

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE LICENCIATURA EM INFORMÁTICA
DESENVOLVIMENTO DE SISTEMAS PARA INTERNET E DISPOSITIVOS
MÓVEIS

JONAS CENTENARO

DESENVOLVIMENTO DE UM SOFTWARE WEB PARA
GERENCIAMENTO DE REQUISITOS DE SOFTWARE

MONOGRAFIA DE ESPECIALIZAÇÃO

FRANCISCO BELTRÃO

2014

JONAS CENTENARO

**DESENVOLVIMENTO DE UM SOFTWARE WEB PARA
GERENCIAMENTO DE REQUISITOS DE SOFTWARE**

Monografia de Especialização apresentada a Coordenação de Licenciatura em Informática, da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Especialista em Desenvolvimento de Sistemas para Internet e Dispositivos Móveis”.

Orientador: Prof. Me. Paulo Júnior Varela.

FRANCISCO BELTRÃO

2014

TERMO DE APROVAÇÃO

Dia 27 do mês de novembro de 2014 às: 16 horas, na sala **Q208** do Câmpus Francisco Beltrão, realizou-se a apresentação pública da monografia pelo estudante **Jonas Centenaro**, intitulada “**DESENVOLVIMENTO DE UM SOFTWARE WEB PARA GERENCIAMENTO DE REQUISITOS DE SOFTWARE**” Finalizada a apresentação e argüição, a Banca Examinadora declarou **aprovada** a monografia do estudante, requisito parcial para obtenção do título de Especialista em Desenvolvimento e Sistemas para Internet e Dispositivos Móveis.

Professor **Paulo Júnior Varela**– UTFPR
(ORIENTADOR)

Professor **Edson dos Santos Cordeiro**- UTFPR
(Convidado)

Professor **Marcos Mincov Tenório**- UTFPR
(Convidado)

Professor **Dr. Ademir Roberto Freddo** – UTFPR
(COORDENAÇÃO)

A folha de aprovação com as rubricas encontram-se disponíveis na Diretoria de Pesquisa e Pós-Graduação, UTFPR, Francisco Beltrão.

AGRADECIMENTOS

Primeiramente agradeço a minha família que sempre me apoiou no crescimento escolar.

Aos professores e amigos que com suas competências contribuíram e muito no decorrer do desenvolvimento deste trabalho.

Agradeço especialmente ao orientador Paulo Júnior Varela, pelas idéias e tempo disposto para avaliação e evolução no decorrer da confecção deste trabalho.

RESUMO

As empresas de desenvolvimento de software em seu alto nível de produção acabam gerando documentos de especificações de software para manter uma organização no processo de desenvolvimento. Esses documentos descrevem basicamente o funcionamento e as características de um sistema. O maior problema está relacionado à identificação de responsabilidades de cada envolvido. Para elaboração desse projeto foram utilizadas técnicas qualitativas, de estudo de caso e pesquisa bibliográfica, tendo como objetivo principal a criação de um sistema capaz de administrar a leitura e atualização de documentos de especificações de software. Através desta solução é possível ter um controle mais amplo dos requisitos de software, e assim proporcionar maiores níveis de satisfação final por todos os envolvidos. Diante disso, o presente trabalho resultou no desenvolvimento de uma aplicação para o gerenciamento de documentos de especificação de software, e possibilitou dessa forma aplicar as técnicas, linguagens de programação e algumas ferramentas para melhorar a usabilidade, como o caso da utilização do *framework Bootstrap*, que além de fornecer componentes ricos em detalhes, também disponibiliza recursos que adaptam a interface de acordo com as dimensões da tela do dispositivo utilizado.

Palavras-Chave: Sistema. Gestão. Requisitos. Software.

ABSTRACT

Companies developing software for its high level of production end up generating software specifications documents to keep an organization in the development process. These documents primarily describe the operation and characteristics of a system. The biggest problem is related to identification responsibilities of each stakeholder. For elaboration of this project were used qualitative techniques, case study and bibliographic research, having as main objective the creation of a system able to manage reading and updating of software specifications documents. With this solution it is possible to have a broader control of software requirements, and thus provide higher levels of ultimate satisfaction for all stakeholders. Thus, the present work resulted in the development of an application for managing software specification documents, and thus enabled to apply the techniques, programming languages and tools to improve usability, as the case of use of Bootstrap framework, which besides providing components rich in detail, also provides features that interface adapted according to the dimensions of the display device used.

Keywords: System. Management. Requirements. Software.

LISTA DE FIGURAS

Figura 1 - Visão geral do sistema Notepad++.....	25
Figura 2 - Visão geral do sistema Google Chrome.....	26
Figura 3 - Visão geral do sistema Eclipse com Android SDK.....	27
Figura 4 - Visão geral do sistema Astah Community.....	28
Figura 5 - Interface principal do Rational RequisitePro	29
Figura 6 - Modelagem de XP usando diagrama de atividades.....	32
Figura 7 - Componentes de interface utilizando framework Bootstrap.....	37
Figura 8 - Diagrama de caso de uso geral da aplicação web.....	42
Figura 9 - Diagrama de classe principal da aplicação web.	43
Figura 10 - Diagrama de entidade e relacionamento.	44
Figura 11 - Diagrama de visão geral da interação do sistema móvel.....	46
Figura 12 - Diagrama de classe principal da aplicação Android.....	46
Figura 13 - Exportar Banco de dados para Arquivo SQL.	48
Figura 14 - Recurso de Inspeccionar Elemento do Google Chrome.....	49
Figura 15 - Tela de login com "Esqueci meu acesso?" expandido.....	50
Figura 16 - E-mail de recuperação de acesso ao sistema Geres.....	51
Figura 17 - Página inicial do sistema.....	52
Figura 18 - Itens de menu Cadastros para usuários Administradores.....	53
Figura 19 - Itens de menu Cadastros para usuários Analistas.....	53
Figura 20 - Itens de menu Cadastros para usuários Leitores.....	53
Figura 21 - Tela de cadastro e manutenção de perfil de usuário.	54
Figura 22 - Interface de Mensagens Recebidas em ordem de recebimento.....	54
Figura 23 - Interface de Mensagens Enviadas em ordem de envio.	55
Figura 24 - Interface de listagem de reservas.	55
Figura 25 - Encerrando sessão.	56
Figura 26 – Interface de listagem de usuários.....	56
Figura 27 - Interface de cadastro de ator.	57
Figura 28 - Interface de listagem de atores.	58
Figura 29 - Interface de cadastro de projeto.....	58
Figura 30 - Interface de listagem de projetos.	59
Figura 31 - Interface de cadastro de pasta.....	59
Figura 32 - Interface de listagem de pastas e documentos.....	60

Figura 33 - Interface de confirmação para cadastro de novo documento de requisitos.	60
Figura 34 - Resultado de busca pelo campo de pesquisa.....	61
Figura 35 - Interface de documento de especificação de software.	61
Figura 36 - Interface do documento em modo de edição.	62
Figura 37 - Opção de reservar documento.....	62
Figura 38 - Identificador de local do documento.....	62
Figura 39 - Guia dos tópicos do documento.....	63
Figura 40 - Tópico de propriedades básicas.	63
Figura 41 - Interface para alteração de pasta do documento.....	64
Figura 42 - Tópico de capa do documento.	64
Figura 43 - Tópico de histórico de revisões.....	65
Figura 44 - Tópico de nome breve descrição.	65
Figura 45 - Tópico de pré-condições.	65
Figura 46 - Tópico de pós-condições.	66
Figura 47 - Tópico de atores.	66
Figura 48 - Tópico de fluxo de eventos.	66
Figura 49 - Tópico de requisitos.	67
Figura 50 - Tópico de requisitos cancelados.....	67
Figura 51 - Tópico de diagramas.....	67
Figura 52 - Tópico de protótipos de interface.....	68
Figura 53 - Tópico de referências.....	68
Figura 54 - Interface AVD - Android Virtual Device.	70
Figura 55 - Tela principal do sistema em um dispositivo móvel.	71
Figura 56 - Tela com menu expandido em um dispositivo móvel.....	71
Figura 57 - Guias de documento em um dispositivo móvel.....	72
Figura 58 - Parte do corpo de documento em um dispositivo móvel.....	72

LISTA DE GRÁFICOS

Gráfico 1 - Estatística de uso de linguagens em servidores web	35
--	----

LISTA DE QUADROS

Quadro 1 - Requisito Aplicação web: Manter Perfil de Usuário.....	38
Quadro 2 - Requisito Aplicação web: Manter Mensagens Recebidas.....	38
Quadro 3 - Requisito Aplicação web: Manter Mensagens Enviadas.....	39
Quadro 4 - Requisito Aplicação web: Apresentar Itens Reservados.....	39
Quadro 5 - Requisito Aplicação web: Manter Cadastro de Usuários.....	39
Quadro 6 - Requisito Aplicação web: Manter Cadastro de Atores.....	39
Quadro 7 - Requisito Aplicação web: Manter Cadastro de Projetos.....	40
Quadro 8 - Requisito Aplicação web: Manter Cadastro de Pastas.....	40
Quadro 9 - Requisito Aplicação web: Manter Cadastro de Documentos.....	41
Quadro 10 - Requisito Aplicação web: Aplicar ferramenta Bootstrap.....	41
Quadro 11 - Requisito Aplicação web: Aplicar linguagens para multiplataforma.....	41
Quadro 12 - Tabela docs: Responsável por registrar os documentos e controlar as reservas dos mesmos.....	44
Quadro 13 - Requisito Aplicação móvel: Comunicação com a Internet.....	45
Quadro 14 - Requisito Aplicação móvel: Interface navegador para Internet.....	45
Quadro 15 - Requisito Aplicação móvel: Definições de configuração.....	45

LISTA DE TABELAS

Tabela 1 - Sistemas operacionais, Crescimento vendas em milhões	33
Tabela 2 - Comparativo de Sistemas	74

LISTA DE SIGLAS

AVD	<i>Android Virtual Device</i>
CSS	<i>Cascading Style Sheets</i>
DNS	<i>Domain Name System</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JS	<i>JavaScript</i>
MVC	<i>Model View Controller</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Hypertext Preprocessor</i>
SDK	<i>Software Development Kit</i>
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
WAMP	Windows, Apache, MySQL, PHP
XML	<i>Extensible Markup Language</i>
XP	<i>Extreme Programming</i>

SUMÁRIO

1.	INTRODUÇÃO.....	14
1.1	Objetivos.....	15
1.1.1	Objetivos Gerais	15
1.1.2	ObjetivosEspecíficos.....	15
1.2	Justificativa	15
1.3	Metodologia	16
1.4	Estrutura do Trabalho	16
2.	REFERENCIAL TEÓRICO	18
2.1	Gerenciamento de requisitos	18
2.2	Versionamento.....	20
2.3	Desenvolvimento de software para web	21
2.4	Desenvolvimento de software para dispositivos móveis.....	22
3.	MATERIAIS E MÉTODOS	24
3.1	Materiais	24
3.2	Métodos	30
3.2.1	Metodologia de desenvolvimento XP.....	30
3.2.2	Android	32
3.2.3	PHP	34
3.2.4	Banco de dados MYSQL	35
3.2.5	BOOTSTRAP.....	36
3.2.6	Modelagem do Software.....	37
4.	RESULTADOS.....	47
4.1	Desenvolvimento do aplicativo Web.....	47
4.1.1	Apresentação do aplicativo Web.....	49
4.2	Desenvolvimento do sistema móvel.....	69
4.2.1	Apresentação do aplicativo Android.....	70
5.	CONCLUSÃO	73
5.1	Problemas Encontrados.....	74
5.2	Trabalhos Futuros.....	75
	REFERÊNCIAS.....	76
	Anexo A – Dicionário de dados.	79
	Anexo B – Listas de códigos desenvolvimento web.....	85

Anexo C – Listas de códigos desenvolvimento móvel.....	92
Anexo D – Exemplo de documento impresso no sistema em ambiente real.....	96

1. INTRODUÇÃO

É comum hoje em dia a percepção de quanto os softwares estão em constante evolução. Como por exemplo pode-se citar a evolução dos sistemas operacionais, gradativamente novas versões são lançadas, o que significa novos recursos, melhor desempenho ou visuais mais atraentes. Fato é que para essas novas versões, as empresas de desenvolvimento de softwares em sua organização efetuam vários planejamentos antes de qualquer alteração ou criação de um recurso. Na fase de planejamento vários documentos são gerados, e um deles é o documento de especificação de software. Esses documentos descrevem basicamente o funcionamento e as características de um sistema.

Porém, a cada nova versão ou alteração dos recursos já existentes, novas edições ou criações desses documentos são realizadas, o que pode acarretar em instabilidades, seja pela parte que a elabora ou pela parte que interpreta o documento, pois este pode conter no mesmo contexto, parte já realizada e parte a realizar.

Diante desse cenário pode ser encontrado como problema a dificuldade da definição de versões, podendo os usuários finais estarem usufruindo um documento em versão não compatível para os mesmos, em casos usuários desenvolvedores acabam realizando tarefas que deveriam ser delegadas a outros desenvolvedores.

Com o propósito de facilitar a identificação das alterações desses documentos, o presente trabalho propõe o desenvolvimento de um sistema *web*, capaz de gerenciar versões de documentos, baseando-se na definição de uma estrutura de um documento de especificação de software.

Para utilização deste aplicativo, o usuário pode utilizar dispositivos como desktops, notebooks, tablets e celulares, com acesso a internet e alguma aplicação com recurso de navegação na internet. Pois o desenvolvimento deste trabalho está relacionado à tecnologia *responsive* (Abordagem de *web* design que fornece uma fácil navegação e visualização de conteúdos *web* em diferentes tamanhos de dispositivos).

1.1 Objetivos

A seguir serão apresentados os objetivos gerais e específicos do trabalho.

1.1.1 Objetivos Gerais

Desenvolver um software web que permita gerenciar versões de documentos de especificações de softwares, permitindo o gerenciamento de edição e leitura.

1.1.2 Objetivos Específicos

Dentre os objetivos específicos do trabalho, destacam-se:

- Realizar a análise da documentação de requisitos, identificando as características essenciais para desenvolver a estrutura básica do software de gerenciamento de versões.
- Aplicar as técnicas e linguagens de programação web e a ferramenta de *Bootstrap* para aperfeiçoar o processo de folha de estilo.
- Melhorar o processo de gerenciamento de documentos de especificação de software.
- Aplicar o software em ambiente real para verificação do comportamento.

1.2 Justificativa

Segundo Blaschek (2002, p.1):

Aos requisitos estão associados os principais problemas do desenvolvimento de software. Requisitos que não refletem as reais necessidades dos usuários, incompletos e/ou inconsistentes, mudanças em requisitos já acordados e a dificuldade para conseguir um entendimento comum entre usuários e desenvolvedores são as principais dificuldades relatadas, provocando retrabalho, atrasos no cronograma, custos ultrapassados e a insatisfação dos clientes e usuários de software.

Diante disso, e como em todo mercado, fábricas de software também se localizam no ambiente de alta competitividade, dentre os agravantes que mantém essas empresas no mercado identificam-se como principais: a qualidade e o tempo de entrega do produto. Sendo os documentos de especificações de software um dos

principais responsáveis pelo resultado final do produto, torna-se indispensável à implantação de um sistema que gerencie esses documentos. Vale salientar também de que um documento de especificação de software mal gerenciado pode causar transtornos como: implementações indevidas, retrabalho, e dúvidas dos *stakeholders* (envolvidos) já que os mesmos compartilham de um mesmo documento em única versão.

1.3 Metodologia

Para a resolução dos problemas encontrados primeiramente foram levantados alguns requisitos para a criação de um novo sistema e como o mesmo poderia ser utilizado no dia-a-dia, em seqüência foi projetado uma idéia de solução, e que por final fora executado e aplicado na organização.

O presente trabalho, do ponto de vista da natureza pode ser classificado como uma pesquisa exploratória aplicada. Do ponto de vista técnico esse trabalho pode ser classificado como um estudo de caso.

As metodologias utilizadas consistem em coletar e analisar informações sobre uma determinada organização, a fim de estudar aspectos relacionados ao tema de gerenciamento de documentos de especificação de software.

Neste método foi buscada a aplicação prática dos conhecimentos para uma solução proposta que resultou no desenvolvimento de um sistema para gerenciamento desses documentos e que será apresentado no decorrer deste trabalho.

