

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-
GRANDENSE - CÂMPUS PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

FABIO VITOR VEDDY

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA WEB RESPONSIVO PARA
GERENCIAMENTO DE ORDEM DE SERVIÇOS**

Prof^a. Ma. Vanessa Lago Machado

**PASSO FUNDO
2017**

FABIO VITOR VEDDY

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA WEB RESPONSIVO PARA
GERENCIAMENTO DE ORDEM DE SERVIÇOS**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientadora: Prof^a. Ma. Vanessa Lago Machado

Coorientadora: Pro^a. Dr^a. Anubis Graciela de Moraes Rossetto

PASSO FUNDO

2017

FABIO VITOR VEDDY

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA WEB RESPOSIVO PARA
GERENCIAMENTO DE ORDEM DE SERVIÇOS**

Trabalho de Conclusão de Curso aprovado em ____/____/____ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

Orientadora: Prof^a. Ma. Vanessa Lago Machado

Coorientadora: Prof^a. Dr^a. Anubis Graciela de Moraes Rossetto

Prof. Ma. Carmen Vera Scorsatto

Prof. Dr. Josué Toebe

Prof. Me. Rafael Marisco Bertei
(Coordenador do Curso)

PASSO FUNDO

2017

*Aos meus pais e família, pela compreensão
e o estímulo
em todos os momentos.*

AGRADECIMENTOS

Agradeço a minha mulher e minha família pela compreensão e apoio em todos os momentos, e por todos os finais de semana e feriados que dediquei exclusivamente aos estudos. Obrigado pelo auxílio.

Aos amigos que fiz durante o curso que poderei levá-los para a vida inteira. Agradeço pelo conhecimento que adquirimos juntos, pelas muitas vezes virando a madrugada estudando e a amizade que fizemos durante nosso convívio.

A minha orientadora Vanessa Lago Machado e demais professores, pela paciência, dedicação e ensinamentos passados em aula e pelo grande auxílio na construção deste trabalho e de meu conhecimento.

“Procure ser um homem de valor, em vez de ser um homem de sucesso.”

Albert Einstein

RESUMO

Atualmente, com a alta competitividade do mundo do trabalho, as empresas necessitam, cada vez mais, buscar novas soluções estratégicas para seus negócios. A partir dessa premissa, foi proposto o desenvolvimento de um sistema web responsivo de gerenciamento de ordens de serviço (OS). A proposta foi desenvolver um sistema para que as empresas possam fazer o gerenciamento de suas OS com mais precisão e menos desperdícios, tanto em relação à mão de obra, quanto aos materiais. Como resultado deste trabalho, foi desenvolvido um sistema genérico que realiza a gestão das OS de forma precisa e segura, assim, tal sistema pode ser utilizado por empresas prestadoras de serviços de diversos segmentos.

Palavras-chave: Ordem de serviço; Java EE; Responsividade; Sistemas web.

ABSTRACT

Today, with a high competitiveness of the world of work, the companies need, more and more, look for new strategic solutions for their business. From this premise, it was proposed the development of a responsive web system for managing services orders (OS). This proposal was the development a system so that the company can manage its OS with more precision and less waste, both for the workforce and for the materials. As a result of this work, a generic system was developed that performs the management of the OS in a precise and safe way, so that such a system can be used by companies providing services of various segments.

Keywords: Services Orders; Java EE; Responsiveness; Web System.

LISTA DE FIGURAS

Figura 1 - Características de outras linguagens integradas pelo JAVA (parte 1).....	15
Figura 2 - Características de outras linguagens integradas pelo JAVA (parte 2).....	15
Figura 3 - Versões do Java 2	17
Figura 4 - Resumo Pacotes Hibernate	22
Figura 5 - Tipos de usuários do sistema	24
Figura 6 - Processo de abertura e encerramento de OS.....	25
Figura 7 – Diagrama de Classes do Sistema de Gerenciamento de OS	26 <u>27</u>
Figura 8 – Diagrama de Caso de Uso do Sistema de Gerenciamento de OS	27 <u>28</u>
Figura 9 -Tela login	28
Figura 10 - Tela inicial	28 <u>29</u>
Figura 11 - Cadastro de Sintoma, Produto e Causa.....	29 <u>30</u>
Figura 12 - Cadastro de Permissões	30
Figura 5 - Cadastro de Usuários	30 <u>31</u>
Figura 14 - Cadastro de OS	31 <u>32</u>
Figura 15 - Listagem de OS.....	32 <u>33</u>

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicativos
EE	<i>Enterprise Edition</i> (Edição empresarial)
EJB	<i>Enterprise JavaBeans</i>
GNU	<i>Gnu is Not Unix</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
J2ME	<i>Java 2 Micro Edition</i>
JDBC	<i>Java Database Connectivity</i>
JEE	<i>Java Enterprise Edition</i>
JPA	<i>Application Programming Interface</i>
JPQL	<i>Java Persistence Query Language</i>
JSF	<i>Java Server Faces</i>
JSP	<i>Java Server Pages</i>
JTA	Java Transaction API
MVC	<i>Model-view-controller</i> (modelo-visão-controlador)
OS	Ordem de Serviço
PNAD	Pesquisa Nacional por Amostra de Domicílios
SQL	<i>Structured Query Language</i>
SI	Sistema de Informações
TIC	Tecnologias de Informação e Comunicação
UI	<i>User Interface</i> (Interface usuário)
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	OBJETIVOS	10 11
1.1.1	Objetivo Geral	11
1.1.2	Objetivos específicos.....	11
1.2	JUSTIFICATIVA.....	11
2	REFERENCIAL TEÓRICO	13
2.1	SISTEMAS DE ORDEM DE SERVIÇO	13
2.2	JAVA	14
2.2.1	Java EE	17
2.3	COMPUTAÇÃO MÓVEL	18 19
2.4	SITES RESPONSIVOS	19
2.5	PRIMEFACES	20
2.6	JPA	20
2.7	HIBERNATE	21
2.8	EJB	22
3	DESENVOLVIMENTO DO SISTEMA.....	23
3.1	LEVANTAMENTO DE REQUISITOS	23
3.1.1	Requisitos funcionais.....	23
3.1.2	Requisitos não funcionais.....	24
3.2	PROJETO.....	24
3.3	MODELAGEM DO SISTEMA.....	25
3.4	<i>TELAS</i>	27 28
4	CONSIDERAÇÕES FINAIS.....	3334
	REFERÊNCIAS.....	3536

1 INTRODUÇÃO

Gerenciamento é fundamental em empresas que operam no ramo de prestação de serviços, porém, percebe-se que muitas dessas empresas acabam consumindo muito tempo e dinheiro para gerenciar os serviços ofertados, por meio de métodos que não utilizam a tecnologia a seu favor.

O controle de produtividade é muito importante também nas organizações, podendo fazer um bom planejamento, por meio do uso eficiente da mão de obra e das ferramentas de atividades. O objetivo principal do controle de produtividade é a programação de atividades, o que possibilita mensurar a disponibilidade da equipe. Um planejamento adequado à condição da empresa pode ocasionar em benefícios, como maior rendimento dos colaboradores perante as suas tarefas, gestão na programação e execução dos trabalhos. Como resultado, equipes produzem mais e, portanto, garantem um melhor desempenho ao negócio.

Outro problema é que as empresas consomem mais papel do que o necessário em suas Ordens de Serviços (OS), as quais são entregues aos técnicos, responsáveis por executar os serviços em campo. Assim, torna-se necessário que o técnico se desloque até a base da empresa para retirar as OS, para só então poder executar os serviços solicitados. Tal fato acaba culminando em lentidão e burocracia na realização dos serviços prestados. Além disso, o formato manual acaba gerando custos desnecessários com telefonemas e aumento do consumo do combustível no caso de alteração de algum dado da OS. Devido a esse cenário, não eficiente, verificou-se que a tecnologia ainda não está sendo usada em sua totalidade nesses casos.

