questões relacionadas para o gerenciamento de transações e na tecnologia de acesso à base de dados são de responsabilidade de outros elementos na infra-estrutura do programa. Apesar de existirem API no Hibernate para possuir operações de controle transacional, ele simplesmente delegará estas funções para a infra-estrutura na qual foi instalada.

No caso de aplicações construídas para serem executadas em servidores de aplicação, o gerenciamento das transações é realizado segundo o padrão JTA. Já nas aplicações *standalone*, o programa delega o tratamento transacional ao driver JDBC.

Hibernate pode ser utilizado em aplicações Java standalone ou em aplicações Java EE, utilizando *servlet* ou sessões EJB beans.

Capítulo 4

Conclusões

4.1 – Análise Crítica

A utilização da metodologia MEGA foi bastante satisfatória. O estudo da mesma mostrou que esta metodologia é bastante flexível podendo ser utilizada para desenvolvimentos em qualquer linguagem ou na implementação de ERP's. Sua utilização abrange todas as fases de um projeto independente do tipo de negócio do mesmo e das ferramentas utilizadas.

Apesar da utilização desta metodologia em outros projetos, mais especificamente em projetos ERP, a MEGA se mostrou completamente adaptada à realidade de um projeto desenvolvido na linguagem Java. Em projetos ERP, esta metodologia é utilizada para desenvolvimentos pontuais, no caso deste projeto a única diferença é a escala do mesmo, onde todas as informações se encontram e se conectam tornando-a de fácil entendimento.

Como pontos negativos, podem se destacar a ausência de informações típicas de projetos realizados em Java, como os casos de uso e descrição de classes entre outras. Para tal, a metodologia deveria sofrer algumas adaptações para que facilitasse o desenvolvimento do sistema.

4.2 – Análise das Ferramentas

A linguagem Java possui muitas funcionalidades, o que aumenta seu nível de dificuldade, porém devido a sua ampla divulgação, a busca por exemplos e sites para que dúvidas sejam tiradas se torna fácil. O banco de dados Mysql, de ampla divulgação como a linguagem Java, se tornou muito eficiente e proporciona uma ótima performance para o sistema. As ferramentas de programação Hibernate e Struts 2

auxiliaram bastante no desenvolvimento do projeto, fazendo com que as regras de negócio do sistema fossem adaptadas de maneira simples à estrutura já disponibilizada, agilizando o desenvolvimento.

Capítulo 5

Próximos Passos

5.1 – Introdução

Neste capítulo serão descritas melhorias que podem ser implantadas em uma nova versão deste sistema, aumentando sua confiabilidade, seu número de funcionalidades e seu valor agregado.

5.2 – Propostas para melhorias

Dentre as melhorias levantadas para possíveis novas versões deste sistema estão:

- Envio de emails para os envolvidos: Esta solução a incorporação neste projeto da instalação e configuração de um servidor de emails para que todos os envolvidos em um chamado sejam comunicados de cada estágio do mesmo.
- Implantação de Índices de Satisfação dos Chamados: Esta solução visa avaliar os atendimentos realizados, monitorando-os e estabelecendo índices como agilidade, qualidade, etc.
- Maior número de informações: Aumento do detalhamento de cada um dos itens apresentados no projeto, isso tornaria o sistema mais informativo e útil ao usuário.
- Integração com software financeiro: Comunicação com algum sistema financeiro de forma a enviar dados de compra criados pelo sistema e que as informações financeiras sejam registradas corretamente.

Bibliografia

[1] PRESSMAN, Roger, *"Engenharia de Software"*. McGraw Hill

[2] DAMIANI, Edgard B., *"Guia de Consulta Rápida Java Script"*. Novatec

[3] www.mysql.com

[4] http://pt.wikipedia.org/wiki/Help_desk

[5] http://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o)

[6] http://pt.wikipedia.org/wiki/Hibernate

[7] http://en.wikipedia.org/wiki/Struts2

[8] http://pt.wikipedia.org/wiki/MySQL

Apêndice A – Documento Funcional

Apêndice B – Documento Técnico

Apêndice C – Plano de Testes