1.4 Estrutura do Trabalho

O trabalho está dividido em 5 capítulos. Destes, o primeiro apresenta a introdução, objetivos e justificativa para a realização deste trabalho.

No Capítulo 2 apresenta o referencial teórico sobre os conceitos e características dos documentos de especificação de softwares, e linguagens de programação utilizadas no desenvolvimento do sistema proposto e apresenta também um sistema similar ao proposto pelo presente trabalho.

O Capítulo 3 apresenta os materiais e métodos utilizados para a realização deste trabalho.

O Capítulo 4 apresenta os resultados obtidos através de um comparativo de utilização entre o sistema proposto e o similar.

Finalizando, no capítulo 5 são apresentadas as conclusões, as dificuldades encontradas e as sugestões de trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho, exibindo as características e conceitos dos documentos de especificação de software e exemplo de aplicação similar ao software desenvolvido.

2.1 Gerenciamento de requisitos

Primeiramente entende-se por requisitos as definições de uma propriedade, comportamento ou restrições de operação que um produto ou serviço deve atender (WTHREEX.COM, 2014)

Gerenciamento de requisitos é um modelo sistemático para compreender e controlar as mudanças nos requisitos de sistemas, nesse processo inclui-se identificar, organizar, documentar os requisitos de sistema e estabelecer e manter acordo entre o cliente e a equipe de desenvolvimento (WTHREEX.COM, 2014)

Quando a organização não dispõe deste processo formalmente definido e amplamente divulgado, os desenvolvedores elaboram as especificações de requisitos de uma forma empírica, executando atividades não padronizadas e definidas individualmente. Se isto ocorre, a qualidade da especificação dependerá exclusivamente da experiência e formação das pessoas, havendo assim uma elevada probabilidade de ocorrerem conflitos e re-trabalho (BLASCHEK, 2014)

A utilização de ferramentas que auxiliam no gerenciamento de requisitos de software oferece muitos benefícios, criando um ambiente que promove a colaboração e a responsabilidade.

Organizações que provem o uso de ferramentas de gerenciamento de requisitos de software e que possuem uma documentação de requisitos precisos e bem detalhados conseguem melhor dividir tarefas, estimar com maior precisão, e conseqüentemente adquirem de um tempo hábil para gerenciar riscos e manutenções provenientes das novas alterações.

Em gerenciamento de requisitos de software várias técnicas de levantamento de requisitos estão definidas. Trata-se de um bom levantamento de requisitos aquele que resulta em uma boa definição do projeto, na conclusão efetiva do projeto, na exposição de informações necessárias a um perfeito diagnóstico e de soluções inteligentes. Do contrário, trata-se de um mau levantamento de requisitos aquele que

resulta em diagnósticos insuficientes com conclusões comprometidas, em não identificar as causas dos problemas, em custos elevados, prazos vencidos ou comprometedores, na omissão de processos fundamentais e descrédito. Como afirma Moraes (2014):

“As técnicas de levantamento de requisitos têm por objetivo superar as dificuldades relativas a esta fase. Todas as técnicas possuem um conceito próprio e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista.”.

A seguir são apresentadas as algumas técnicas que podem ser empregadas no levantamento de requisitos (MORAES, 2014):

Entrevistas e Questionários: É uma das técnicas mais simples, mas de grandes resultados principalmente nas fases iniciais do projeto. Para esta técnica é necessário haver um plano de entrevista ou questionário bem definido, para que no momento da entrevista não haja desvios e dispersão do assunto principal, entrevistas longas e cansaço pode acarretar em prejuízos na coleta de requisitos por esta técnica.

Workshops de requisitos: Essa técnica trata-se de uma reunião estruturada. Um grupo de pessoas para debate. Podem fazer parte desse grupo analistas e *stakeholders* que representam a organização e/ou o contexto do sistema em que será debatido. Essas reuniões têm por objetivo acionar o trabalho em equipe, levantar problemas e declarar tomadas de decisões.

Cenários (Série de Eventos Hipotéticos): Essa técnica tem por objetivo a elaboração de cenários que possibilitam os *stakeholders* a imaginarem o funcionamento de um sistema. É necessário o levantamento e apresentação do estado inicial do cenário antes de qualquer interação do usuário. Deve ser relatado e apontado as seqüências de eventos esperados no cenário, mensagens esperadas, e se alguma outra atividade pode ser executada ao mesmo tempo deste cenário.

Prototipagem: Essa técnica tem por objetivo identificar qual a melhor alternativa de interface para o usuário. Esses protótipos podem ser elaborados representando interfaces, relatórios textuais, relatórios gráficos, entre outras. Esses protótipos podem ser apenas imagens genéricas que focam no contexto do requisito, ou parte do produto implementando de forma rápida um pequeno subconjunto de funcionalidades do produto.

Estudo etnográfico: Essa técnica promove a observação do usuário, a fim de identificar requisitos não observáveis nas demais técnicas, essa observação pode

ser acompanhada de áudio/vídeo ou presencial, a utilização dessa técnica assume-se de que o usuário representante da organização desempenha as funções de maneira correta.

Os analistas podem conforme a necessidade utilizar uma ou mais técnicas para capturar os requisitos, conforme análise na organização e a estrutura pré-definida do documento utilizado, os analistas depois de identificado os requisitos, devem documentá-los nos documentos de especificação de software, mantendo juntamente um histórico de revisões, contendo responsável, data, hora, identificador da versão corrente e breve descrição da manutenção do documento, tendo prévios direitos de acessos e alterações a estes documentos, uma vez que qualquer alteração nele condiz com um funcionamento do sistema.

2.2 Versionamento

Versionamento ou sistemas de controle de versão são aplicações que possibilitam o gerenciamento de diferentes versões no desenvolvimento de qualquer documento.

Os sistemas de controle de versão, comumente utilizados no desenvolvimento de software, são ferramentas cujo objetivo é gerenciar as diferentes versões dos documentos produzidos ao longo do processo de desenvolvimento.

De acordo com Murta (2006), sob a perspectiva de desenvolvimento o gerenciamento de configuração de software é dividido em três sistemas principais, que são: controle de modificações, controle de versões e gerenciamento de construção.

Segundo Masson (2006 apud FREITAS, 2010, p. 2), um sistema de controle de versão consiste, basicamente, em um local para armazenamento de artefatos gerados durante o desenvolvimento de sistemas de software.

Atuando como uma espécie de “máquina do tempo” dos desenvolvedores, as ferramentas de controle de versão permitem retornar múltiplos arquivos de múltiplos autores a qualquer versão anterior (ARAUJO, 2011)

Segundo Araujo (2011) as vantagens encontradas nos sistemas que utilizam mecanismos de controle de versão são:

- **Controle de histórico:** Possibilita o usuário analisar versões antigas, exportar versões antigas para possíveis comparações, e recurso de desfazer alterações voltando o documento ao estado em que se identificava nas versões anteriores.
- **Suporte a colaboração:** Possibilita a identificação de alteração de determinado documento. Dependendo do sistema, essa vantagem pode permitir o usuário interessado em alterar o mesmo documento, optar por aguardar a liberação do usuário reservado ao documento em questão, ou trabalhar de forma paralela, e posteriormente os usuários podem efetuar o merge (mesclar informações de dois ou mais documentos a fim de gerar um só).
- **Suporte a marcação e resgate de versões estáveis:** Alguns sistemas de controle de versão possuem propriedades que identificam versões estáveis, possibilitando através do histórico, selecionar uma versão estável para exportar e utilizar.
- **Ramificação de projeto:** Possibilita o trabalho por equipes diferentes em um mesmo produto, tornando obrigatória a execução da atividade de merge após a conclusão do trabalho das equipes.

Nos documentos de especificação de software o emprego desses mecanismos de gerenciamento de versões é essencial. Os requisitos mudam freqüentemente, alguns motivos são: os problemas os quais se refere a um requisito não foram inteiramente definidos, entendimento dos desenvolvedores no decorrer do tempo evolui, melhorias em sistemas antigos ou automatização de algum processo manual é realizado, e modificações das empresas, legislações, regras e ambiente técnico também são fatores para a manutenção nos requisitos de software. Dessa forma, requisitos podem ser alterados, incluídos ou excluídos.

Certo de que estes documentos representam determinada funcionalidade ou propriedade de um sistema, e impõe grandes responsabilidades de quem for alterá-los, fica necessário o rastreamento das mudanças, por conta disso, opções de acesso a leitura de versões anteriores, para o usuário poder comparar versões, são peças fundamentais nos gerenciadores de requisitos de software.

2.3 Desenvolvimento de software para web

Sistemas de informação deixaram de ser algo opcional para as empresas, e se tornaram alvo das maiores necessidades que uma empresa pode possuir, o papel

que esses sistemas possuem em relação à organização é de extrema importância, uma vez que nestes ficam as responsabilidades de manter, e gerar novos dados, que por sua vez podem se caracterizar como um dos maiores patrimônios da empresa.

Dados sigilosos, informações de clientes, informações de produtos e estoque, financeiro, relatórios gerenciais e muitas outras informações ficam a cargo desses sistemas.

A atualidade do mercado esta contida num mundo globalizado, e a internet tornou-se parte integrante desse novo sistema de negociação. Dia a dia cresce o número de usuários, serviços, informações e facilidades disponíveis online. Desenvolvimento de softwares para web diferencia de um desenvolvimento desktop, exatamente no que diz respeito ao público alvo, maneira de acesso e manutenção do mesmo.

A maneira de acesso torna-se mais acessível uma vez que é um sistema web e pode ser acessado através de navegadores de internet e com características multiplataforma já que não obriga o usuário final a possuir determinado sistema operacional, e sim apenas ter um navegador de internet.

E em relação à manutenção, fica mais simples a manutenção e atualização do mesmo já que não necessita o usuário final a executar procedimentos de instalação, facilita o suporte não necessitando diretamente visitas ao usuário final nem assistência remota, pois os arquivos estão armazenados em servidor web e sua manipulação afeta instantaneamente o usuário final.

2.4 Desenvolvimento de software para dispositivos móveis

O desenvolvimento de software possui uma grande importância para a sobrevivência das empresas, mas outro ponto que deve ser observado também é a tamanha importância do desenvolvimento de softwares para dispositivos móveis, área que está em constante crescimento.

Ao desenvolver softwares para dispositivos móveis devem ser observados alguns itens, como, o tamanho, complexidade do sistema que pretende desenvolver, pois trata-se de equipamentos sem fio, providos de bateria e que devem em primeiro lugar ter uma durabilidade aceitável do usuário final, além de que dependendo do equipamento a capacidade de processamento pode variar bastante.

Alguns dos benefícios que se tem ao desenvolver um software para dispositivos móveis são o acesso e praticidade. Determinadas informações podem ser úteis para o usuário final em determinadas ocasiões, e estar provido de dispositivo móvel e com recurso suficiente para atender as necessidades do usuário é o ponto alvo no que diz respeito ao acesso e praticidade.

3. MATERIAIS E MÉTODOS

Este capítulo apresenta as ferramentas utilizadas para a elaboração do aplicativo móvel e o sistema web. Também serão apresentadas as metodologias utilizadas para atingir o resultado final.

3.1 Materiais

Os materiais utilizados no desenvolvimento do sistema proposto foram:

- **Aplicativo Web**

- a) **Bootstrap**

Bootstrap é um *framework* especializado para desenvolvimento da interface gráfica de aplicações web. Agrupa uma coleção de elementos visuais (botões, grades, títulos, textos, imagens...) de uma página e seus efeitos de transição e estado, como o posicionar do mouse sobre um botão, o carregamento de uma página com efeito de barra de progresso e muitos outros. *Bootstrap* é *open-soucee* e está disponível no próprio site do Bootstrap (BOOTSTRAP_GET, 2014)

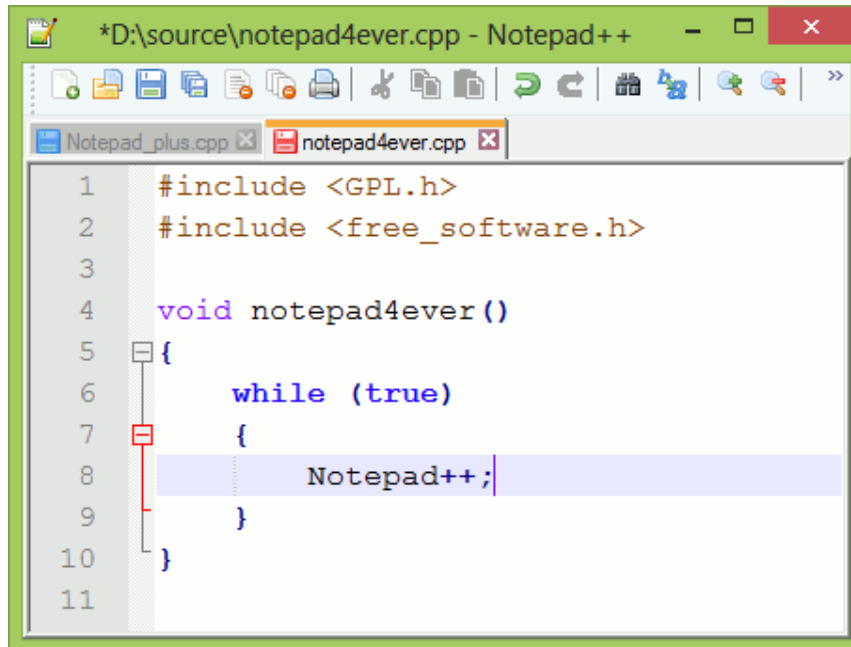
- b) **Notepad++**

Notepad++ é um editor de texto livre e editor de códigos fonte, suportando varias linguagens de programação.

Possui funcionalidades especiais como auto-completar, sistema de busca e substituição, gravação e execução de macros, navegação por abas e muito mais.

Notepad++ é gratuito e está disponível no próprio site do Notepad++ (NOTEPAD_DOWNLOAD, 2014)

A Figura 1 apresenta visão geral do sistema:

A screenshot of the Notepad++ application window. The title bar reads '*D:\source\notepad4ever.cpp - Notepad++'. The window contains two tabs: 'Notepad_plus.cpp' and 'notepad4ever.cpp'. The code in the active tab is as follows:

```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever ()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
11
```

Figura 1 - Visão geral do sistema Notepad++.
Fonte: NOTEPAD-PLUS-PLUS, 2014.

c) Google Chrome

Google Chrome é um navegador para internet, criado pela Google, combina tecnologias sofisticadas com um design simples para tornar a Web mais rápida, mais segura e mais prática. (GOOGLE, 2014)

Um dos recursos que essa ferramenta possui e foi utilizado no decorrer do desenvolvimento do sistema proposto, é o recurso de inspeção de código. Com ele foi possível escolher características visuais para o sistema, depurar códigos e acompanhar algumas funcionalidades. Este recurso pode ser acessado pela tecla de atalho “F12” ou botão direito do mouse sobre o conteúdo web, em menu suspenso, opção “Inspecionar Elemento”.