Logo, considerando as questões apresentadas acima, o foco do presente trabalho foi a análise e o desenvolvimento de um sistema para gerenciamento de OS, por meio do qual será possível gerenciar equipes de trabalhos remotamente com o objetivo de priorizar a qualidade do serviço ofertado.

1.1 OBJETIVOS

Nesta sessão são apresentados o objetivo geral e os objetivos específicos do trabalho realizado.

1.1.1 Objetivo Geral

Analisar e desenvolver um sistema responsivo para gerenciar OS por meio de um sistema web.

1.1.2 Objetivos específicos

- Conhecer outros sistemas similares;
- Analisar os requisitos.
- Desenvolver o layout do sistema,
- Desenvolver um sistema genérico para gerenciamento de OS, por meio de protótipo da solução; e
- Realizar os devidos testes.

1.2 JUSTIFICATIVA

Devido ao grande número de empresas que disponibilizam serviços para Internet, o desenvolvimento Web tornou-se um ramo muito importante na indústria de software.

A mudança de ambiente trouxe de volta uma série de problemas já resolvidos pelas aplicações tradicionais. Várias soluções já estão disponíveis para resolver tais problemas, mas, devido à grande oferta, é difícil escolher uma solução para a organização. Até mesmo identificar quais são as opções disponíveis e estáveis acaba tornando-se uma tarefa complexa.

O desenvolvimento do sistema foi motivado pela falta de sistemas gratuitos para pequenas empresas, as quais muitas vezes não possuem condições para realizar esse tipo de investimento. Assim, as empresas poderão prestar serviços melhores, com a devida gerência na prestação desses.

Este trabalho está organizado desta maneira: o capítulo 1 consiste em mostrar a motivação do desenvolvimento do trabalho, seus objetivos gerais e específicos. O capítulo 2 mostra o referencial teórico, o qual descreve toda a base teórica de diferentes autores que realizaram estudos de sistemas de gerenciamento, sistemas de informação e alguns assuntos que rondam um sistema de gerenciamento de ordens de serviços. Por fim, o capítulo 3 mostra todo o

desenvolvimento de sistemas, que é composto pela análise de requisitos, desenvolvimento da codificação e estruturação do sistema.

2 REFERENCIAL TEÓRICO

Nesta seção é apresentada uma revisão bibliográfica sobre sistemas de ordem de serviços, um breve histórico acerca da linguagem Java e suas plataformas, assim como sobre computação móvel e sites responsivos.

2.1 SISTEMAS DE ORDEM DE SERVIÇO

De acordo com Laudon e Laudon (1999 apud Emmendoerfer, 2002), um sistema de informação (SI) pode ser definido como um conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informação com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em empresas e outras organizações. Os sistemas contêm dados sobre pessoas, lugares e coisas de interesse, no ambiente ao redor da organização e dentro da própria organização.

Os SI basicamente coordenam o fluxo de trabalho de uma empresa, transformando a informação em uma forma que possa ser utilizada, ou seja, auxiliando todos os envolvidos no processo para uma tomada de decisão mais ágil, na análise de assuntos complexos, e também na resolução de outros problemas que possam vir a surgir. Portanto, é necessário que a informação esteja sempre atualizada. Também cabe ressaltar que os SI possuem um fluxo com três ciclos básicos: entrada, processamento e saída (LAUDON e LAUDON, 1999 apud EMMENDOERFER, 2002).

Ainda, os autores explicam que algumas ações são necessárias quando os SI são computadorizados, como, por exemplo, para captação dos dados de uma organização (sejam dados internos ou externos), a qual, até o momento, é realizada por meio de formulários em papel ou registros telefônicos. Contudo, essas informações devem ser registradas e inseridas em um sistema computacional.

Assim, as atividades de captação dos dados fazem parte do ciclo de entrada, tais como registro, codificação, classificação e edição, preocupando-se em assegurar que os dados necessários se encontrem corretos e completos. Durante o ciclo de processamento, os dados são organizados, analisados e manipulados por

meio de cálculos, comparações, resumos e classificações, objetivando uma forma de disposição mais significativa e útil.

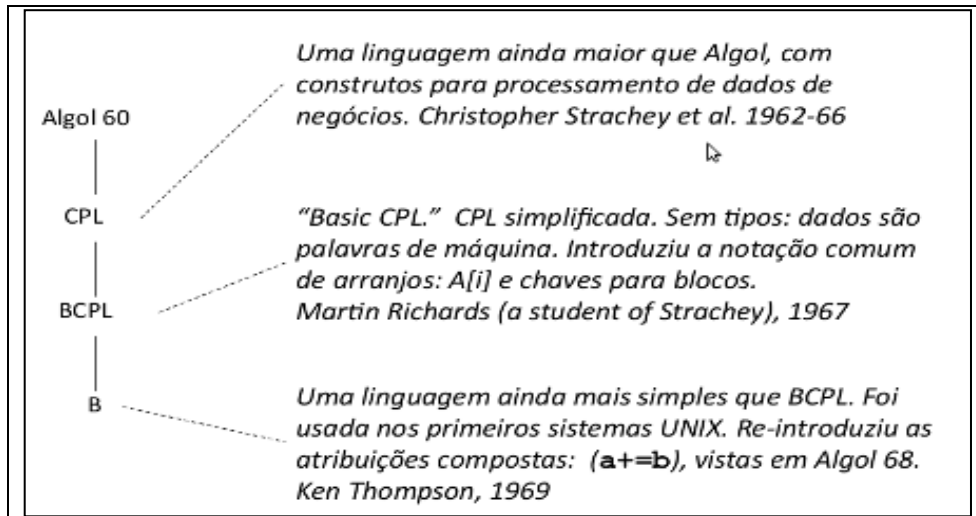
No decorrer do ciclo de saída, são realizadas as atividades que transmitem os resultados do processamento a locais onde serão usados, como: por exemplo projeto, inovação, coordenação e controle, além das tomadas de decisões. Desse modo, o resultado, ou a saída, do SI pode ser apresentado de várias formas - como relatórios impressos, apresentações gráficas, vídeos, som ou dados a serem enviados a outros SI's (LAUDON e LAUDON, 1999 apud EMMENDOERFER, 2002).

2.2 JAVA

O surgimento dos compiladores, na década de 50, associado com a evolução dos microprocessadores, nas décadas seguintes, proporcionou o desenvolvimento em massa dos computadores pessoais. Muitos estudiosos acreditavam que a próxima inovação seria a associação dos microprocessadores aos dispositivos eletrônicos. Reconhecendo essas ideias, a Sun Microsystems, lançou e financiou, em 1991, um grupo de pesquisa denominado *Green Project*, que resultou no desenvolvimento de uma linguagem baseada em C e C++, a qual seu criador, James Gosling, chamou de Oak, podendo ser traduzido para “carvalho”, em homenagem a uma árvore de carvalho que podia ser vista na janela do seu escritório. Entretanto, mais tarde, descobriu-se que já havia uma linguagem de programação com esse nome, foi então que sugeriu o nome Java (nome de uma cidade exportadora de café) devido ao grande consumo da bebida durante a realização do projeto. Atualmente, a Oracle é a empresa que desenvolve e mantém a linguagem (GUIMARÃES, 2012).