A Figura 2 apresenta visão geral do sistema com recurso de inspeção de código ativo:

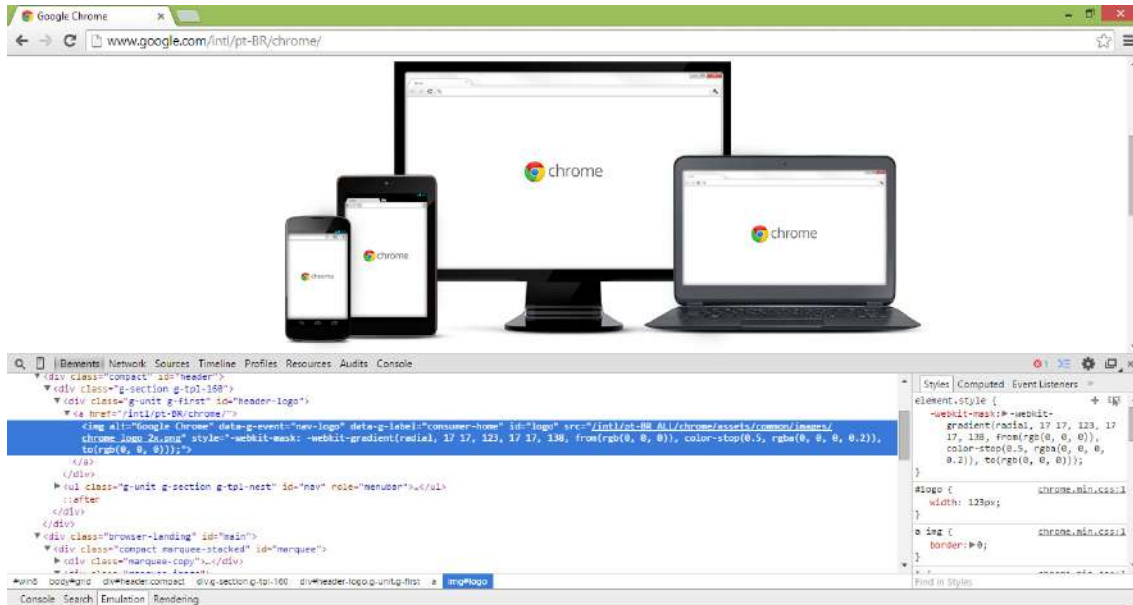


Figura 2 - Visão geral do sistema Google Chrome.
Fonte: Print screen da aplicação no sistema operacional Windows 8.

d) WAMP

WAMP (Windows, Apache, MySQL, PHP) é um conjunto de programas que cria um ambiente de desenvolvimento para web, e permite criar aplicações para web sob um sistema operacional Windows com um servidor Apache, servidor web livre compatível com protocolo HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto), através desse protocolo é possível se ter uma comunicação de redes por sistemas de informação de hipermídia, linguagem de programação PHP e banco de dados MySQL. Por outro lado possui uma ferramenta PhpMyAdmin que permite gerenciar facilmente os bancos de dados de forma gráfica. WAMP é gratuito e está disponível no próprio site da WampServer (WAMP_DOWNLOAD, 2014)

- **Aplicativo Móvel**

a) IDE de desenvolvimento Eclipse com Android SDK

Eclipse é uma interface para desenvolvimento de aplicações *open-source*, suporta varias linguagens C/C++, PHP, ColdFusion, Python, Scala, Plataforma android e principalmente Java. Android SDK é um kit que fornece bibliotecas e ferramentas necessárias para desenvolver, testar e depurar aplicativos Android. Uma das ferramentas utilizadas desse kit foi o AVD (*Android Virtual Device* - Dispositivo virtual com sistema operacional Android) que fornece uma interface

gráfica para testes das aplicações Android. Esse conjunto está disponível no próprio site do Android (ANDROID_SDK, 2014)

A Figura 3 apresenta visão geral do sistema:

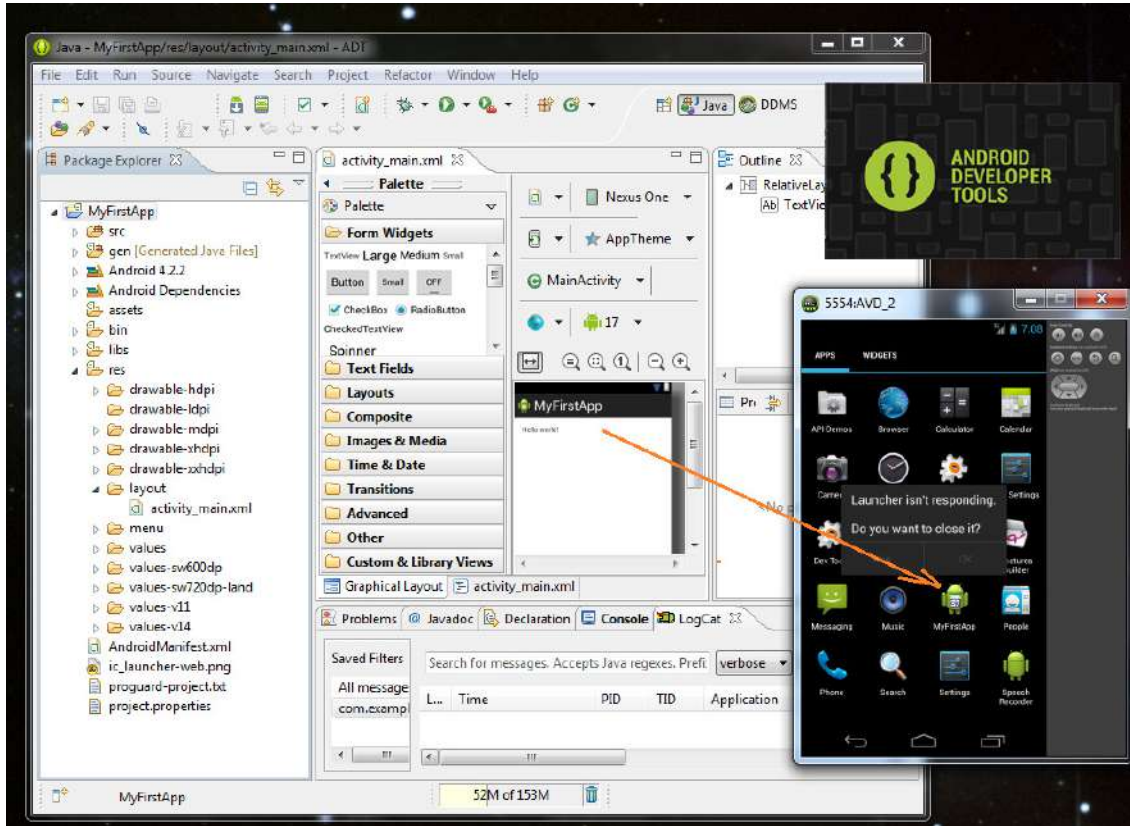


Figura 3 - Visão geral do sistema Eclipse com Android SDK
Fonte: ALEX_SENTCHA'S BLOG, 2014.

- **Geral**

- a) **Astah Community**

Astah Community é uma ferramenta de modelagem, utiliza especificamente à linguagem UML (*Unified Modeling Language* – Linguagem de modelagem unificada), metodologia de desenvolvimento, esta que auxilia na visualização dos desenhos e a comunicação entre os objetos. Com ela é possível manipular vários diagramas, são eles: Diagrama de Classes, Diagrama de Objetos, Diagrama de Componentes, Diagrama de Instalação, Diagrama de Pacotes, Diagrama de Estrutura, Diagrama de Caso de Uso, Diagrama de Estados e Diagrama de Atividade.

Astah possui versões pagas, a versão Community é gratuita e *open-source*, disponível no próprio site da Astah (ASTAH_DOWNLOAD, 2014)

A Figura 4 apresenta visão geral do sistema:

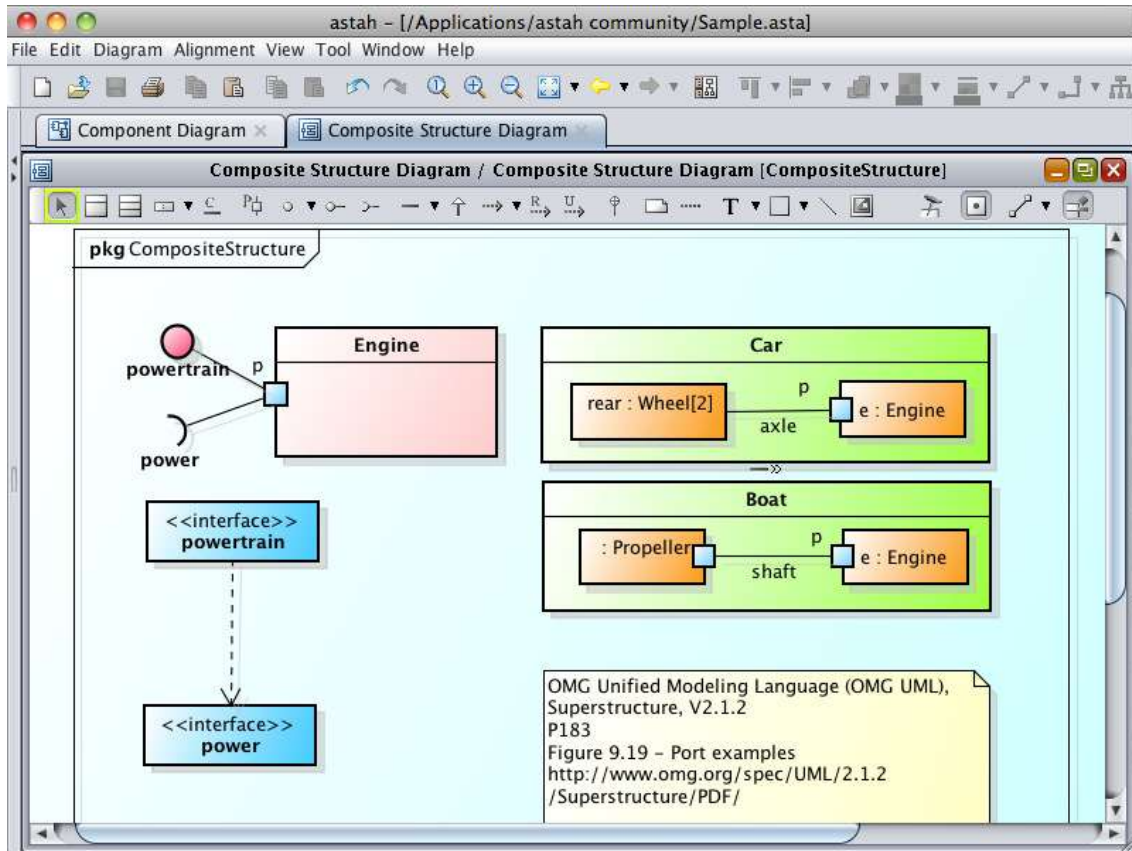


Figura 4 - Visão geral do sistema Astah Community
Fonte: ASTAH, 2014.

b) Estrutura do documento de especificação de software

A estrutura do documento definida e mantida como base para o desenvolvimento deste trabalho esta definida no seguinte contexto: Capa, Histórico de Revisões, Índice, Nome / Breve descrição, Pré-Condições, Pós-Condições, Atores, Fluxo de eventos, Requisitos, Requisitos Cancelados, Diagramas, Protótipos de interface, e Referências.

c) Aplicativo similar para gerenciamento de requisitos de software.

Como base para o desenvolvimento deste trabalho foi analisado um sistema similar para gerenciamento de requisitos de software, especificamente o “Rational RequisitePro”, ferramenta desenvolvida pela IBM que permite o gerenciamento de requisitos, baseado em documentos, essa possui integração ao Microsoft®Word. Para evitar retrabalho esses documentos ficam ligados a registros em um próprio

banco de dados da ferramenta. A Figura 5 apresenta a interface principal do sistema Rational RequisitePro:

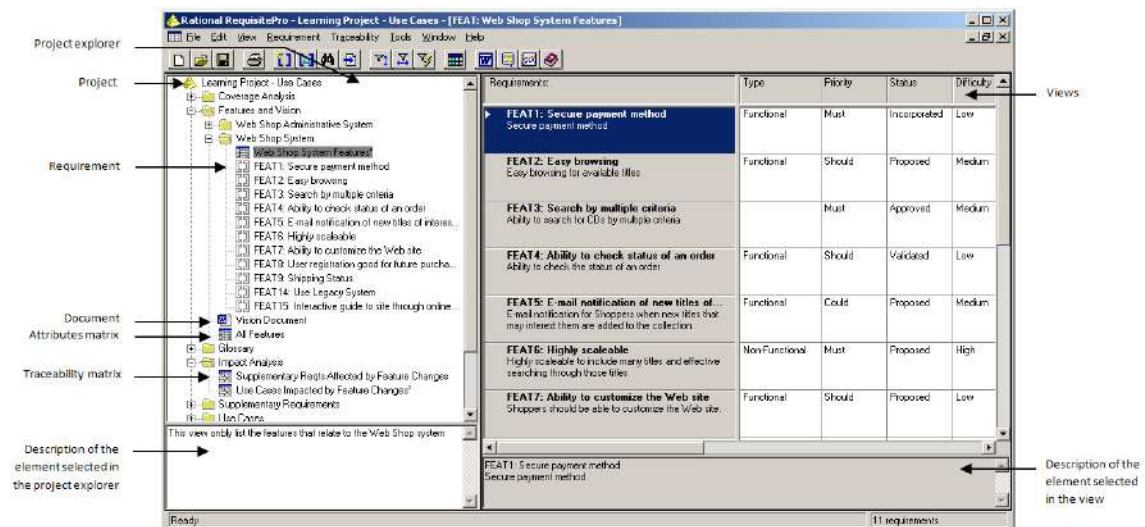


Figura 5 - Interface principal do Rational RequisitePro
Fonte: XCARECROWS.COM, 2014.

Essa ferramenta disponibiliza acesso aos documentos através de uma árvore de pastas criada pelos usuários do mesmo, essas pastas podem identificar sistemas diferentes ou módulos diferentes em um mesmo sistema. Cada documento apresentado na ferramenta é um link que possibilita a leitura e edição do documento através da ferramenta Microsoft® Word, em casos de documentos que possuem mais de uma versão, para o usuário poder ver e analisar alguma versão de caráter mais antigo é possível exportar arquivo escolhendo a versão desejada. Ferramentas de comparação de textos como o Kdiff3 (Comparador de diferenças em arquivos de texto e mesclagem), não possuem uma boa eficácia ao tentar comparar arquivos do Word, pois os arquivos do Word além de manter os dados textuais, guardam também informações do estilo da fonte e todo o visual do documento, estes que prejudicam o bom funcionamento das ferramentas de comparações de texto.

Como um diferencial, a proposta deste trabalho engloba a facilidade de acesso as versões através de opção disponível na grade de histórico de revisões. Identificando o usuário leitor se estiver lendo algum documento em edição por outro usuário e a versão que esta visualizando, fica dessa forma preparado o ambiente também para um trabalho futuro de integração com sistemas comparadores de texto, além de acesso rápido a estrutura do documento por meio de guia fixo disponível a lateral esquerda do documento, não necessitando o usuário percorrer o documento utilizando apenas a barra de rolagem.

3.2 Métodos

3.2.1 Metodologia de desenvolvimento XP

Para o desenvolvimento do sistema de versionamento de documentos de requisitos de software foi utilizado à metodologia XP (*Extreme Programming – Programação Extrema*)

Programação extrema é uma metodologia ágil, para desenvolvimento de software, voltada para pequenas e médias equipes onde os requisitos são vagos e mudam freqüentemente, tem como principais tarefas a codificação e testes com ênfase menor nos processos formais de desenvolvimento (WILDT, LACERDA, 2014)

A estratégia é baseada no constante acompanhamento, executando vários testes, e a implementação de pequenos ajustes no decorrer do desenvolvimento.

A metodologia XP pode ser dividida em quatro atividades:

a) Planejamento

Para esta atividade foi levantado os requisitos, tendo como base análise de aplicativo existente e a estrutura do documento de especificação de software definida, planejando assim como pode ser elaborada uma solução que facilite a utilização destes documentos tanto pelas pessoas que os criam (Analistas) como para as que os lêem (Programadores, Testadores).

b) Projeto

Para a atividade de projeto, foi elaborada uma arquitetura, através da confecção de alguns diagramas dispostos no decorrer deste trabalho. Para este foi utilizado à ferramenta Astah Community, está utiliza a linguagem UML para modelagem e auxilia na visualização dos produtos dos trabalhos e a comunicação entre os objetos em diagramas padronizados.

c) Codificação

Para esta atividade foram utilizadas ferramentas Notepad++, ferramenta de edição de texto livre especialmente configurada para as linguagens PHP, HTML, JS e SQL, utilizada para a confecção e manutenção dos arquivos PHP, HTML e JS do sistema web, e Eclipse, ferramenta interface de desenvolvimento disponível

individualmente no próprio site da Eclipse. (ECLIPSE_DOWNLOAD, 2014), utilizada para confecção e manutenção da aplicação móvel.

d) Testes

Para esta atividade vários testes unitários foram executados a fim da validação do sistema, fornecendo no decorrer do desenvolvimento do sistema sinais de alertas que garantem assim uma melhoria contínua e a qualidade do sistema. Foram utilizadas para os testes as ferramentas Google Chrome, navegador para internet disponível no próprio site da Google (CHROME_DOWNLOAD, 2014), utilizando o recurso de inspeção de código (Atalho “F12”), e o AVD - *Android Virtual Device*, Dispositivo virtual com sistema operacional Android disponível e distribuído juntamente com Eclipse ADT no próprio site do Android (ANDROID_SDK, 2014)

As praticas ágeis mais utilizadas no decorrer deste trabalho foram:

- **Releases pequenos:** Contendo poucas alterações, vários releases foram sendo criadas a fim de possibilitar testes exploratórios, facilitando a identificação de novas alterações e novos requisitos.
- **Testes constantes:** Independente a quantidade de alterações realizadas, a cada publicação, testes eram realizados, propondo a assegurar que o sistema estará em funcionamento livre de inconsistências.
- **Cliente envolvido:** Além da análise da estrutura básica do documento utilizado, o envolvimento com o cliente teve um papel importante a fim de delimitar alterações no sistema e para esclarecimento de dúvidas.

A Figura 6 apresenta a modelagem de XP:

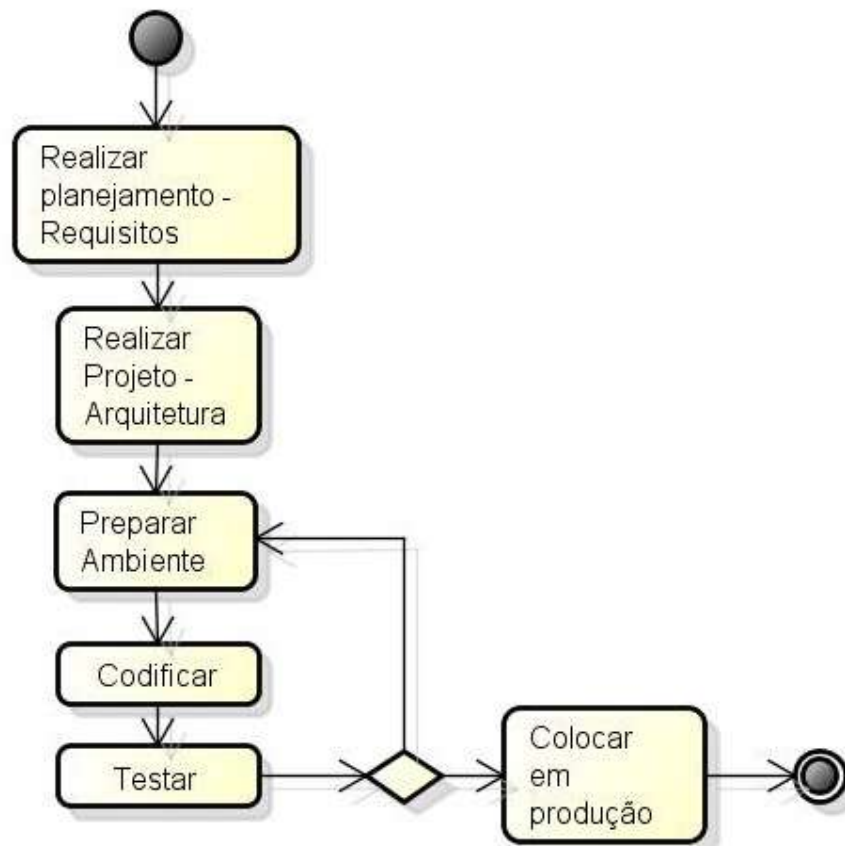


Figura 6 - Modelagem de XP usando diagrama de atividades.
 Fonte: Autoria Própria

3.2.2 Android

Empresas e desenvolvedores buscam formas de desenvolverem aplicações corporativas que auxiliam nos negócios e melhoram os lucros. Usuários comuns cada vez mais buscam dispositivos elegantes, modernos, de fácil utilização e repletos de recursos.

Para atender esse nicho de mercado várias empresas estão apostando no desenvolvimento de novas plataformas e tecnologias, a Google é uma delas, e lança ao mercado o Android, uma nova plataforma de desenvolvimento para aplicativos móveis, baseada em um sistema operacional Linux (LECHETA, 2013)

Android é um sistema operacional *open-source* (Programa de computador de código aberto) para dispositivos móveis baseado em Linux (Sistema operacional), e esta em alto crescimento desde que surgiu. A partir de Abril de 2009 várias versões foram publicadas, e a elas foram dados codinomes em ordem alfabética correspondente a nome de sobremesas e bolos: *Cupcake*, *Donut*, *Eclair*, *Froyo*,

Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean e atualmente *KitKat*. Segundo Eduardo Campos Dias (2014) Android cresce 40,3 % nas vendas entre 2012 a 2013.

Sistemas Operacionais	3º Trimestre de 2013 Volumes de vendas	3º Trimestre de 2012 Volumes de venda	Crescimento (%)
Android	226,1	161,1	40,3
iOS	51,0	47,8	6,7
Windows Phone	8,8	6,0	46,7
BlackBerry	1,7	7,4	-77,0
Outros	2,0	6,7	-70,1
Total	289,6	229,0	26,5

Tabela 1 - Sistemas operacionais, Crescimento vendas em milhões

Fonte: DIAS, 2014.

Esse sistema operacional proporciona o desenvolvimento de aplicativos com a linguagem Java.

Java inicialmente era um projeto financiado para atingir o mercado de dispositivos eletrônicos inteligentes destinados ao consumidor final, mas o mesmo não teve tanto crescimento como esperado, felizmente em 1993 a web explodiu em popularidade e a Sun viu o potencial de utilizar o Java para adicionar conteúdo dinâmico, como iteratividade e animações as paginas da web. Hoje Java é utilizado para desenvolver aplicativos de grande porte, aprimorar a funcionalidade de servidores web, fornecer aplicativos para dispositivos voltados para o consumo popular e para muitos outros propósitos (DEITEL e DEITEL, 2010)

A arquitetura de interface gráfica do android é baseada no padrão MVC (Model – View – Controller), Model, camada responsável pela regra de negócio da aplicação, View, camada responsável pela interface gráfica e Controller, camada responsável por controlar os eventos de cada tela (ANDROID ON BOARD, 2014)

Para maiores conhecimentos ficam os conceitos iniciais para desenvolvimento para plataforma android:

- **AndroidManifest.xml** – Arquivo XML (Extensible Markup Language) solto na raiz do projeto sendo o principal arquivo de configuração do aplicativo, nele

ficam definidos ícone e nome da aplicação, Activitys, definições de segurança, controle de versão, e outras informações.

- Activity – Assumindo papel da camada *Controller*, essa é a parte do comportamento do sistema, ações de botão e outros.
- Layout XML – Assumindo papel da camada *view*, este é responsável pela apresentação gráfica, botões, campos de entrada de dados, navegador, entre outros.
- Classe R – Essa classe é responsável pela ligação entre as camadas *View* e *Controller*, respectivamente layouts XML e Activitys
- Strings.xml – Responsável pela guarda de literais, facilita a manutenção textual da aplicação e internacionalização quando necessário.

Estes são apenas algumas considerações iniciais, o sistema operacional android está em constante evolução, e disponibiliza uma SDK (*Software Development Kit* - Kit de Desenvolvimento de Software) com grande leque de recursos a ser explorado pela criatividade e necessidade, para este projeto será criado uma aplicação com recurso de navegação na internet, utilizando componente visual “Webview” capaz de navegar a url correspondente ao sistema proposto.

3.2.3 PHP

PHP (*Hypertext Preprocessor*) é uma linguagem de programação para web de script *open-source*, trabalha mesclado ao HTML (*Hypertext Markup Language*) e é executado no lado servidor, o que possibilita que o site seja dinâmico, e garante mais segurança nas informações, já que o código PHP é todo processado no lado servidor e o lado cliente fica responsável apenas de apresentar os dados a tela do usuário.

O PHP era uma solução para utilização em suas páginas pessoais, contudo necessidades foram surgindo e novas versões foram lançadas, contendo novos recursos, como, interpretador de comandos SQL (Linguagem de consulta estruturada), orientação a objetos, herança, polimorfismo, apontadores de memória e outros (PHP.NET, 2014)

Hoje essa linguagem é utilizada em massa, o que demonstra o Gráfico 2:

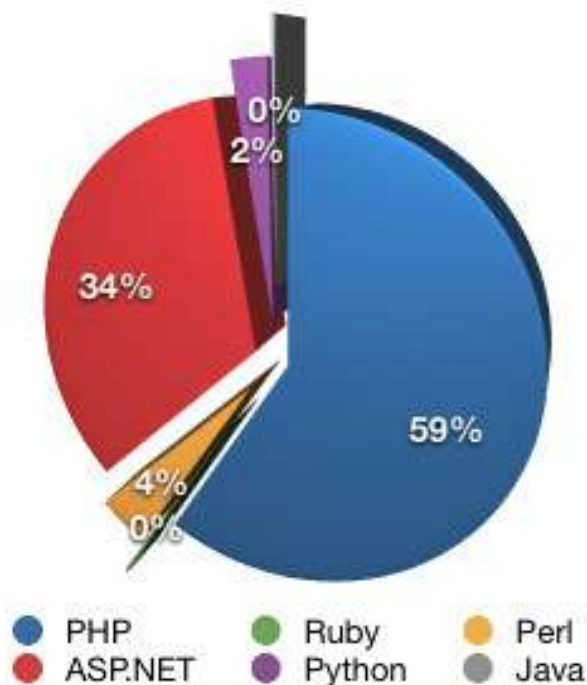


Gráfico 1 - Estatística de uso de linguagens em servidores web
Fonte: ALSHANETSKY, 2014.

Sendo esta utilizada por aplicações referências como MediaWiki, Facebook, Drupal, Joomla, WordPress, Magento e o Oscommerce. Essa linguagem também disponibiliza suporte a um grande número de bases de dados, inclusive MySQL o qual será utilizado no contexto desse trabalho.

Segundo Bento (2014) alguns motivos de usar o PHP e o MySQL são: PHP nasceu para a web e sua integração com servidores web é simples, tem uma curva de aprendizado suave, comparada a outras linguagens, PHP e MySQL são tecnologias livres, é fácil de encontrar serviços de hospedagem que oferecem PHP e MySQL. Serviços de hospedagem que as oferecem são mais baratos que serviços semelhantes para outras tecnologias e MySQL é leve e rápido, mesmo para quantidades razoavelmente grandes de dados.

3.2.4 Banco de dados MYSQL

Banco de dados é um local no qual é possível armazenar informações, para consulta ou utilização, quando necessário. Todos os bancos de dados são constituídos por três elementos básicos: campos, registros e tabelas. Cada banco de dados possui complexas estruturas internas de funcionamento, uma diferente da

outra, a fim de facilitar o acesso aos elementos do banco de dados foi criada e distribuída pela IBM em 1981 uma linguagem de consulta, o SQL, que se tornou uma linguagem de acesso aos bancos de dados muito articulada e funcional que pode ser empregada em computadores de arquiteturas totalmente diferentes e é a linguagem utilizada pelo MySQL (FERRARI, 2007)

MySQL é o banco de dados de código aberto mais popular do mundo, que possibilita a entrega econômica de aplicações de banco de dados confiáveis, de alto desempenho e escaláveis, com base na Web e incorporadas (ORACLE, 2014)

Nele ficam armazenadas todas as informações dos usuários, documentos e caminhos de imagens mantidos pelo sistema proposto.

As operações suportadas por esse banco são inclusão, exclusão, atualização de registros, alterações de estrutura e outras.

O banco de dados MySQL além de ter um excelente desempenho, possui compatibilidade e fácil integração com a linguagem de programação PHP, um dos motivos da escolha desse kit, PHP + MySQL, para o desenvolvimento deste trabalho.

3.2.5 BOOTSTRAP

Bootstrap é o mais popular *framework* (Abstração de códigos que prove uma funcionalidade genérica) *open-source* de HTML, CSS (Cascading Style Sheets - Folhas de estilo) e JS (JavaScript – Linguagem de programação para internet) responsivo disponível para internet, foi projetado para todos, tornando o desenvolvimento visual de uma página para internet mais rápido e fácil. *Bootstrap* é uma coleção de vários elementos (formulários, botões, navegação, e outros) e funções personalizáveis.

Bootstrap foi criado por Mark Otto e Jacob Thornton no Twitter, a fim de eliminar inconsistências e diminuir o número de manutenções internas, após a primeira apresentação interna, várias equipes iniciaram a utilização da nova ferramenta (BOOTSTRAP, 2014)

Em agosto de 2011 o Twitter lançou *Bootstrap* como código aberto, em fevereiro de 2012 foi o projeto de desenvolvimento no GitHub mais utilizado, posição mantida em junho de 2014 (GITHUB, 2014)

A Figura 7, mostra alguns componentes visuais do *Bootstrap*:

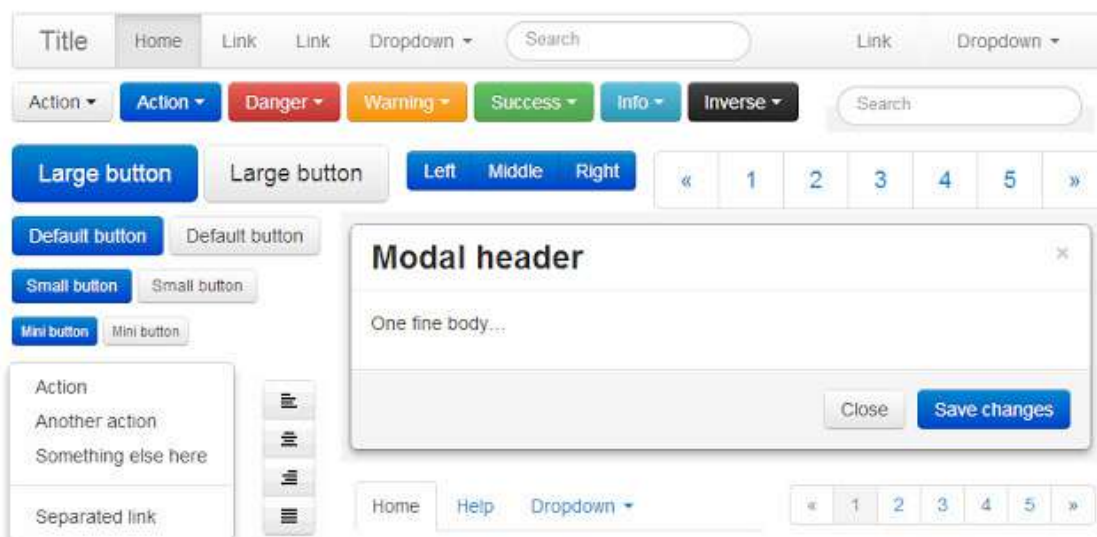


Figura 7 - Componentes de interface utilizando framework Bootstrap
Fonte: D3NOOB.ORG, 2014.