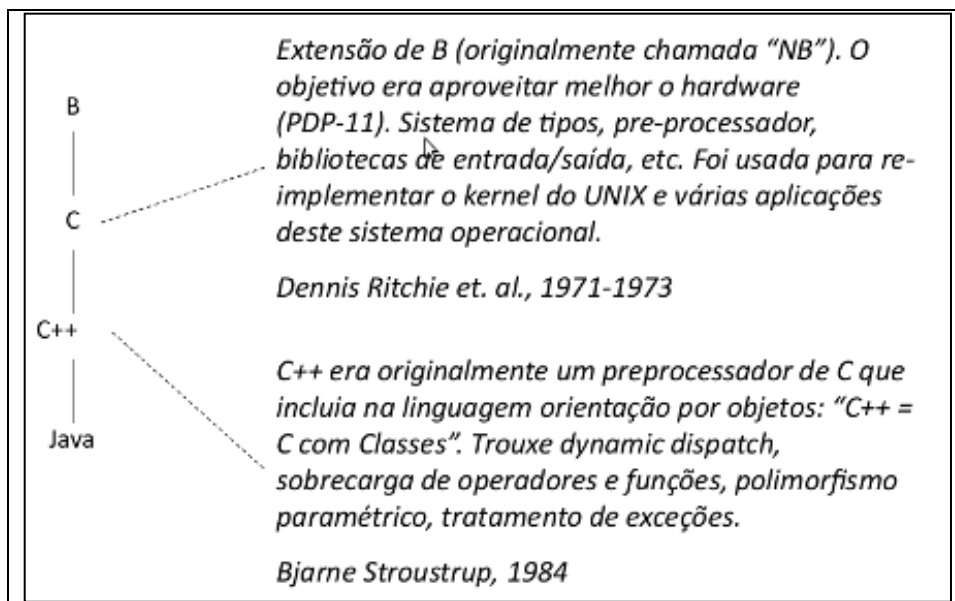
Nas Figuras 1 e 2 é possível visualizar um pouco das linguagens que foram integradas durante a criação da linguagem Java. Cada uma delas contribuiu com um pouco, seja estrutura, função, etc.

Figura 1 - Características de outras linguagens integradas pelo JAVA (parte 1)



Fonte: Guimarães, 2012.

Figura 2 - Características de outras linguagens integradas pelo JAVA (parte 2)



Fonte: Guimarães, 2012.

A primeira invenção da equipe *Green Project* foi o *7 (*StarSeven*), um controle remoto com interface *touchscreen*, contendo um “mascote” que ensinava o usuário a utilizar o controle. Ele foi um antecessor da ideia de interação digital, que está presente nos televisores digitais atualmente. Porém, infelizmente a tecnologia da época não estava preparada para tamanho avanço (PACIEVITCH, 2016).

A tecnologia Java teve uma enorme utilização, e logo grandes empresas, como a IBM, anunciaram que dariam suporte a linguagem, ou seja, os produtos

dessas empresas iriam rodar aplicativos desenvolvidos em Java. Estudos apontam que, até o momento, Java foi a tecnologia com maior aceitação e integração com o mercado de trabalho. Em 2003, ela já tinha mais de quatro milhões de desenvolvedores. A ideia inicial do *Green Project* começou a se concretizar e a linguagem passou a ser utilizada em dezenas de produtos diferentes como computadores, celulares, palmtops e a maioria dos produtos da Apple (PACIEVITCH, 2016).

Em 2006, diversos segmentos e partes do Java foram transferidos para a licença de software livre, e a maioria já estava disponível para o público gratuitamente, tudo sob licença GNU. A linguagem Java foi uma revolução na interatividade e sua utilização aumenta a cada dia. Ademais, Java trata-se de uma linguagem de programação relativamente simples e dinâmica, que permite criar programas e aplicações para a web sem depender de outra linguagem (PACIEVITCH, 2016).

Conforme Palmeira (2012), a independência da plataforma possibilita que o programa escrito na linguagem Java possa ser executado em diferentes plataformas por meio do emulador *Java Virtual Machine* (JVM), o qual pode ser traduzido como máquina virtual Java. Pode-se também ser denominada como uma máquina virtual baseada em software que é executada dentro dos aparelhos eletrônicos que irá ler e executar os *bytes codes* do Java.

De acordo com o Fioresi (2015), as plataformas Java estão divididas em JSE, JEE e JME, as quais se encontram ilustradas na Figura 3:

Figura 3 - Versões do Java 2



Fonte: Fioresi, 2015.

- a) *Java Standard Edition* (JSE): Tecnologia projetada para computadores pessoais e ambientes de trabalho;
- b) *Java Enterprise Edition* (JEE): Tecnologia direcionada para aplicações baseadas no servidor, contendo suporte interno para JSP (*Java Server Pages*), XML (*eXtensible Markup Language*) e servlets;
- c) *Java Micro Edition* (JME): Tecnologia direcionada para dispositivos com poucos recursos computacionais como, por exemplo, palms e telefones celulares.

2.2.1 Java EE

A plataforma JEE é o padrão no *software* da empresa voltado para a comunidade, o qual é desenvolvido usando o *Java Community Process*, com contribuições de especialistas da indústria, organizações comerciais e *open source*, *Java User Groups*, e inúmeras pessoas. Cada versão integra novos recursos que se alinham com as necessidades da indústria, melhora a portabilidade de aplicações e aumenta a produtividade do desenvolvedor (ORACLE, 2014).

Em todos os setores, as empresas enfrentam o desafio de acomodar exigências cada vez maiores de acessos de alta velocidade de dados, de diversos clientes e transações seguras, e sem gerar custos adicionais extensos. Para ampliar

os investimentos de TI existentes e atender a essas demandas, os desenvolvedores têm consistentemente se adaptado à plataforma JEE (ORACLE, 2014).

Dentre as tecnologias da plataforma JEE, encontra-se *Java Server Faces* (JSF), que incorpora características de um framework no padrão *Model, View e Control* (MVC) para Web e de um modelo de interfaces gráficas baseado em eventos. Por basear-se no padrão de projeto MVC, uma de suas melhores vantagens é a clara separação entre a visualização do sistema e as regras de negócio (PITANGA, 2009).

Ainda, de acordo com Pitanga (2009):

A ideia do padrão MVC é dividir uma aplicação em três camadas: modelo, visualização e controle. O modelo é responsável por representar os objetos de negócio, manter o estado da aplicação e fornecer ao controlador o acesso aos dados. A visualização representa a interface com o usuário, sendo responsável por definir a forma como os dados serão apresentados e encaminhar as ações dos usuários para o controlador. Já a camada de controle é responsável por fazer a ligação entre o modelo e a visualização, além de interpretar as ações do usuário e as traduzir para uma operação sobre o modelo, onde são realizadas mudanças e, então, gerar uma visualização apropriada.

A tecnologia JSF é uma especificação que serve para criar interfaces gráficas para aplicações Web, a qual se trata de um padrão para quem usa JEE, por meio dela é possível criar aplicações ricas e dinâmicas com certa facilidade. Dentre as principais vantagens do uso de JSF em projetos, Pitanga (2009) cita:

- a) Permite que o desenvolvedor crie suas próprias *Users Interfaces* (UIs) a partir de um conjunto de componentes pré-definidos;
- b) Fornece um conjunto próprio de tags JSP para acessar os componentes;
- c) Reusa componentes das páginas;
- d) Associa eventos do lado cliente com os manipuladores dos eventos do lado servidor (os componentes de entrada possuem um valor local representando o estado no lado servidor); e
- e) Fornece separação de funções que estão envolvidas na construção de aplicações web.

2.3 COMPUTAÇÃO MÓVEL

A tecnologia tem trazido inúmeras inovações, todos os dias são criadas e divulgadas novas ferramentas que facilitam a execução de tarefas de uma forma prática e ágil. Uma dessas tecnologias são os *smartphones*, que permitem acesso à internet, assistir TV, fazer transações bancárias, jogar, escutar música, entre outros, ou seja, une diversos equipamentos em um só.

O acesso à internet é o principal recurso utilizado no *smartphone*. De acordo com o *site* Meio & Mensagem (2016), o celular ultrapassou o *desktop* como meio mais usado de acesso à internet no Brasil. E ainda, de acordo com o Suplemento de Tecnologias de Informação e Comunicação (TIC), da Pesquisa Nacional por Amostra de Domicílios (Pnad, 2014) realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o celular já é usado em 80,4% das casas com acesso à internet e o computador em 76,6%.

O *site* acrescenta que é a primeira vez que o celular toma a dianteira de acesso à web. Mais da metade dos 67 milhões de domicílios brasileiros passaram a ter acesso à internet em 2014, ou seja, 54,9%. Em 2013, esse percentual era 48%. Mais de 60% dessas casas estavam na área urbana e cerca de 136,6 milhões de pessoas, com idade superior a 10 anos, tinham celular em 2014, no país. O número representa 77,9% dessa população, com um aumento de quase 5% em relação a 2013 (6,4 milhões de pessoas), e de 142,8% em relação a 2005.

2.4 SITES RESPONSIVOS

Nem sempre é possível garantir que o cliente ou funcionário esteja em frente a um *desktop* no momento em que ele precisa usar o sistema ou os serviços que as empresas fornecem. Esse é o caso em que é de extrema importância que a fornecedora do serviço possua um site responsivo, que pode ser acessado desde um *desktop* à plataforma *mobile*, ou seja, um site que se adapta às diferentes resoluções das telas automaticamente, possibilitando, dessa forma, que o cliente entre em contato, ou que o funcionário tenha acesso aos dados independentemente de onde estejam.

O design responsivo é aquele que tem em suas linhas de código parâmetros para melhorar a forma como o *layout* será exibido em diferentes resoluções de tela, que basicamente se resume em *smartphones*, *tablets*, *notebooks*, *desktops* e televisores *smart* (RENE, 2016).

Por sua vez, *Bootstrap* trata-se de um *framework* CSS (*Cascading Style Sheets*) que auxilia no desenvolvimento *front-end*, parte visível do site. É uma solução criada em 2010 por dois desenvolvedores do Twitter, Mark Otto (@mdo) e Jacob (@fat). Apesar desse *framework* ajudar muito no desenvolvimento, ele não faz tudo sozinho. Para começar é necessário conhecimento em HTML e CSS. Com ele pode-se reproduzir qualquer *layout* presente na web, desde que tenha o conhecimento necessário em CSS para realizar os ajustes necessários no código (Rene, 2016).

Assim, quando se fala em JSF existe um *framework* desenvolvido para essa tecnologia, o *Bootsface*. Ele possui várias características do *Bootstrap*, o qual possui como foco o desenvolvimento *front-end* de aplicativos corporativos de modo rápido, fácil e que permite que o desenvolvedor personalize a aparência dos componentes.

2.5 PRIMEFACES

O *PrimeFaces* é um *framework* para projetos JSF que pode ser usado para desenvolver aplicações sofisticadas para empresas, ou no desenvolvimento de websites, ou seja, trata-se de uma biblioteca de componentes de interface gráfica para as aplicações web baseadas em JSF. Esse *framework* possui como características a flexibilidade e o fato de ser personalizável, com uma grande opção de componentes, para os mais diversos fins (DE ANDADE, 2016). Ainda, o autor ressalta que é muito comum JSF e *PrimeFaces* estar na lista de pré-requisitos de boas vagas de emprego para programadores Java.

2.6 JPA

JPA é uma especificação de persistência da plataforma Java, para persistir às classes Java em bancos de dados relacionais. Assim, Gonçalves (2011) afirma que a tecnologia JPA foi incluída no JEE, e é uma API (*Application Programming Interface*) que resolve problemas de persistência de dados, reunindo os modelos de orientação a objetos e relacional. Dessa forma, torna possível a independência do SQL (*Structured Query Language*), ao passo que é uma abstração sobre o JDBC

(*Java Database Connectivity*). De acordo com Gonçalves (2011), os principais componentes da API são:

- O mapeamento de objetos para dados como mecanismo para armazenar dados em uma base de dados relacional;
- Gerenciamento de entidades para realizar operações de criar, ler, atualizar e excluir no banco de dados;
- A linguagem JPQL (*Java Persistence Query Language*) que realiza recuperação de dados com consultas orientadas por objetos;
- Controle de acesso concorrente com a JTA (*Java Transaction API*) por meio de mecanismos de transações e bloqueio;
- Inclusão de lógica funcional no ciclo de vida de um objeto persistente com uso de *callbacks* e escutadores.

2.7 HIBERNATE

Segundo Gonçalves (2011), o Hibernate é audacioso, pois tem o objetivo de fazer o relacionamento chamado mapeamento objeto/relacional com o banco de dados, dessa forma, o desenvolvedor abstrai o modelo relacional e preocupa-se apenas com a lógica do negócio. É necessário seguir algumas regras para persistir suas classes, a fim de que uma mudança de banco de dados seja facilmente implementada alterando apenas algumas configurações do *Hibernate*.

Mapeamento objeto/relacional (ORM) refere-se à técnica de mapeamento de uma representação de dados em um modelo de objetos para um modelo de dados relacional baseado em um esquema Entidade Relacionamento (ER). O *Hibernate* não cuida somente do mapeamento das classes Java para tabelas do banco de dados (e dos tipos de dados Java para os tipos de dados SQL), mas também provê facilidades para consultar e retornar os dados da consulta, e pode reduzir significativamente o tempo de desenvolvimento em contrapartida ao alto tempo gasto pelas operações manuais dos dados feitos com SQL e JDBC. O projeto *Hibernate* é composto por vários pacotes Java. Cada pacote tem uma funcionalidade específica e alguns deles só são disponíveis a partir da versão 5.0 do JSE e do JEE (PRIMO, 2011).

Na Figura 4 é possível visualizar um breve resumo sobre pacotes que compõem o *Hibernate*.

Figura 4 - Resumo Pacotes Hibernate

Pacote	Descrição
Hibernate Core	Hibernate para Java, APIs nativas e metadados XML
Hibernate Annotations	Mapeia as classes com anotações do JDK 5.0
Hibernate EntityManager	API de persistência Java padrão para Java SE e Java EE
Hibernate Shards	Framework para particionamento horizontal de dados
Hibernate Validator	API para anotação e validação da integridade dos dados
Hibernate Search	Integração do Hibernate com o Lucene para indexação e consulta a dados.
	"

Fonte: De Andade, 2016

2.8 EJB

EJB é uma sigla de *Enterprise Java Beans*, especificação de componentes para construção modular de aplicações. A especificação padroniza a implementação da regra de negócio das aplicações.

Segundo Dalton (2013), *Enterprise Java Bean* (EJB) é a tecnologia Java padronizada para a criação de componentes em uma arquitetura distribuída, escalável, confiável e robusta. Em uma aplicação com várias camadas com esse tipo de necessidade, geralmente é utilizado EJBs para as camadas de negócios

Para executar uma aplicação que emprega EJBs é necessário um servidor de aplicação, o qual é chamado de contêiner

Segundo Sriganesh et al. (2006, p. 10) EJB é um padrão para desenvolvimento e instalação de componentes de servidor distribuídos em Java. Ele define um acordo (contrato) entre componentes e servidores de aplicação que possibilita qualquer componente ser executado em qualquer servidor de aplicação compatível.

3 DESENVOLVIMENTO DO SISTEMA

Este capítulo apresenta o sistema proposto, bem como o desenvolvimento de cada uma das fases para implementação deste, desde o projeto, contendo os requisitos e modelagens do sistema, até a interface final e funcionamento do sistema, contendo o fluxo de utilização.

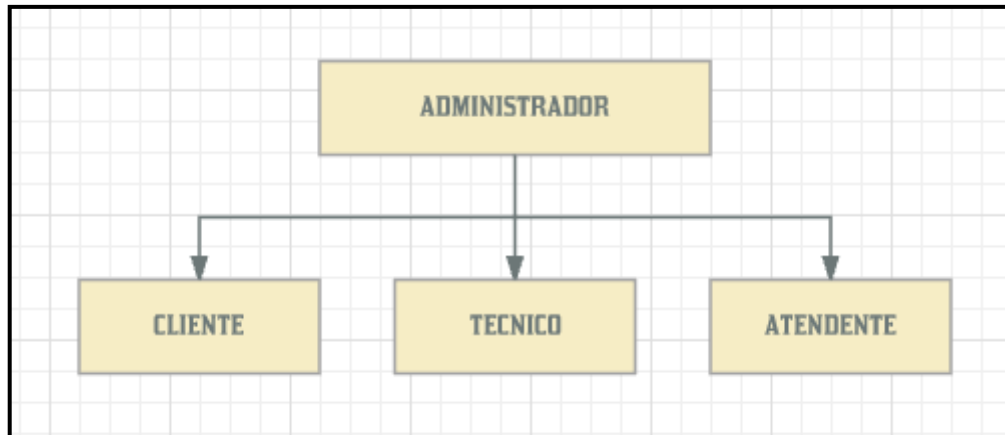
3.1 LEVANTAMENTO DE REQUISITOS

Este documento busca descrever o problema identificado neste estudo e especificar os requisitos necessários para a solução encontrada. Essa solução foca no desenvolvimento de um SI para cadastro e gerenciamento de OS que deve ser construído com base nas informações que serão descritas em breve. O objetivo desta pesquisa é gerar um sistema que seja capaz de atender a todas as necessidades do cliente, ao garantir que esse gerenciamento seja seguro e eficiente. As especificações de requisitos têm como objetivo resumir os requisitos funcionais e não-funcionais necessários para o desenvolvimento do sistema.