Esse *framework* está disponível para *download* (baixar, Carregar informações ou programas no computador) e utilização através do site <<http://getbootstrap.com/>>. Vários exemplos de utilização e seus componentes são apresentados através do site do próprio Bootstrap (BOOTSTRAP_COMPONENTES, 2014)

3.2.6 Modelagem do Software

Esta seção apresenta a análise realizada para o sistema abordando as diferenças em subseções separadas, são elas web e móvel, citando os requisitos funcionais e não funcionais, fatores determinantes do funcionamento do sistema. Para auxiliar no desenvolvimento do sistema foram criados alguns diagramas que facilitam o entendimento de modo geral.

a) Análise da aplicação Web

Como primeiro passo na análise e modelagem do sistema, os requisitos funcionais e não funcionais foram elaborados a fim de delimitar características e operações que o sistema deve e não deve possuir. Os requisitos serão apresentados nos quadros seguintes:

F1 Manter Perfil de Usuário		Oculto ()		
Descrição: O sistema deve possibilitar a edição do perfil do usuário. Alterar Nome, Email, Usuário e Senha.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Identificação do Usuário	O código único referente ao usuário deve ser gerado automaticamente pelo sistema, e não pode ser editado.	Automação	()	(X)
NF1.2 Foto do usuário	O sistema deve manter relacionada uma imagem do usuário.	Interface	(X)	()
NF1.3 Projeto Default	O sistema deve manter relacionado um projeto default para o usuário.	Interface	(X)	()
NF1.4 Nível de Usuário	O usuário Administrador poderá no sistema manter relacionado um nível de usuário que define o papel entre Administrador, Analista, Leitor ou Bloqueado	Segurança	()	(X)

Quadro 1 - Requisito Aplicação web: Manter Perfil de Usuário
Fonte: Autoria Própria

F2 Manter Mensagens Recebidas		Oculto ()		
Descrição: O sistema deve manter as mensagens recebidas de outros usuários.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Apresentação de mensagens	O sistema deve apresentar as mensagens recebidas de outros usuários para o usuário logado.	Interface	()	(X)
NF2.2 Manter por tempo indeterminado	O sistema não deve permitir a exclusão das mensagens, apenas identificar se esta pendente de leitura, se já foi lido ou se foi respondido.	Histórico	()	(X)
NF2.3 Responder	O sistema deve possibilitar a resposta na mesma tela de leitura.	Interface	()	(X)

Quadro 2 - Requisito Aplicação web: Manter Mensagens Recebidas
Fonte: Autoria Própria

F3 Manter Mensagens Enviadas		Oculto ()		
Descrição: O sistema deve manter as mensagens enviadas para outros usuários.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Apresentação de mensagens	O sistema deve apresentar as mensagens enviadas para outros usuários do usuário logado.	Interface	()	(X)
NF3.2 Manter por tempo indeterminado	O sistema não deve permitir a exclusão das mensagens, apenas identificar se esta pendente de leitura, se já foi lido ou se foi respondido.	Histórico	()	(X)
NF3.3 Responder	O sistema deve possibilitar o envio de uma nova resposta da própria mensagem.	Interface	()	(X)

Quadro 3 - Requisito Aplicação web: Manter Mensagens Enviadas
Fonte: Autoria Própria

F4Apresentar Itens Reservados		Oculto ()		
Descrição: O sistema deve apresentar os itens reservados, a fim de facilitar o acesso ao mesmo.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Apresentação dos itens	O sistema deve apresentar os itens reservados do usuário logado.	Interface	()	(X)
NF4.2 Acesso ao documento	O sistema deve possibilitar o acesso ao documento ao simples clique do item reservado.	Usabilidade	()	(X)

Quadro 4 - Requisito Aplicação web: Apresentar Itens Reservados
Fonte: Autoria Própria

F5Manter Cadastro de Usuários		Oculto ()		
Descrição: O sistema deve manter o cadastro de usuários.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF5.1 Acesso	O sistema só deve permitir acesso a manutenção de cadastro de usuários pelo usuário administrador.	Segurança	()	(X)
NF5.2 Apresentação de usuários	O sistema deve apresentar os usuários e suas características particulares.	Interface	()	(X)
NF5.3 Ocultação de senha	O sistema deve manter oculta a senha dos usuários, apresentando apenas o caractere * na quantidade equivalente a quantidade de caracteres da senha descriptografada.	Segurança	()	(X)
NF5.4 Acesso ao cadastro de usuário	O sistema deve possibilitar o acesso ao cadastro do usuário ao simples clique nos usuários listados.	Usabilidade	()	(X)

Quadro 5 - Requisito Aplicação web: Manter Cadastro de Usuários
Fonte: Autoria Própria

F6Manter Cadastro de Atores		Oculto ()		
Descrição: O sistema deve manter o cadastro de atores.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF6.1 Apresentação dos atores	O sistema deve apresentar os atores.	Interface	()	(X)
NF6.2 Acesso	O sistema só deve permitir acesso à manutenção de cadastro de atores para usuários Administrador ou Analista.	Segurança	()	(X)
NF6.3 Tipo de Ator	O sistema deve manter relacionado um tipo para o ator, entre Pessoa, Hardware e Software.	Interface	()	(X)

Quadro 6 - Requisito Aplicação web: Manter Cadastro de Atores
Fonte: Autoria Própria

F7Manter Cadastro de Projetos		Oculto ()		
Descrição: O sistema deve manter o cadastro de projetos.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF7.1 Apresentação dos projetos	O sistema deve apresentar os projetos.	Interface	()	(X)
NF7.2 Acesso	O sistema só deve permitir acesso à manutenção de cadastro de projetos para os usuários Administrador e Analista.	Segurança	()	(X)

Quadro 7 - Requisito Aplicação web: Manter Cadastro de Projetos
Fonte: Autoria Própria

F8Manter Cadastro de Pastas		Oculto ()		
Descrição: O sistema deve manter o cadastro de pastas.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF8.1 Apresentação	O sistema deve apresentar as pastas de forma hierárquica, resultando numa árvore de pastas.	Interface	()	(X)
NF8.2 Manutenção	O sistema deve permitir a manutenção do cadastro de pastas para os usuários Administrador e Analista.	Segurança	()	(X)
NF8.3 Listagem de Documentos	O sistema deve listar os documentos relacionados as pastas quando acessado a listagem de pastas através do menu principal.	Interface	()	(X)
NF8.4 Inativos	O sistema deve exibir as pastas com status inativo, e as pastas posteriores a ela não serão exibidas.	Interface	(X)	()

Quadro 8 - Requisito Aplicação web: Manter Cadastro de Pastas
Fonte: Autoria Própria

F9Manter Cadastro de Documentos		Oculto ()		
Descrição: O sistema deve manter o cadastro de Documentos.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF9.1 Identificação do Usuário	O sistema deve identificar se o usuário logado esta com o documento reservado e disponibiliza interface de edição.	Interface	()	(X)
NF9.2 Opção de Reserva	O sistema ao identificar que o documento não esta reservado deve disponibilizar a opção de reservar documento caso o usuário seja Administrador ou Analista.	Interface	()	(X)
NF9.3 Leitura de Documento em edição	O sistema deve alertar o usuário leitor que o documento esta sendo editado.	Interface	(X)	()
NF9.4	O sistema deve identificar a	Interface	(X)	()

Identificação de versão que esta sendo visualizada	versão que o usuário está lendo.			
NF9.5 Layout responsivo	O sistema deve identificar a largura da tela do dispositivo e redesenhar a tela.	Usabilidade	()	(X)
NF9.6 Guia	O sistema deve dispor de um guia para facilitar o acesso aos tópicos do documento	Usabilidade	(X)	()
NF9.7 Histórico	O sistema deve manter registros de alterações a fim de diferenciar versões.	Histórico	()	(X)
NF9.8 Identificação de Alteração	O sistema deve disponibilizar a cada tópico uma opção de gravação quando uma alteração for feita.	Interface	()	(X)
NF9.9 Validações	O sistema deve validar campos importantes, identificando com visual avermelhado ou apresentando balões de aviso, impedindo que a gravação continue.	Segurança	()	(X)
NF9.10 Publicação de versão	O sistema deve no tópico de Histórico de revisões, disponibilizar opção para Concluir alterações e lançar versão do documento.	Interface	()	(X)

Quadro 9 - Requisito Aplicação web: Manter Cadastro de Documentos

Fonte: Autoria Própria

F10 Aplicar ferramenta Bootstrap	Oculto ()			
Descrição: O sistema deve possuir uma interface agradável e adaptável.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF10.1 Componentes Visuais	O sistema deve possuir uma interface agradável, e de fácil manutenção.	Interface	()	(X)
NF10.2 Interface Adaptável	O sistema ao identificar que está em execução em telas de tamanhos menores, deve adaptar a interface para melhor visualização do conteúdo.	Interface	()	(X)

Quadro 10 - Requisito Aplicação web: Aplicar ferramenta Bootstrap

Fonte: Autoria Própria

F11 Aplicar linguagens para multiplataforma	Oculto ()			
Descrição: O sistema deve funcionar multiplataforma.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF11.1 Visualização	O sistema deve funcionar e manter uma interface semelhante em diferentes plataformas.	Interface	(X)	()

Quadro 11 - Requisito Aplicação web: Aplicar linguagens para multiplataforma

Fonte: Autoria Própria

Na seqüência foram elaborados alguns diagramas, a fim de facilitar o entendimento de usabilidade do sistema e facilitar o desenvolvimento. Na Figura 8 pode ser visualizado diagrama de caso de uso, os atores representados nessa figura identificam os usuários do sistema, que diferenciam seus papéis e níveis de acesso:

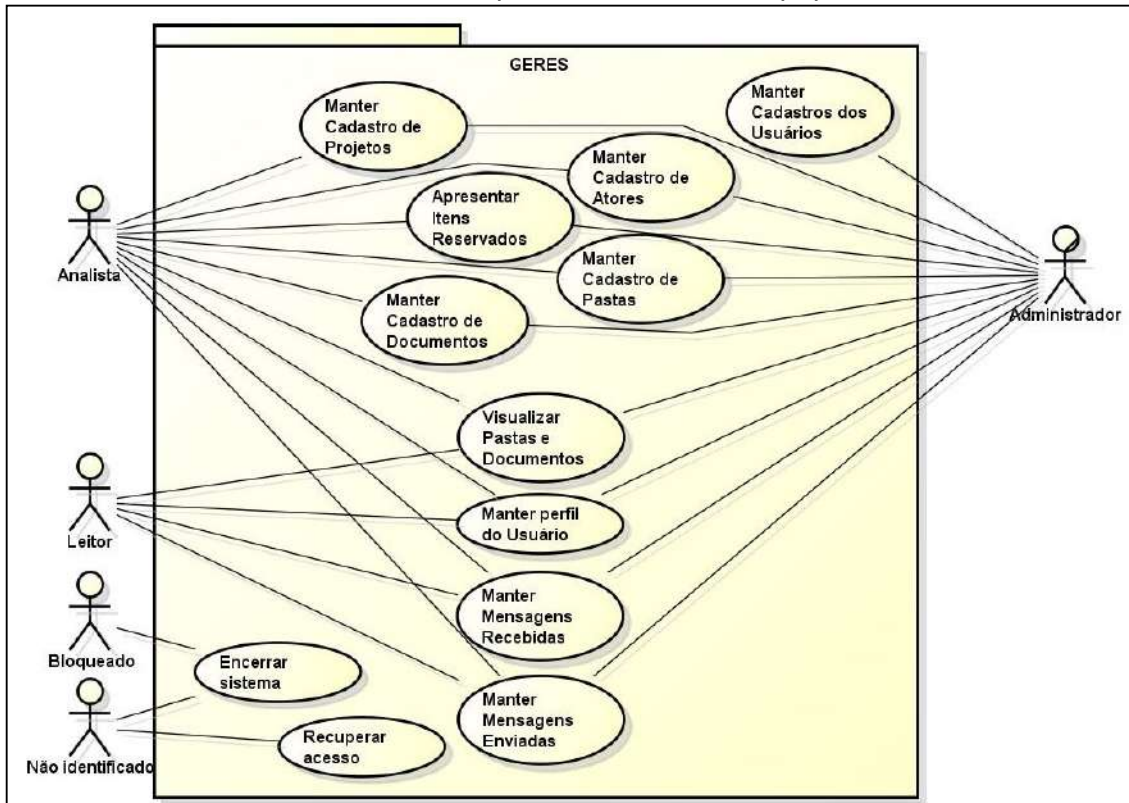


Figura 8 - Diagrama de caso de uso geral da aplicação web.
Fonte: Autoria Própria

Na Figura 9 é apresentado um diagrama de classe que identifica de uma visão geral da principal parte do sistema:

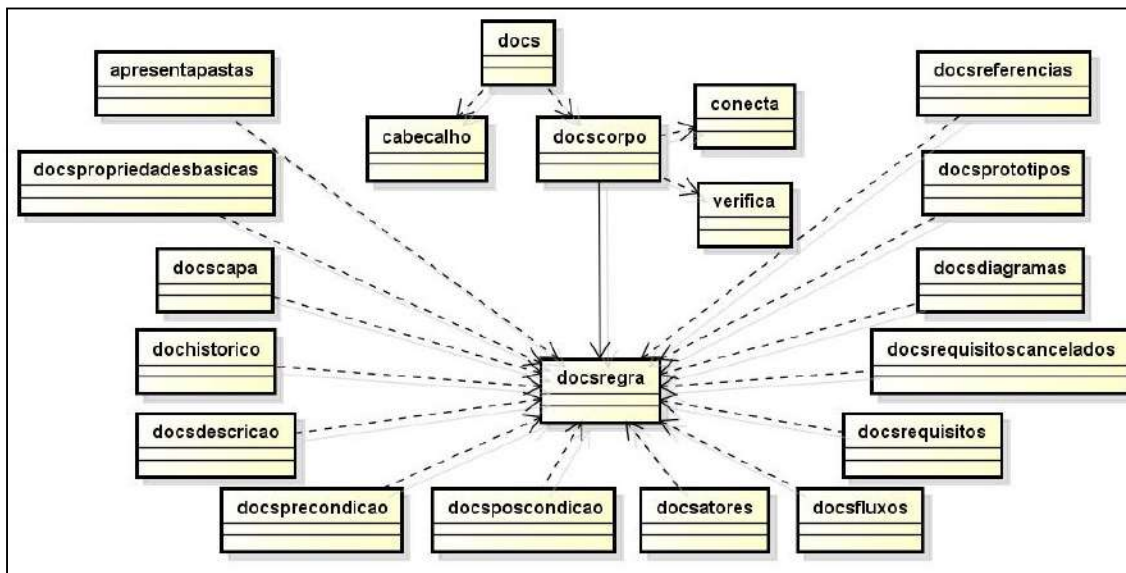


Figura 9 - Diagrama de classe principal da aplicação web.

Fonte: Autoria Própria

A Figura 10 representa o diagrama de entidade e relacionamento.

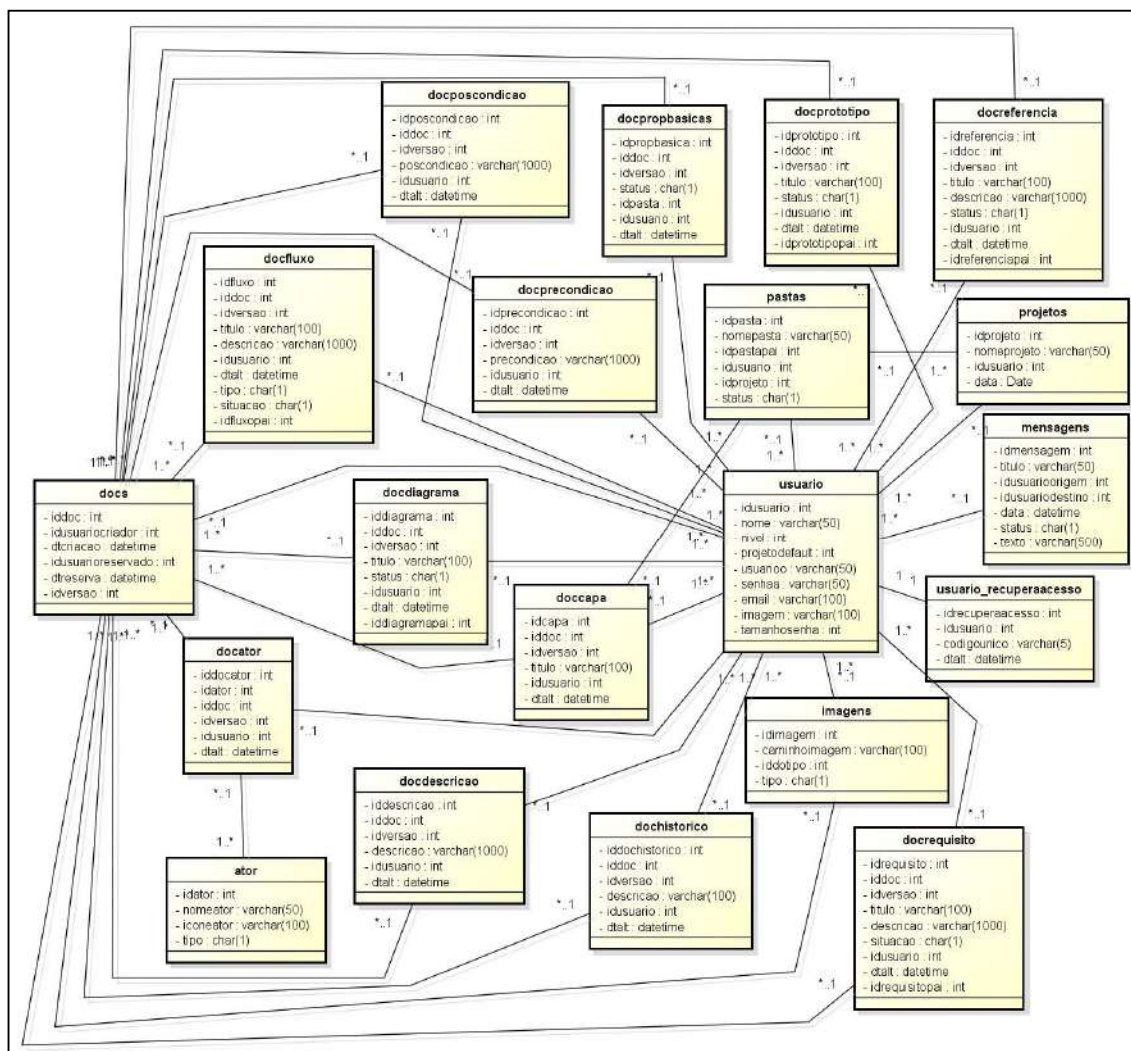


Figura 10 - Diagrama de entidade e relacionamento.
Fonte: Autoria Própria

O quadro 12 representa o dicionário de dados da tabela “docs”, responsável por registrar os documentos e controlar as reservas dos mesmos. Os demais quadros referentes ao dicionário de dados completo estão disponíveis no anexo A.

Coluna	Tipo	Nulo	Omissão	Comentários
iddoc	int	Não		PK Auto_Increment
idusuariocriador	int	Não		FK com usuario.idusuario
dtcriacao	datetime	Não		
idusuarioresevado	int	Sim	NULL	FK com usuario.idusuario
dtreserva	datetime	Sim	NULL	
idversao	int	Não	0	

Quadro 12 - Tabela docs: Responsável por registrar os documentos e controlar as reservas dos mesmos.
Fonte: Autoria Própria

b) Análise da aplicação Android

Novamente para a análise e modelagem do sistema, primeiramente foi elaborado a identificação dos requisitos funcionais e não funcionais, delimitando assim as operações que o aplicativo móvel deverá ter. Os quadros 13, 14 e 15 apresentam os requisitos levantados:

F12 Comunicação com a Internet		Oculto ()		
Descrição: O sistema deve estar conectado a internet.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF12.1 Permissão	O sistema deve solicitar permissão de usuário para estar conectado com a internet	Conexão	()	(X)

Quadro 13 - Requisito Aplicação móvel: Comunicação com a Internet
Fonte: Autoria Própria

F13 Interface navegador para internet		Oculto ()		
Descrição: O sistema deve possuir uma interface para navegar na internet.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF13.1 Carregamento	O sistema não deve ser redirecionado para outra aplicação que navega pelo site, o sistema deve ser carregado na interface do próprio aplicativo.	Interface	(X)	()

Quadro 14 - Requisito Aplicação móvel: Interface navegador para Internet
Fonte: Autoria Própria

F14 Definições de configuração		Oculto ()		
Descrição: O sistema deve manter configurações de endereço do sistema.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF14.1 Configuração	O sistema não deve perder a configuração default, deve ser disponível uma opção de marcação para voltar a configuração original.	Configuração	()	(X)

Quadro 15 - Requisito Aplicação móvel: Definições de configuração
Fonte: Autoria Própria

Posteriormente a elaboração dos requisitos, foram desenhados alguns diagramas. A Figura 11 apresenta o diagrama de caso de uso da usabilidade do sistema móvel.



Figura 11 - Diagrama de visão geral da interação do sistema móvel.
Fonte: Autoria Própria

A Figura 12 apresenta um diagrama de classe representando o funcionamento do aplicativo Móvel em comunicação com o Aplicativo Web.

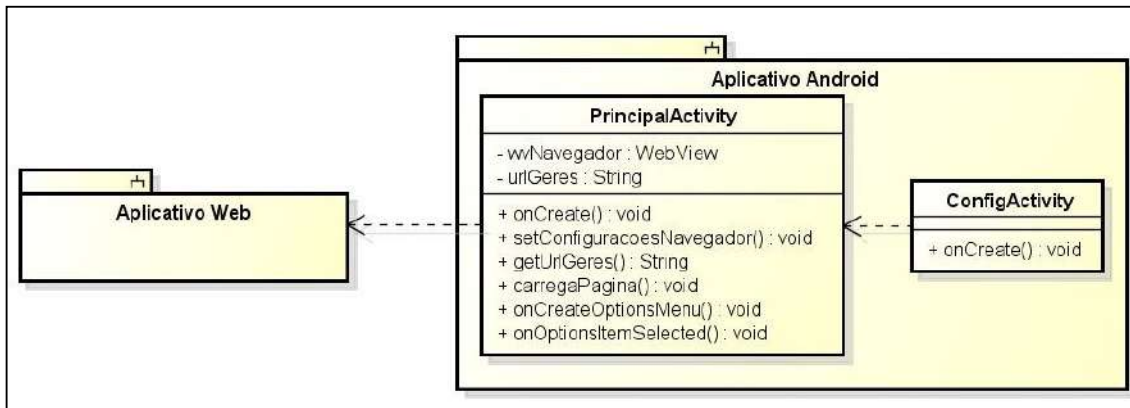


Figura 12 - Diagrama de classe principal da aplicação Android.
Fonte: Autoria Própria

A análise realizada levantou uma arquitetura para o desenvolvimento do aplicativo móvel que não depende de acesso ao banco de dados diretamente, e sim o acesso ao sistema web em um navegador incorporado ao aplicativo móvel, sendo assim, não foi elaborado nenhum dicionário de dados para este.

4. RESULTADOS

Este capítulo apresenta a análise realizada tanto para aplicativo móvel e web, alguns diagramas que exemplificam a utilização do sistema proposto, e outros que representam uma visão de classes e objetos.

São apresentados também o procedimento de desenvolvimento dos aplicativos, as telas, funcionalidades e alguns códigos, que representam algumas funcionalidades principais.

4.1 Desenvolvimento do aplicativo Web

Para o desenvolvimento da aplicação web, primeiramente foi instalado e configurado o ambiente de desenvolvimento, como primeiro passo, foi efetuado o download do aplicativo Google Chrome, instalado e configurado seguindo configurações típicas oferecidas pelo instalador. Na seqüência foi efetuado o download do aplicativo WAMP, instalado conforme orientações do próprio instalador, seguindo configurações típicas também, na seqüência foram testadas algumas funcionalidades como o servidor apache, testando o acesso ao endereço <http://localhost/> no navegador, e também foi testado o acesso a ferramenta gráfica de manutenção para o banco de dados, o PhpMyAdmin, acessando o endereço <http://localhost/phpmyadmin/>no navegador.

Aproveitando o teste de acesso às ferramentas do servidor e acessando a ferramenta PhpMyAdmin, foram criados manualmente as tabelas conforme discriminado no dicionário de dados, na análise da aplicação web citada anteriormente.

Foi também utilizada como forma de segurança, a geração de arquivos de backup do banco de dados. Que pode ser obtido seguindo os passos: Acessar a ferramenta PhpMyAdmin, selecionar o banco de dados desejado, selecionar a aba “Exportar”, manter as seleções padrões, “Export Method: Quick” e “Format: SQL” e selecionar o botão Executar, um arquivo será carregado no computador, conforme a Figura 13.

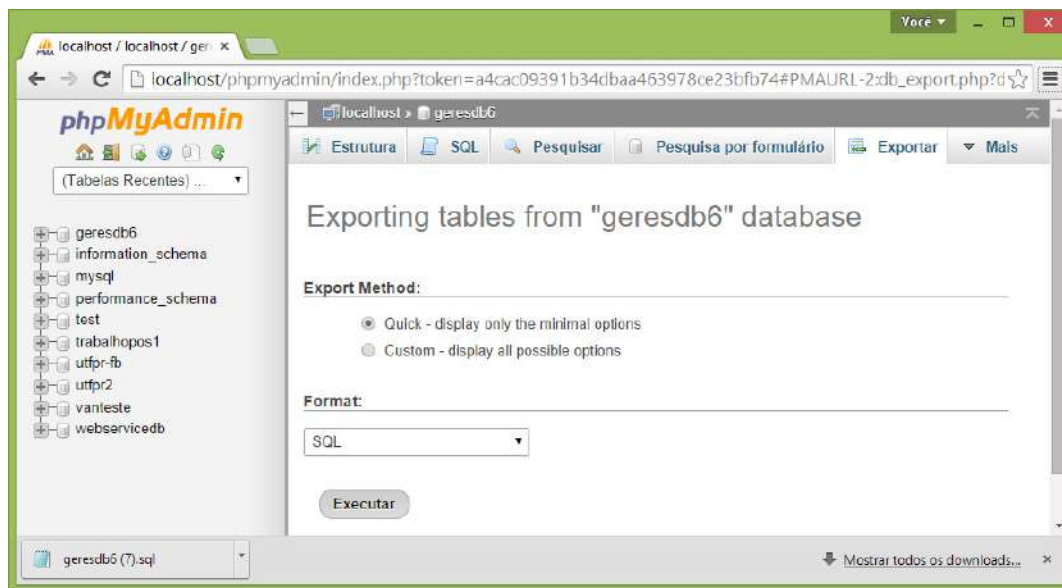


Figura 13 - Exportar Banco de dados para Arquivo SQL.
Fonte: Autoria Própria

Após, foi carregado e instalado o aplicativo Notepad++ no computador, para proporcionar uma interface de desenvolvimento, com a ferramenta Notepad++ foram criados e mantidos os arquivos que definem a interface gráfica, a regra do negócio e toda a interação e comunicação com o banco de dados.

Na seqüência foram criados todos os arquivos necessários para o funcionamento do sistema, alguns códigos e trechos tidos como principais são apresentados em listagens no Anexo B.

No decorrer do desenvolvimento do sistema vários testes foram efetuados a fim de garantir o funcionamento do mesmo. Para isso foi utilizado o recurso de inspecionar elemento do Google Chrome, que pode ser acessado através do clique direito do mouse sobre a área desejada a inspecionar e selecionar a opção "Inspecionar Elemento" ou pressionar a tecla de atalho "F12" e na área que será apresentada navegar até o elemento desejado. Através desse recurso é possível validar e acompanhar valores de algumas variáveis javascript, testar métodos e até fazer ajustes temporários, sejam eles de visual ou até mesmo do comportamento do sistema.

A Figura 14 apresenta o recurso de Inspecionar elemento sendo utilizado:

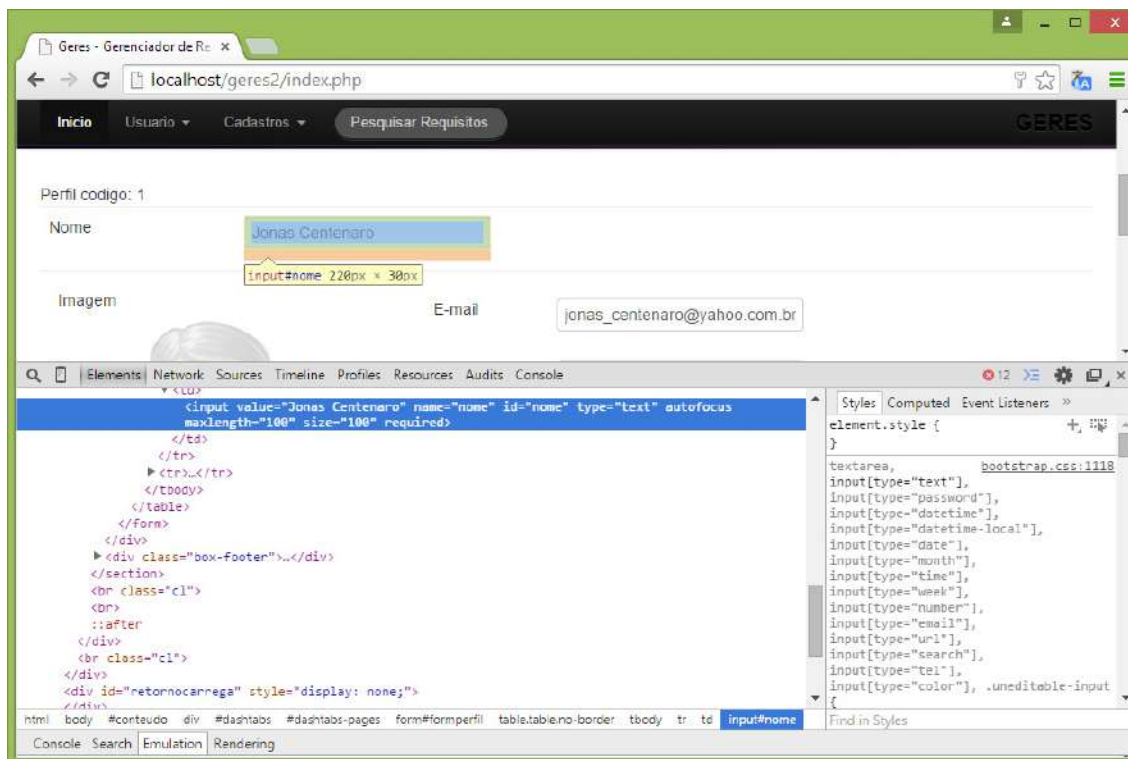


Figura 14 - Recurso de Inspeccionar Elemento do Google Chrome
Fonte: Autoria Própria

4.1.1 Apresentação do aplicativo Web

Neste tópico serão apresentadas algumas telas da aplicação web, produto do sistema objeto principal deste trabalho.

Como primeira tela, apresenta-se uma interface de autenticação, a tela de login, nela pode-se informar os valores de usuário e senha e pressionar o botão “Acessar” para possibilitar o acesso ao sistema, o acesso ao sistema é permitido e fica relacionado aos usuários cadastrados apenas 3 níveis, Administrador, Analista ou Leitor.

Na mesma tela de login pode-se observar uma área identificada como “Esqueci meu acesso?”, nela é possível o usuário solicitar a recuperação de acesso quando o mesmo não o consegue, essa área é expansível, ao expandir é disponibilizado um campo para o usuário informar um e-mail válido, esse e-mail deve estar relacionado ao usuário de acesso ao sistema GERES. Depois de informado dados corretos e solicitar a recuperação, o sistema retorna um informativo com a seguinte mensagem: “E-mail enviado com sucesso, Verifique em seu email os procedimentos para recuperar o acesso. Aguardamos..”

Sob a tentativa de acesso com informações inválidas, seja usuário ou senha incorretos, a tela de login é apresentada com um informativo: Usuário ou Senha incorretos.

A Figura 15 apresenta a tela de login com a área de recuperação de acesso expandida.

GeRes - Gerenciador de Requisitos de Software - Login

Usuário: Informe seu Usuário.

Senha:

Acessar

Esqueci meu acesso?

Informe seu email:

Receberá em seu email Procedimentos de recuperação do acesso.

Enviar

Figura 15 - Tela de login com "Esqueci meu acesso?" expandido.
Fonte: Autoria Própria

Na seqüência ao verificar a caixa de e-mails recebidos é possível identificar um email com assunto "Acesso Geres" com o seguinte conteúdo:

"Ola Nome do Usuário,
recebemos uma solicitação de recuperação de acesso ao sistema GERES,
se ainda precisar, disponibilizamos um link para recuperar o mesmo
e permanecerá disponível por 10 minutos.
Link: [Clique aqui para recuperar seu acesso.](#) Obrigado.."

Conforme Figura16.

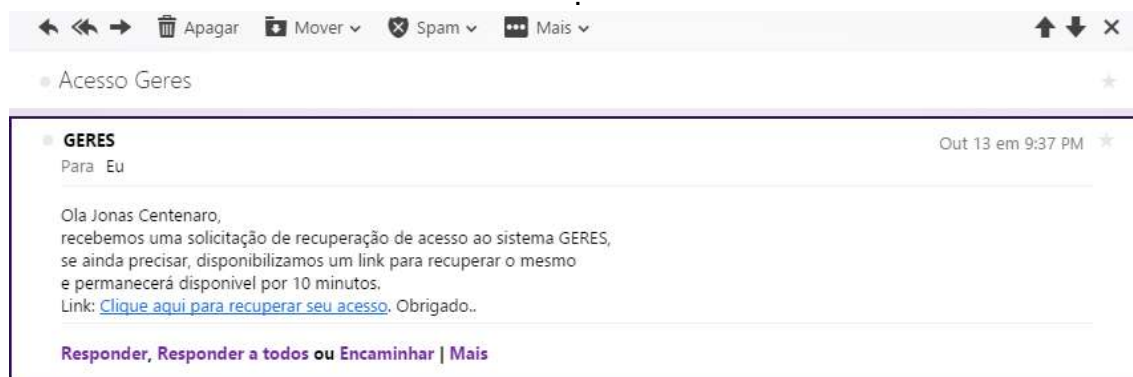


Figura 16 - E-mail de recuperação de acesso ao sistema Geres.