3.1.1 Requisitos funcionais

- O sistema deverá ter opções de acordo com o tipo de categoria do usuário. As categorias são definidas de acordo com o perfil do usuário, sendo eles administrador, cliente, técnico e atendente, conforme Figura 5. Sendo assim, o administrador do sistema cadastra a pessoa fornecendo a ela um login e senha;
- Todos os cadastros deverão ter paginação e busca;
- O sistema deve oferecer telas apropriadas para o usuário ler os dados armazenados; e
- O usuário, dependendo da categoria, terá um tipo de acesso a determinadas áreas do sistema, sendo específico de cada um.

Figura 5 - Tipos de usuários do sistema



Fonte: Do autor.

3.1.2 Requisitos não funcionais

- O sistema será acessado tanto em um navegador web quanto em um celular ou *tablet* e deverá ter o mesmo desempenho;
- O sistema deve ter um login e senha para o usuário acessá-lo;
- Linguagem de programação Java;
- O sistema deve ser confiável, devendo atender as suas especificações com sucesso;
- O software deve ser utilizado em qualquer plataforma;
- O sistema deve tratar acessos não autorizados, garantindo alto grau de segurança e, ainda, controlar o acesso às funcionalidades através de grupos de administradores e demais usuários;
- A persistência das informações deve ser implementada em um Sistema Gerenciador de Bancos de Dados Relacional (SGBDR) livre (PostgreSQL).

3.2 PROJETO

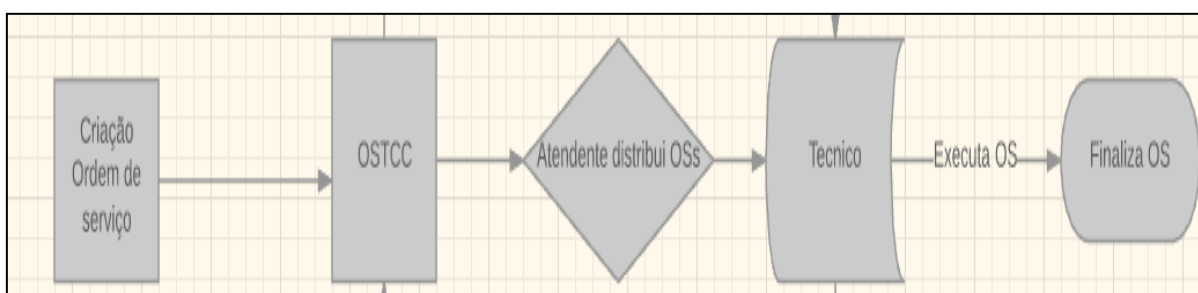
O presente projeto visa desenvolver um sistema genérico para gerência de OS. Para o desenvolvimento do sistema foi realizada uma análise de requisitos e projetado o sistema.

Inicialmente foi realizado um estudo teórico, por meio de pesquisa bibliográfica, sobre ordens de serviços e *websites* responsivos. Esse estudo embasou algumas decisões no desenvolvimento, tal como a necessidade de o sistema ser utilizado em diferentes dispositivos, seja em *smartphone* ou em um computador pessoal. Para isso o sistema foi desenvolvido usando a linguagem de programação Java, sob a plataforma JEE, em que para tornar o sistema responsivo optou-se pelo uso do framework *Bootsface* e *Primefaces*. Com essas duas bibliotecas podemos fazer sites ricos e bonitos ao mesmo tempo. São fáceis de mexer e geram resultados incríveis. Para armazenamento dos dados foi utilizado o banco de dados PostgreSQL por ser robusto e *opensource*. Também foi utilizado *hibernate* para mapeamento de classe abstraindo o SQL junto com o JPA que faz a persistência dos dados.

3.3 MODELAGEM DO SISTEMA

A OS vai ser aberta por um usuário, seja ele cliente, técnico ou um atendente. A partir desta abertura o atendente vai definir para qual técnico direcionar. Após direcionado o técnico poderá executar e finalizar o serviço. A Figura 6 demonstra o diagrama que representa o fluxo do processo de abertura e encerramento de OS.

Figura 6 - Processo de abertura e encerramento de OS



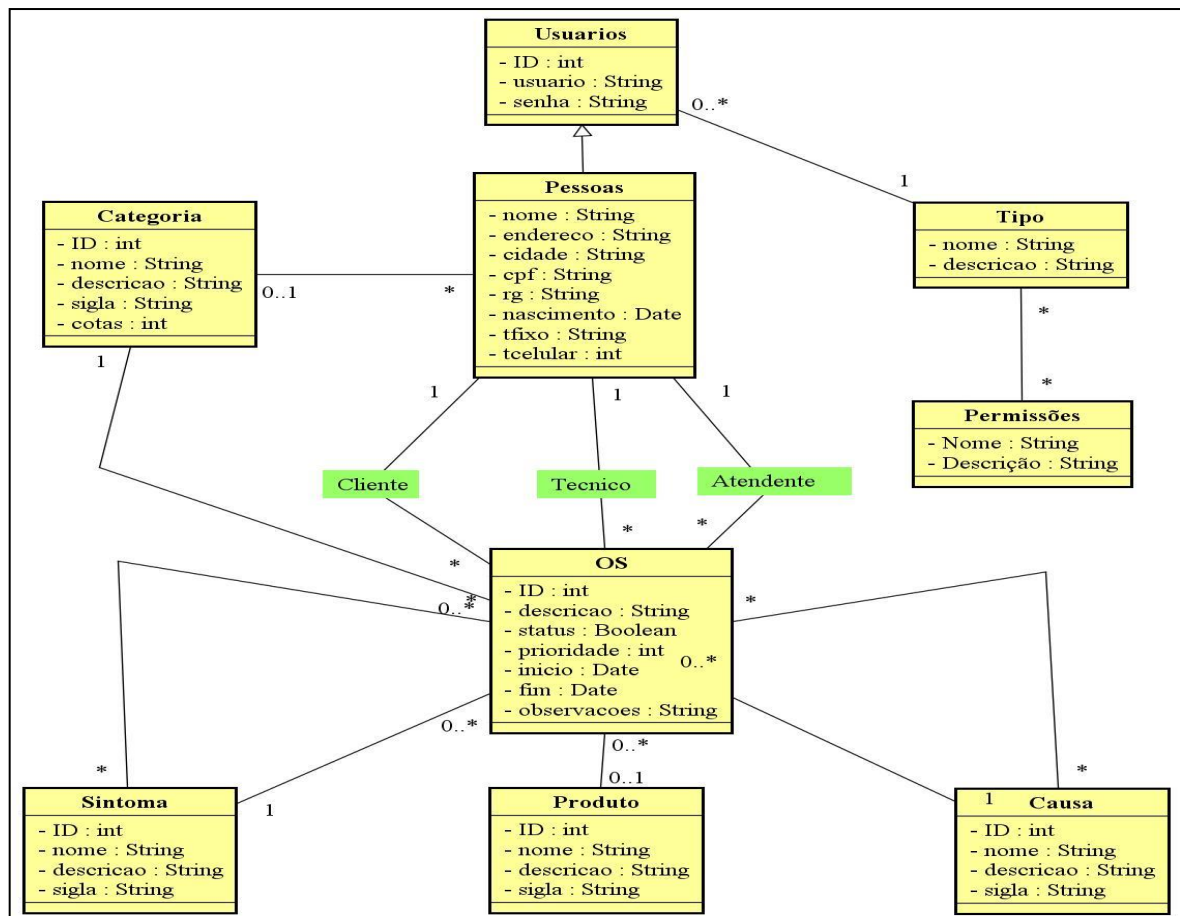
Fonte: Do autor.