Fonte: Autoria Própria

Esse conteúdo possui um link que direciona para uma tela de recuperação de acesso que prove o direito a criar uma senha nova, esse link de recuperação de acesso possui algumas características ocultas de segurança, como um código único, gerado aleatoriamente, e que fica relacionado ao usuário, e-mail, e tempo em que o mesmo foi gerado, limitando ao tempo de 10 minutos para a tentativa de recuperação, caso o tempo esteja esgotado, um informativo é apresentado, com a seguinte mensagem: "Identificação de recuperação de acesso ou prazo expirado."

Depois de efetuado acesso ao sistema com sucesso, a tela apresentada na Figura 17 pode ser visualizada, disposta de alguns itens de acesso rápido que são eles Perfil, Mensagens, Logoof / Sair, e sempre disponível um campo de pesquisa rápida para acesso aos documentos.



Figura 17 - Página inicial do sistema.
Fonte: Autoria Própria

O diferencial entre os níveis de acesso serão listados:

Administrador: Não possuem restrições, todas as opções disponíveis de acesso;

Analista: Não possui acesso ao cadastro e listagem de usuários, e não possui direito de alteração de tipo de nível de usuário. Quando aos documentos pode criar e/ou reservar quando disponível, ou efetuar leitura de quaisquer versões desde que não esteja em edição pelo próprio usuário.

Leitor: Não possui acesso a opção de listagem de reservas, sob o item de cadastros possui acesso apenas a listagem de pastas. E quanto aos documentos apenas pode efetuar a leitura de quaisquer versões disponíveis.

Em seqüência as Figuras 18, 19 e 20 são apresentadas identificando algumas das diferenças citadas.



Figura 18 - Itens de menu Cadastros para usuários Administradores.
Fonte: Autoria Própria



Figura 19 - Itens de menu Cadastros para usuários Analistas.
Fonte: Autoria Própria



Figura 20 - Itens de menu Cadastros para usuários Leitores.
Fonte: Autoria Própria

Na opção de menu “Usuário >Perfil” o usuário pode manter seu próprio cadastro, alterar campos nome, e-mail, projeto default, nível, usuário, senha e foto.

A Figura 21 representa a tela de cadastro e edição de perfil de usuário.

Figura 21 - Tela de cadastro e manutenção de perfil de usuário.
Fonte: Autoria Própria

No item “Mensagens Recebidas” é apresentada uma listagem de todas as mensagens recebidas para o usuário corrente, ficam dispostas nessa interface os seguintes campos informativos: quantidade de mensagens recebidas, ações, título, data, de e status. Os campos informativos em nível de linha são links que direcionam a mensagem correspondente para visualizá-la. Nessa tela tem uma opção de cadastrar nova mensagem, que se clicada direciona o sistema para a interface para a confecção de nova mensagem.

A Figura 22 representa a interface de Mensagens recebidas.

(9) Mensagens Recebidas				
Ações	Título	Data	De	Status
Visualizar	Re: teste de mensagem 1	2014-04-09 01:38:59	Usuario Analista	Respondido
Visualizar	Re: Re: Re: teste de mensagem 1	2014-05-10 19:52:17	Jonas Centenaro	Pendente
Visualizar	Re: Re: Re: teste de mensagem 1	2014-05-10 19:52:17	Jonas Centenaro	Respondido

Figura 22 - Interface de Mensagens Recebidas em ordem de recebimento.
Fonte: Autoria Própria

No item de “Mensagens Enviadas” é apresentada uma listagem de todas as mensagens enviadas pelo usuário corrente, semelhante à interface anteriormente citada, ficam dispostos nessa interface os seguintes campos informativos: quantidade de mensagens enviadas, ações, título, data, para e status. Os campos informativos em nível de linha são links que direcionam a mensagem correspondente para visualizá-la. Nessa tela tem uma opção de cadastrar nova mensagem, que se clicada direciona o sistema para a interface para a confecção de nova mensagem.

A Figura 23 representa a interface de Mensagens recebidas.



(15) Mensagens Enviadas				
Acoes	Titulo	Data	Para	Status
Visualizar	Re: teste de mensagem 1	2014-04-09 01:41:53	Usuario Analista	Lido
Visualizar	teste para leitor	2014-04-09 01:52:02	Usuario Leitor	Lido
Visualizar	Re: teste de mensagem 1	2014-04-09 01:58:05	Usuario Analista	Respondido

Figura 23 - Interface de Mensagens Enviadas em ordem de envio.
Fonte: Autoria Própria

Na opção de Itens reservados é apresentada uma listagem de todos os documentos que estão reservados com o usuário corrente, essas reservas objetivam a privacidade do conteúdo temporariamente até que sua versão seja lançada.

Quando um documento esta em reserva com o usuário nenhum outro usuário consegue visualizar suas alterações sem antes o usuário de posse da reserva efetuar o lançamento do mesmo.

A Figura 24 representa a interface de listagem de reservas.



(5) Reservas			
Acoes	Documento	Projeto	Pasta
Visualizar	titulo de doc 1 novo	Projeto 1	teste outra pasta filha de arquivo
Visualizar	titulo	Projeto 1	teste outra pasta filha de arquivo

Figura 24 - Interface de listagem de reservas.
Fonte: Autoria Própria

E ainda no item de menu usuário há uma opção de “Logout / Sair”, que se clicado desvincula o usuário da sessão corrente, e o direciona para a interface de login.

A Figura 25 apresenta o encerramento de uma sessão, essa tela é apenas temporária até o carregamento da página de login e não disponibiliza nenhuma interação com o usuário.



Figura 25 - Encerrando sessão.
Fonte: Autoria Própria

No item de menu cadastros, a opção de novo usuário, se clicada direciona o sistema para a interface de cadastro de novo perfil de usuário. Essa interface é a mesma apresentada pelo item de menu “Usuário > Perfil” e pela Figura 21, com o único diferencial de que ao invés de manter um mesmo cadastro, é criado um novo.

Na opção “Listar usuários”, uma interface de listagem dos usuários é apresentada, ficam dispostos nessa interface os seguintes campos informativos: quantidade de usuários, ações, nome, nível, projeto default, usuário, senha (apenas asteriscos representando o tamanho da senha) e e-mail. Nessa tela tem uma opção de cadastrar novo usuário, que se clicada direciona o sistema para uma interface para o cadastro de um novo perfil.

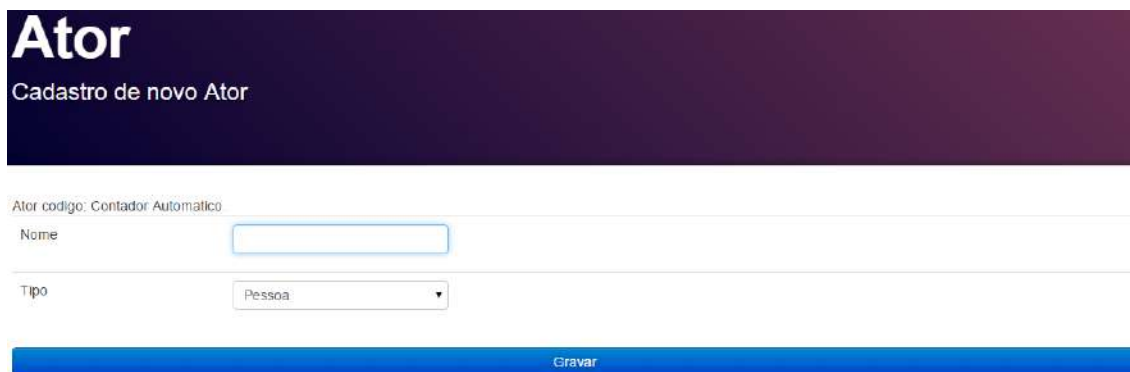
A Figura 26 apresenta a interface de listagem de usuários.

(8) Usuarios						
Acoes	Nome	Nivel	Projeto Default	Usuario	Senha	Email
Visualizar	Usuario Leitor	Leitor	Projeto 1	leitor	*****	email2@geres.com.br

Figura 26 – Interface de listagem de usuários.
Fonte: Autoria Própria

A opção Novo ator, se clicada direciona o sistema para uma interface para cadastro de novos atores, como entrada de dados, o usuário pode preencher os seguintes campos: nome e tipo, sendo que o campo “tipo” possibilita escolher entre pessoa, hardware ou software, cujos propósitos respectivamente são cadastrar um ator que represente um humano (usuário, cliente, entre outros), um objeto físico (produto, impressora, carro, entre outros) e objeto virtual (programa, imagem, vídeo, entre outros). Nessa tela possui um botão de gravar que ao clicado procede com a gravação dos dados em banco.

A Figura 27 representa a tela de cadastro de ator.



Ator código: Contador Automático

Nome

Tipo

Gravar

Figura 27 - Interface de cadastro de ator.
Fonte: Autoria Própria

No item listar atores é apresentado uma listagem de todos os atores que estão cadastrados no sistema. Ficam dispostos nessa interface os seguintes campos informativos: quantidade de atores, ações, nome e tipo. Os campos informativos em nível de linha são links que direcionam ao cadastro do ator correspondente. Nessa tela tem uma opção de cadastrar novo ator, que se clicada direciona o sistema para a interface para o cadastro de um novo ator.

A Figura 28 representa a interface de listagem de atores.

Atores
Listagem de Atores.

(3) Atores

Acoes	Nome	Tipo
Visualizar	Sistema	Software
Visualizar	Usuario	Pessoa
Visualizar	Impressora	Hardware

Cadastrar novo Ator

Figura 28 - Interface de listagem de atores.
Fonte: Autoria Própria

A opção novo projeto, se clicada direciona o sistema para uma interface de cadastro de projeto, a título de apenas identificação. Como entrada de dados, o usuário pode preencher o campo nome, e visualizar valor de data que será registrado. Nessa tela possui um botão de gravar que ao clicado procede com a gravação dos dados em banco.

A Figura 29 representa a tela de cadastro de projeto.

Projeto
Cadastro de novo Projeto

Projeto codigo: Contador Automatico.

Nome:

Data:

Gravar

Figura 29 - Interface de cadastro de projeto.
Fonte: Autoria Própria

No item listar projetos é apresentado uma listagem de todos os projetos que estão cadastrados no sistema. Ficam dispostos nessa interface os seguintes campos informativos: quantidade de projetos, ações, nome e data. Os campos informativos em nível de linha são links que direcionam ao cadastro do projeto correspondente. Nessa tela tem uma opção de cadastrar novo projeto, que se clicada direciona o sistema para a interface de cadastro de um novo projeto.

A Figura 30 representa a interface de listagem de projetos.

(3) Projetos		
Acoes	Nome	Data
Visualizar	Projeto 1	2014-03-13
Visualizar	Projeto 2	2014-04-27
Visualizar	Projeto 3	2014-03-20

Cadastrar novo Projeto

Figura 30 - Interface de listagem de projetos.
Fonte: Aatoria Própria

A opção nova pasta, se clicada direciona o sistema para uma interface de cadastro de pasta. Como entrada de dados, o usuário pode preencher os campos nome, “subpasta de” e status. Lembrando que esse cadastro considera o projeto default para relacionar a pasta a seu cadastro. Nessa tela possui um botão de gravar que ao clicado procede com a gravação dos dados em banco.

A Figura 31 representa a tela de cadastro de pasta.

Pasta
 Cadastro de nova Pasta

Pasta código: Contador Automático

Nome

Sub pasta de

Status

Gravar

Figura 31 - Interface de cadastro de pasta.
Fonte: Aatoria Própria

No item listar pastas é apresentado uma listagem de todas as pastas que estão cadastrados no sistema. Ficam dispostos nessa interface: quantidade de pastas, pastas em disposição de formato de árvore hierárquica, documentos, editar, criar novo documento de especificação de software e cadastrar nova pasta que se clicada direciona o sistema para a interface de cadastro de uma nova pasta.

A Figura 32 representa a interface de listagem de pastas.

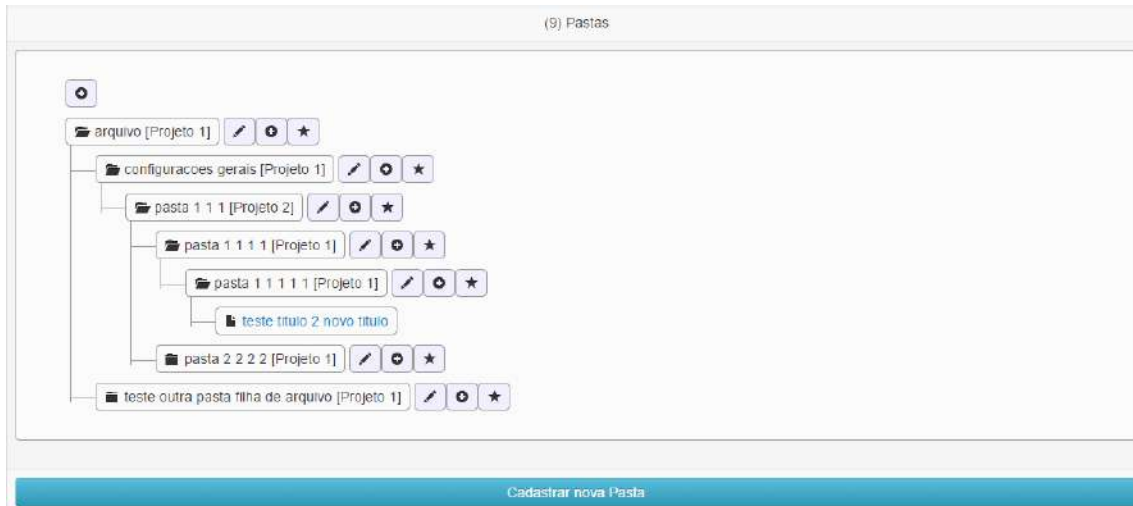


Figura 32 - Interface de listagem de pastas e documentos.
Fonte: Autoria Própria

Ao selecionar a opção cadastrar um novo documento de especificação de software uma tela de confirmação é apresentada conforme a Figura 33.



Figura 33 - Interface de confirmação para cadastro de novo documento de requisitos.
Fonte: Autoria Própria

Nessa etapa apenas a opção de criar novo documento esta disponível a fim de confirmação, e conforme o informativo presente nessa tela, após selecionar a opção de criar novo documento, um novo documento será criado, permanecerá reservado para alterações ao usuário até sua liberação e o sistema será direcionado para a interface de manutenção do documento.

Na listagem de pastas é possível localizar os documentos previamente publicados e acessá-los através de um clique. Os documentos podem ser localizados também através do campo de pesquisa disponível na barra de pesquisa.

A pesquisa pode ser efetuada pelo código do documento ou por parte do título.

A Figura 34 representa um resultado quando uma busca é feita pelo campo de pesquisa disponível na barra superior.

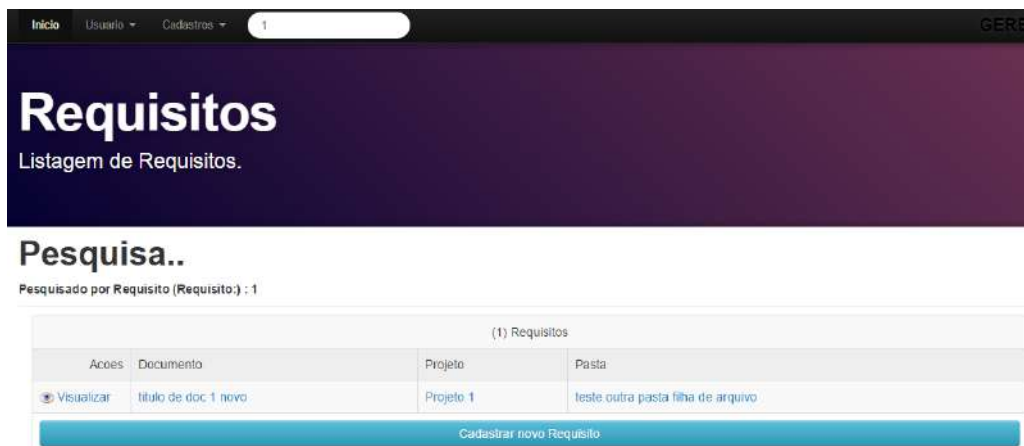


Figura 34 - Resultado de busca pelo campo de pesquisa.
Fonte: Aatoria Própria

Na Figura anterior os campos informativos em nível de linha são links que direcionam ao documento correspondente, interface apresentada na Figura 35.