O diagrama de classe, disposto na Figura 7, demonstra a organização de classes do projeto, facilitando assim o entendimento de como o sistema foi implementado. Assim, pode-se verificar que o sistema de OS conta com uma classe “Usuarios”, a qual modela os dados de usuário e senha de cada usuário autorizado a utilizar o sistema. A classe “Pessoas” modela as informações gerais de cada tipo de usuário, além de herdar informações de “Usuarios”. A classe “OS” modela o

funcionamento do sistema em si, a qual contém informação dos três tipos de usuários relacionados a OS (o cliente, o técnico e o atendente), além das informações relacionadas ao problema em si, como: causas, sintomas e o produto. As informações de causas e sintomas da OS permitem registrar qual é a causa e o sintoma principal.

A partir dessa modelagem é possível vincular técnicos por categorias, possibilitando o envio de OS somente a técnicos especialistas por determinado problema e produto. Além disso, é possível o controle da produtividade dos serviços executados, o qual é realizado por meio de um registro de cota por categoria, possibilitando a verificação dos serviços realizados e análises futuras utilizando relatórios, ou até mesmo por alertas sobre os técnicos que não atingiram a cota.

Figura 7 – Diagrama de Classes do Sistema de Gerenciamento de OS

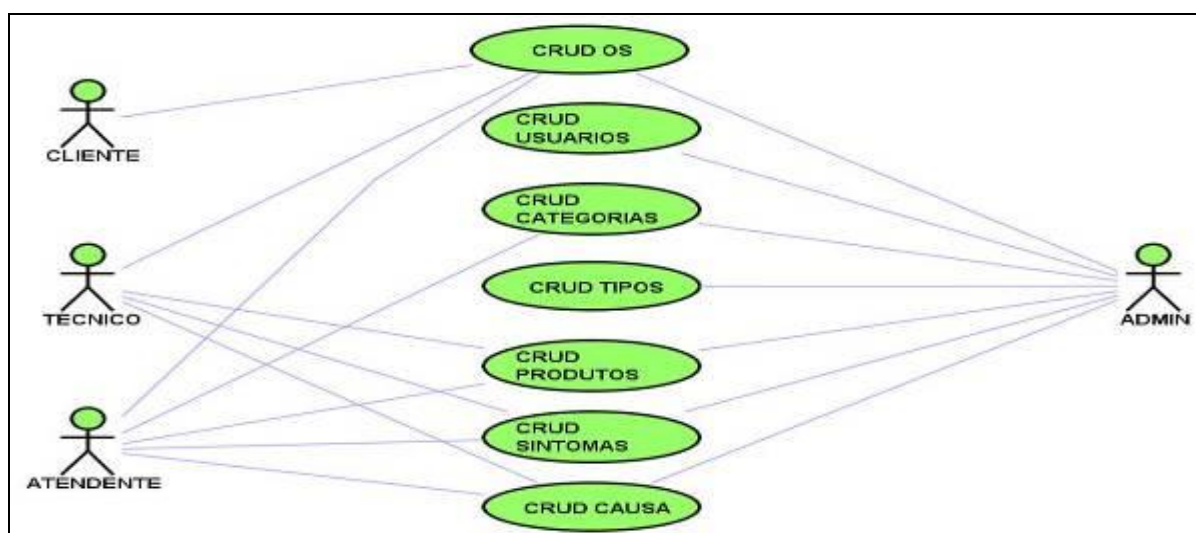


Fonte: Do autor.

O diagrama de caso de uso encontra-se ilustrado na Figura 8, em que são apresentadas as ações realizadas no sistema, demonstrando as funcionalidades e

interações que cada tipo de usuário e cada perfil diferente realizará. Assim, verifica-se que o administrador (Admin) do sistema possui acesso total ao sistema, mas sua principal função é definir categorias e cotas dos técnicos, além de realizar o cadastro e futuras manutenções dos usuários. O Atendente é responsável por realizar os cadastros e modificações de usuários, categorias, sintomas, causas e produtos, além da inserção dos chamados (OS), realizando o gerenciamento das informações. O Cliente possui acesso à consulta de suas OS's, permitindo realizar a abertura de uma OS, contendo restrição na escolha do técnico responsável. O Técnico também possui acesso às OS's que irá executar e poderá criar novos sintomas, produtos e causas. A exclusão de qualquer dado do sistema só será permitida por um usuário administrador.

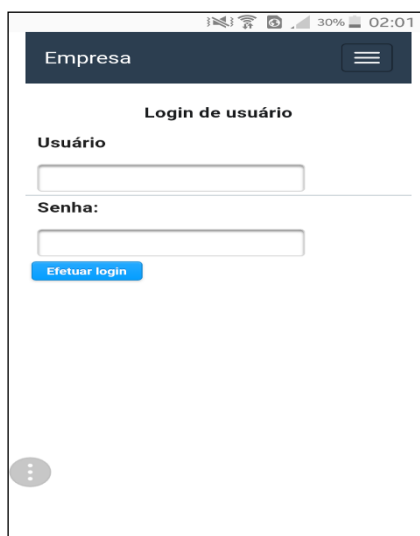
Figura 8 – Diagrama de Caso de Uso do Sistema de Gerenciamento de OS



Fonte: Do autor.

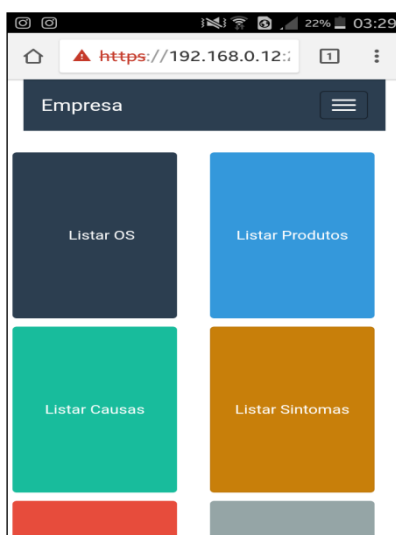
3.4 TELAS

Nesta seção são apresentadas as telas que compõe o sistema. Assim, a Figura 9 ilustra o funcionamento da tela de login do programa. Nela é necessário o usuário fazer a autenticação para habilitar o uso do aplicativo. Será necessário seu cadastro pelo administrador do sistema.

Figura 9 -Tela login

Fonte: Do autor.

Após sua autenticação, o usuário será direcionado à tela inicial, ilustrada na Figura 10, em que cada tipo de usuário terá acesso somente as suas atribuições conforme o seu perfil. Essa tela possui botões grandes para rápido acesso às consultas do sistema, os quais foram desenvolvidos pensando nos acessos por dispositivos móveis.

Figura 10 - Tela inicial

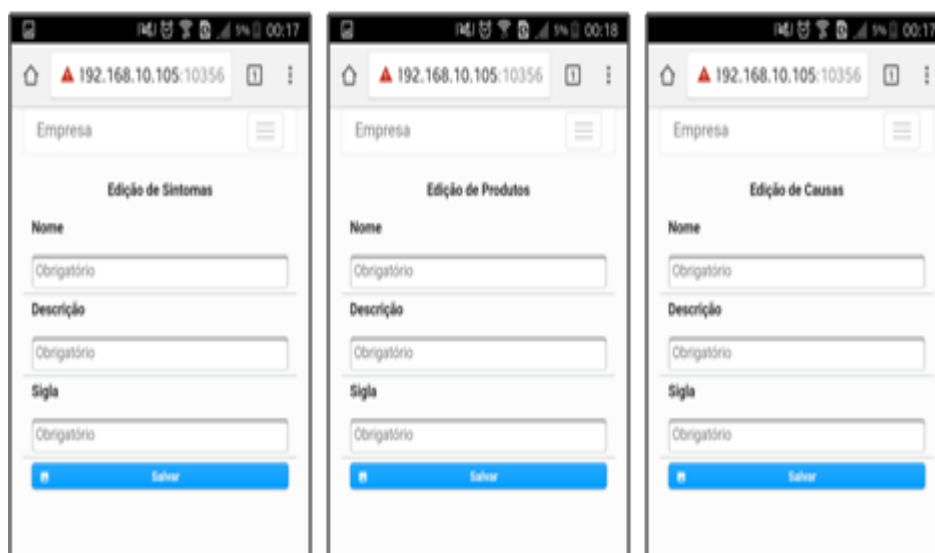
Fonte: Do autor.

O cadastro de OS, realizado pelo atendente, possibilita a seleção de todos os componentes previamente cadastrados no sistema, como: Produto, Cliente, Técnico,

Causa e Sintoma. No caso de cadastro pelo cliente, alguns campos estarão indisponíveis para edição, como técnico, causa e sintoma.

Os itens de Causa e Sintoma possibilitam o registro de uma base de dados para consulta rápida aos problemas e às soluções correspondentes, trazendo agilidade para a resolução do problema, mesmo quando essa não for conhecida pelo técnico, mas outro técnico já tenha resolvido problemas semelhantes. Assim, a Figura 11 apresenta o cadastro de causas, sintomas e produtos sendo necessário informar o nome, descrição e sigla para cada uma delas. Esses itens são usados para preencher a OS.

Figura 11 - Cadastro de Sintoma, Produto e Causa.



The image displays three side-by-side screenshots of a mobile application interface. Each screenshot shows a form for editing a specific item: 'Edição de Sintomas' (left), 'Edição de Produtos' (middle), and 'Edição de Causas' (right). At the top of each screen is a header bar with a home icon, a red triangle warning icon, and the IP address '192.168.10.105:10356'. Below the header is a dropdown menu labeled 'Empresa'. The main content area of each form contains three input fields: 'Nome', 'Descrição', and 'Sigla'. Each of these fields has the word 'Obrigatório' (required) written below it. At the bottom of each form is a blue button with a white arrow and the text 'Salvar' (Save).

Fonte: Do autor.

Na Figura 12 é apresentada a tela de cadastro de permissões, que é responsável pela parte do sistema de liberação de acesso aos conteúdos, ou seja, os papéis que cada tipo de usuário terá no sistema. Contudo, só terá acesso a essa funcionalidade os usuários do tipo Administrador.

Figura 12 - Cadastro de Permissões

Empres

Edição de Permissões

Nome

Obrigatório

Descrição

Obrigatório

Salvar

Fonte: Do autor.

Desse modo, para cadastrar um usuário é necessário cadastrar seu perfil (tipo). Além disso, é necessário informar, também, os dados de acesso ao sistema, como usuário e senha, conforme ilustrado na Figura 13.

Figura 5 - Cadastro de Usuários

Empres

Inserir de Usuários

Tipo

Selecione um registro

Nome de usuário

Obrigatório

Senha

Obrigatório

Nome *

Obrigatório

Endereço *

Obrigatório

Cidade *

Obrigatório

CPF *

RG *

806658914

Email *

fabio.vedoy@hotmail.com

Nascimento *

20/10/1984

Telefone Fixo

Não Obrigatório

Telefone Celular *

(54)99116-9397

Salvar

+ Nova permissão

Permissões do Usuário

Nome	ADMINISTRADOR
Descrição	Acesso Total
Ações	

Fonte: Do autor.

Já em relação à parte de abertura de um chamado, ou seja, do cadastro de uma OS, os diferentes tipos de usuário terão acessos diferenciados, em que o cliente poderá criar uma nova OS, além de ter acesso às ordens abertas em seu nome e o status. O técnico terá acesso às OSs que foram atribuídas a ele, contendo duas listagens: uma de OSs abertas e outra com todas, finalizadas ou não. O atendente por sua vez terá acesso à criação e também será responsável por destinar as OSs ao técnico responsável. Assim, na OS estará descrita a data de início do serviço e qual o serviço que será executado, quem realizou a abertura do chamado, o produto, o sintoma e a causa. A figura 14 mostra o formulário de cadastro da OS.

Figura 14 - Cadastro de OS



A imagem mostra a interface de um navegador móvel acessando um sistema web. No topo, há uma barra de endereço com o URL "192.168.10.105:10356" e o horário "01:42". O formulário principal contém os seguintes campos:

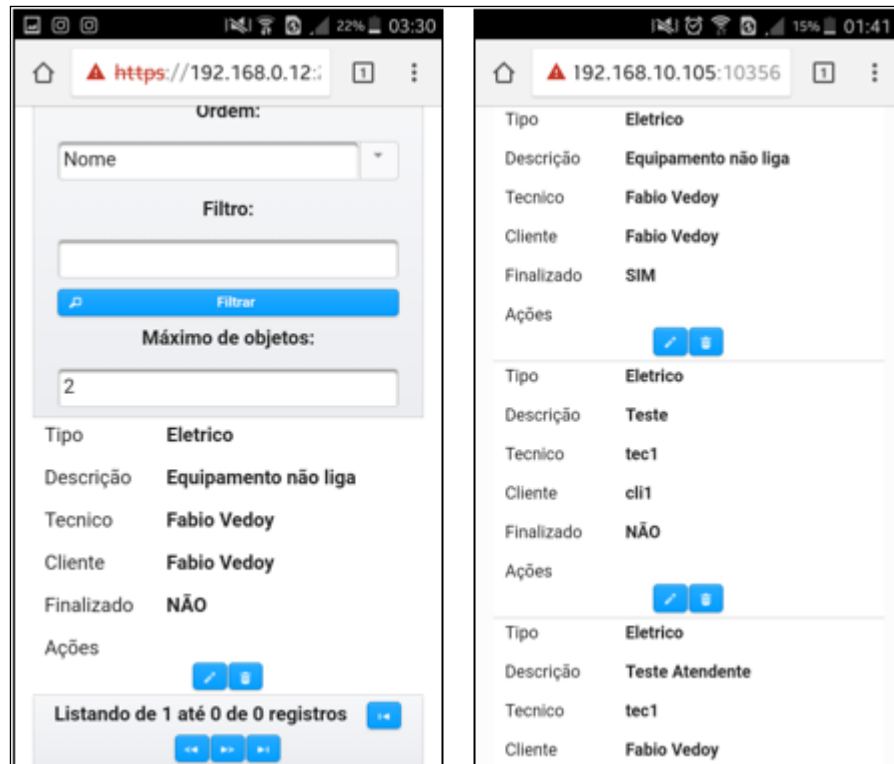
- Descrição:** Campo de texto com o valor "Obrigatório".
- Tipo OS:** Menu suspenso com o texto "Selecione um registro".
- Prioridade:** Menu suspenso com o texto "Selecione um registro".
- Início *:** Campo de texto com o valor "Obrigatório".
- Cliente:** Menu suspenso com o texto "Selecione um registro".
- Tecnico:** Menu suspenso com o texto "Selecione um registro".
- Produto:** Menu suspenso com o texto "Selecione um registro".

Fonte: Do autor.

No formulário esta os principais dados de cadastro de uma OS, conforme o tipo de usuário, o qual na Figura apresenta o acesso do usuário do tipo Administrador, que tem acesso a todos os campos. Assim, alguns campos não aparecerão para determinados tipos de usuários e outros campos serão inseridos automaticamente, conforme regra de negócio do sistema, em que a data de encerramento da OS e o usuário que fez a abertura da mesma são registradas

automaticamente de acordo com a data do sistema e o usuário logado. Vale ressaltar que após o usuário atendente designar o técnico na OS um e-mail será enviado para o cliente informando os dados dela e o andamento, além de permitir a visualização de suas OS no sistema, em que na listagem aparecerá o seu status, conforme Figura 15.

Figura 15 - Listagem de OS



Fonte: Do autor.

4 CONSIDERAÇÕES FINAIS

Tendo em vista a demanda por automatizar os sistemas de OS, baseado nas tecnologias atuais, o presente trabalho teve como objetivo o desenvolvimento de um sistema genérico de gerenciamento de OS com *layout* responsivo, que pudesse ser utilizado em diferentes dispositivos, como celular e computador.

Verifica-se que o objetivo principal do projeto foi alcançado (desenvolvimento do sistema), em que o sistema já pode ser implantado para uso em empresas. Foram implementadas as funcionalidades básicas para que um sistema possa ser utilizado, auxiliando no processo de gerenciamento e decisões dentro de uma prestadora de serviços.

Contudo, algumas funcionalidades, como o controle de produtividade, não foram implementadas e também não é possível realizar a validação e a comparação com outros programas devido alguns atrasos no desenvolvimento. Ao realizar este trabalho, foi possível inferir a real necessidade e vantagens acerca da utilização de um sistema de gerenciamento, quanto ao tempo de execução das tarefas, quanto em relação aos desperdícios que são evitados, ou amenizados. Dessa forma, com o sistema instalado e em funcionamento, os funcionários podem trabalhar com maior segurança, sendo que o sistema executa muitas tarefas automaticamente. Uma das funcionalidades do sistema é de o cliente consultar suas OS sem sair de casa, ou mesmo sem a necessidade de ligar para o prestador de serviços, sendo necessário apenas possuir um cadastro ativo. Também irá receber um e-mail das informações sempre que a OS for alterada.

O sistema foi desenvolvido para que pudesse ser utilizado por qualquer prestadora de serviços independentemente de sua categoria, tratando-se de um sistema genérico.

Algumas dificuldades de desenvolvimento ao longo do tempo de desenvolvimento foram percebidas. Uma das principais a serem citadas é a falta de material para consulta sobre a tecnologia utilizada. Além disso, outra dificuldade que também deve ser citada, é que ao precisar criar algumas funcionalidades, foi necessária a realização de implementações próprias, não sendo possível utilizar funções prontas, o que demandou esforço e tempo de pesquisa e programação. Tal fato acabou implicando no processo de validação, pois o sistema foi concluído sem

tempo de implantar em uma empresa para análise e validação. Assim, como trabalho futuro verifica-se a necessidade de implantar a função de controle de produtividade dos técnicos, ou funcionários, que executam as OS, podendo assim ter um gerenciamento mais completo da mão de obra e suas disponibilidades. Além da necessidade de implantar o sistema em uma empresa e acompanhar seu funcionamento como validação.

REFERÊNCIAS

DALTON, Giulian. Java Magazine 74 - **Enterprise Java Beans 3.1**. 2013. Disponível em:

<<http://www.devmedia.com.br/artigo-java-magazine-74-enterprise-java-beans-3-1/15013>>. Acesso em: 01 nov. 2017.

DE ANDRADE, Thiago Farias. **PrimeFaces: Uma visão geral da tecnologia e de mercado**. [São Paulo], [2016].

Disponível em: <<http://blog.algaworks.com/tecnologia-e-mercado-do-primefaces/>>. Acesso em: 31 out. 2017.

EMMENDOERFER, Fernanda. **Estudo para o desenvolvimento de um sistema de ordem de serviço na gestão de pós-vendas para a Spectrum Informática Ltda.**

Florianópolis, 2002. Acesso em:

<<https://repositorio.ufsc.br/bitstream/handle/123456789/111012/CAD0409-M.pdf?sequence=1&isAllowed=y>>. Acesso em: 02 abr. 2016.

FILGUEIRAS, Fellipe. Java: a origem. **Tableless**, [S.l.], 7 abr. 2015. Disponível em: <<http://tableless.com.br/java-origem/>>. Acesso em: 25 mai. 2016.

FIORESI, Cristiano. Conceitos básicos das plataformas Java e J2ME. **Devmedia**, [Rio de Janeiro], [2015]. Disponível em: <<http://www.devmedia.com.br/conceitos-basicos-das-plataformas-java-e-j2me/6484>> Acesso em: 30 abr. 2016.

GOMES, Fabio. Conheça o Bootsface: Bootstrap para Java Server Face. **Devmedia**, [Rio de Janeiro], [2015]. Disponível em: <<http://www.devmedia.com.br/conheca-o-bootsface-bootstrap-para-java-server-face/31070>>. Acesso em: 20 maio 2016.

GONÇALVES, Antonio. **Introdução à Plataforma Java™ EE 6 com GlassFish™ 3**. Rio de Janeiro: Ciência Moderna, 2011.

GUIMARAES, Gleyser. A história da linguagem Java. **Pet news**, 2012. Disponível em: <http://www.dsc.ufcg.edu.br/~pet/jornal/dezembro2012/materias/historia_da_computacao.html>. Acesso em: 01 maio 2016.

KOETZ, Felipe da Silva. **OSSYS sistema de ordem de serviço**. Pelotas, novembro de 2014. Disponível em: <<http://informatica.ucpel.edu.br/uploads/tccs/2014/2/0002.pdf>>. Acesso em: 05 abr. 2016.

LAUDON, Kenneth C., LAUDON, Jane P. **Sistemas de informação**. 4. ed. Rio de Janeiro: LTC, 1999.

MÉDICE, Roney. A importância da segurança da informação: visão corporativa. Profissionais TI, [S.l.], 12 jul. 2013. Disponível em:

<<https://www.profissionaisiti.com.br/2013/07/a-importancia-da-seguranca-da-informacao-visao-corporativa/>>. Acesso em: 12 abr. 2016.

MEIO & MENSAGEM. **CELULAR é o principal meio de acesso à web**. São Paulo, 7 abr. 2016. Disponível em: <http://www.meioemensagem.com.br/home/midia/2016/04/07/celular-e-o-principal-meio-de-acesso-a-web.html?fb_action_ids=1254957667864985&fb_action_types=og.likes>. Acesso em: 30 mai. 2016.

ORACLE. **Java Developer Center**. São Paulo, 2014. Disponível em: <<http://www.oracle.com/technetwork/pt/java/index.html>>. Acesso em: 06 mai. 2016.

PACIEVITCH, Yuri. História do Java. **InfoEscola**, [S.l.], 2016. Disponível em: <<http://www.infoescola.com/informatica/historia-do-java/>> Acesso em: 22 abr. 2016.

PALMEIRA, Thiago Vinícius Varallo. Java: História e Principais Conceitos. <<http://www.devmedia.com.br/java-historia-e-principais-conceitos/25178#ixzz2T0UyDXKG>> Acesso em: 01 jun. 2016.

PAMPLONA, Vitor Fernando. Tutorial Java: o que é Java? **Javafree**, São Paulo, 26 março 2015. Disponível em: <<http://javafree.uol.com.br/artigo/871498/Tutorial-Java-O-que-e-Java.html>> Acesso em: 23 mai. 2016.

PITANGA, Talita. JavaServer Faces: a mais nova tecnologia Java para desenvolvimento WEB. **Grupo de usuários Java**, [S.l.], 30 jul. 2009. Disponível em: <<http://www.cin.ufpe.br/~jvwr/JSF/jsf.pdf>> Acesso em: 01 jun. 2016.

PRIMO, Izalmo. **Desenvolvendo com Hibernate**. [Rio de Janeiro], [2011]. Disponível em: <<http://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>>. Acesso em: 31 out. 2017.

RENE, Thierry. **Introdução ao Bootstrap para iniciantes**, [São Paulo], [2016]. Disponível em: <<http://blog.algaworks.com/tecnologia-e-mercado-do-primefaces/>>. Acesso em: 31 out. 2017.

SRIGANESH, P. et al. **Mastering Enterprise Java Beans 3.0**, Quarta Edição, Indianápolis: Wiley, 2006.

VINICIUS, Thiago. Java: história e principais conceitos. **Devmedia**, [Rio de Janeiro], [2015]. Disponível em: <<http://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>>. Acesso em: 15 abr. 2016.