Figura 35 - Interface de documento de especificação de software.
Fonte: Aatoria Própria

Ao acessar a interface dos documentos de especificação de software algumas validações e identificações são efetuadas. É validado o usuário autenticado, identificando se o mesmo possui o documento em posse de reserva e a interface é totalmente apresentada em modo de edição conforme parte exemplar da Figura 36.



Figura 36 - Interface do documento em modo de edição.
Fonte: Aatoria Própria

Quando não identificado reserva do documento, é apresentada uma opção no topo do corpo do documento que possibilita a reserva do documento caso o usuário seja um administrador ou analista, conforme Figura 37.

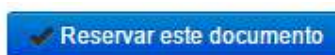


Figura 37 - Opção de reservar documento.
Fonte: Aatoria Própria

Também no topo é apresentado de forma resumida o caminho onde se encontra o documento em questão, conforme Figura38.

arquivo / configuracoes gerais / pasta 1 1 1 / pasta 1 1 1 1 / pasta 1 1 1 1 1

Figura 38 - Identificador de local do documento.
Fonte: Aatoria Própria

Na seqüência um guia fica disponível sempre flutuante na lateral esquerda da tela, o mesmo pode identificar a área do documento que esta sendo alterada ou lida, e também possibilita o acesso imediato de determinada área a escolha do usuário ao comando de um simples clique, a Figura 39 representa o guia.

Propriedades Básicas	
Capa	>
Histórico de Revisões	>
Nome / Breve Descrição	>
Pré-Condições	>
Pós-Condições	>
Atores	>
Fluxo de Eventos	>
Requisitos	>
Requisitos Cancelados	>
Diagramas	>
Protótipos de Interface	>
Referências	>

Figura 39 - Guia dos tópicos do documento.
Fonte: Autoria Própria

Ao lado direito da tela é apresentado todo o corpo e conteúdo do documento em questão, essa interface quando em modo de edição ao identificar alterações do usuário disponibiliza opções de salvar tópico a tópico alterado. Os tópicos seguem a ordem em que é apresentado no guia, e conforme segue:

Propriedades básicas: Esse tópico pode apresentar informativo de alerta quando algum usuário estiver lendo o documento e o mesmo estiver em edição de outro usuário, com a seguinte mensagem: “Este documento está em edição pelo usuário NOME DO USUÁRIO”. Ficam dispostos nesse tópico também o status, e a pasta, conforme a Figura 40.

Propriedades Básicas

Atenção!

Este documento está em edição pelo usuário JONAS CENTENARO.

Status: Ativo

Pasta: teste outra pasta filha de arquivo

Figura 40 - Tópico de propriedades básicas.

Fonte: Autoria Própria

A opção de pastas quando em modo de edição abre uma interface para seleção de pasta, nessa interface apenas pastas são apresentadas, conforme a Figura 41.

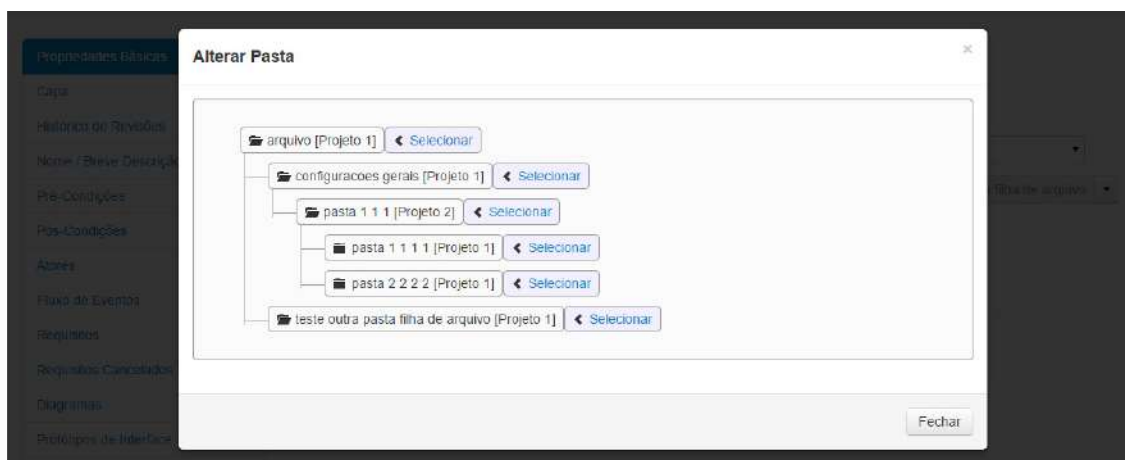


Figura 41 - Interface para alteração de pasta do documento.
Fonte: Autoria Própria

Capa: Esse tópico apresenta os seguintes valores: Título, projeto, usuário que criou o documento, e versão corrente. Quando em modo de edição apenas o título fica editável. A Figura 42 apresenta a capa em modo de edição.

Capa

titulo de doc 1 novo

Projeto / Sistema: *Projeto 1*
Jonas Centenaro
Versão: 8

Figura 42 - Tópico de capa do documento.
Fonte: Autoria Própria

Histórico de Revisões: Esse tópico apresenta os seguintes valores: Quantidade de revisões, data, versão, descrição, usuário e quando em modo de leitura ação de visualizar versões anteriores (Opção identifica a versão que esta sendo lida com valor “Visualizando”). Em modo de edição é disponibilizado um

campo de texto, para informar uma descrição, devendo essa ser a última ação do usuário em edição, pois ao gravar essa alteração, uma versão do documento é publicada e eliminada a reserva corrente, justificando a identificação do botão disponibilizado: “Concluir alterações e lançar versão deste documento”. A Figura 43 apresenta a interface de histórico de revisões em modo de edição.

Histórico de Revisões

(0) Revisões			
Data	Versão	Descrição	Usuário
24/10/2014 00:44	1	<input type="text" value="Criação deste documento"/>	Jonas Centenaro

Concluir alterações e lançar versão deste documento

Figura 43 - Tópico de histórico de revisões.
Fonte: Aatoria Própria

Nome / Breve Descrição: Esse tópico apresenta apenas o valor do nome e/ou breve descrição. A Figura 44 apresenta a interface de Nome / Breve descrição em modo de leitura.

Nome / Breve Descrição

Sem Descrição

Figura 44 - Tópico de nome breve descrição.
Fonte: Aatoria Própria

Pré-Condições: Esse tópico apresenta apenas o valor das pré-condições. A Figura 45 apresenta a interface de pré-condições em modo de leitura.

Pré-Condições

Sem Pré-Condição

Figura 45 - Tópico de pré-condições.
Fonte: Aatoria Própria

Pós-Condições: Esse tópico apresenta apenas o valor das pós-condições. A Figura 46 apresenta a interface de pós-condições em modo de leitura.

Pós-Condições

Sem Pós-Condição

Figura 46 - Tópico de pós-condições.
Fonte: Aatoria Própria

Atores: Esse tópico apresenta os seguintes valores: atores selecionados. Quando em modo de edição apresenta também um seletor de atores. Caso queira relacionar mais atores ao documento, também é possível remover relação ao clicar no ator selecionado. A Figura 47 apresenta a interface em modo de edição.

Atores

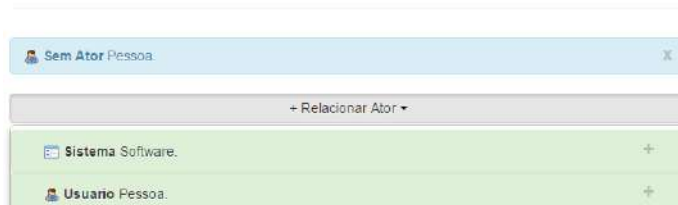


Figura 47 - Tópico de atores.
Fonte: Aatoria Própria

Fluxo de Eventos: Esse tópico apresenta os seguintes valores: Fluxos básicos e fluxos alternativos, ambos com título, descrição e opções de excluir e adicionar. A Figura 48 apresenta a interface de fluxo de eventos em modo de edição.

Fluxo de Eventos



Figura 48 - Tópico de fluxo de eventos.
Fonte: Aatoria Própria

Requisitos: Esse tópico apresenta os seguintes valores: Título, descrição e opções de cancelar e adicionar. A Figura 49 apresenta a interface de requisitos em modo de edição.

Requisitos

teste de primeiro requisito ativo

descricao de teste de primeiro requisito

Figura 49 - Tópico de requisitos.
Fonte: Autoria Própria

Requisitos Cancelados:Esse tópico apresenta os seguintes valores: Título, descrição e opção de ativar. A Figura 50 apresenta a interface de requisitos cancelados em modo de edição.

Requisitos Cancelados

Sem Título

Sem Descriçãoasfasdfasdfasd

Figura 50 - Tópico de requisitos cancelados.
Fonte: Autoria Própria

Diagramas:Esse tópico apresenta os seguintes valores: Título, imagem, opções de alterar diagramas (antes de alterar este deve ser marcado), excluir e adicionar. A Figura 51 apresenta a interface de diagramas em modo de edição.

Diagramas

digrama 1 teste titulo

Escolher arquivo Nenhum arquivo selecionado

Figura 51 - Tópico de diagramas.
Fonte: Autoria Própria

Protótipos de Interface:Esse tópico apresenta os seguintes valores: Título, imagem, opções de alterar protótipos (antes de alterar este deve ser marcado), excluir e adicionar. A Figura 52 apresenta a interface de protótipos em modo de edição.

Protótipos de Interface



Figura 52 - Tópico de protótipos de interface.
Fonte: Autoria Própria

Referências:Esse tópico apresenta os seguintes valores: Título, imagem ou texto, opções de alterar referências (antes de alterar este deve ser marcado), excluir, adicionar e opções de escolha única entre texto ou imagem. A Figura 53 apresenta a interface de referências em modo de edição.

Referências

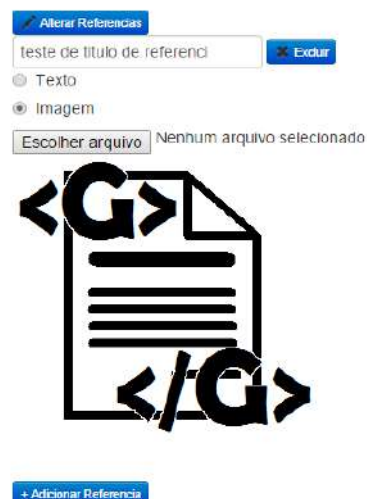


Figura 53 - Tópico de referências.
Fonte: Autoria Própria

4.2 Desenvolvimento do sistema móvel

Para o desenvolvimento da aplicação móvel, primeiramente foi instalado e configurado o ambiente de desenvolvimento, como primeiro passo, efetuei o download do aplicativo Eclipse com Android SDK. Na seqüência configurei o acesso a rede interna ao servidor do WAMP, seguindo os passos: WAMP > apache > httpd.conf, localizar a linha referente a permissão de acesso “Deny from all” e trocá-la por “Allow from all”, e para testar foi utilizado um dispositivo móvel com acesso a rede interna e no navegador de internet do mesmo foi digitado o endereço de IP da máquina com o servidor apache.

Na seqüência foi criado um projeto android no Eclipse, configurado para ser compatível com a versão de android api 7, android mínimo versão 2.2.

No contexto desse projeto vários arquivos foram criados, e alguns foram elencados como os principais e são apresentados com mais detalhes e em listagens no anexo C.

No decorrer do desenvolvimento do sistema móvel, vários testes foram efetuados a fim de garantir o funcionamento do mesmo. Para isso foi utilizado o recurso de Android Virtual Device, com ele foi criado um dispositivo virtual que possibilitou os testes na própria máquina. Também foram efetuados alguns testes em dispositivos reais, e o aplicativo possuiu o mesmo comportamento. Estando então o aplicativo homologado para utilização em dispositivos móveis a modo de leitura, sendo que alguns recursos nativos ainda são incompatíveis como o recurso de selecionar imagem do HTML, não suportado pela versão Android utilizada.

A Figura 54 apresenta a interface da AVD – Android Virtual Device sendo utilizada.



Figura 54 - Interface AVD - Android Virtual Device.
Fonte: Autoria própria.

4.2.1 Apresentação do aplicativo Android

Neste tópico serão apresentadas algumas telas da aplicação Android, produto secundário do sistema objeto principal deste trabalho.

Como esse sistema atua como um intermediador, no dispositivo móvel como um navegador de internet incorporado na aplicação, as telas do sistema principal terão algumas mudanças de layout, pelo fato de o *Bootstrap* possuir recursos responsivo, se adéqua o visual dependendo to tamanho da tela do dispositivo.

Importante lembrar que a regra de funcionamento da aplicação é a mesma, e alguns recursos de edição como o carregamento de imagens poderão não funcionar nos dispositivos móveis.

A seguir uma seqüência de figuras correspondente as principais telas do sistema a título de exemplares serão apresentadas.

A Figura 55 apresenta a tela principal do sistema com menu recolhido.



Figura 55 - Tela principal do sistema em um dispositivo móvel.
Fonte: Autoria própria.

A Figura 56 apresenta a tela de perfil do usuário com menu expandido.

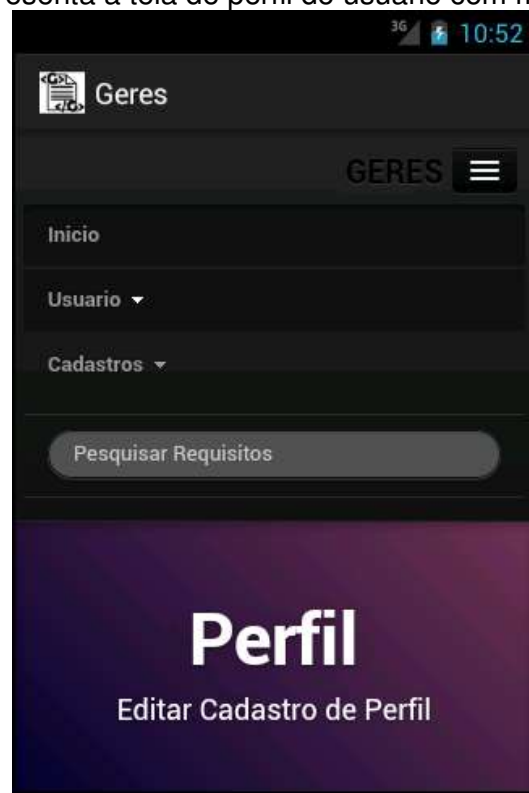


Figura 56 - Tela com menu expandido em um dispositivo móvel.
Fonte: Autoria própria.

A Figura 57 apresenta guias do documento de especificação de software.



Figura 57 - Guias de documento em um dispositivo móvel.
Fonte: Autoria própria.

A Figura 58 apresenta parte do documento de especificação de software.

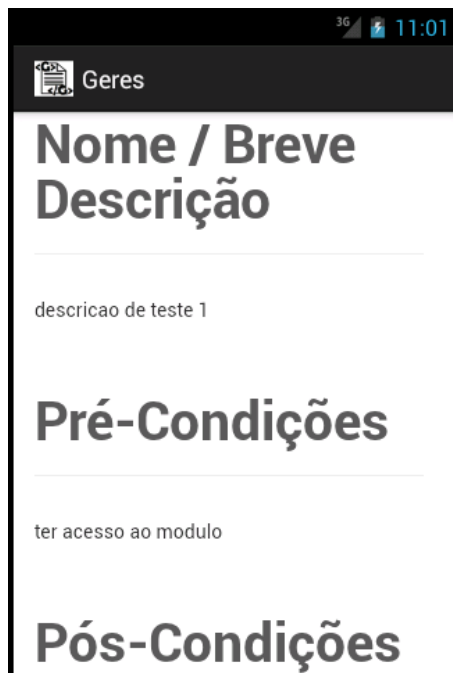


Figura 58 - Parte do corpo de documento em um dispositivo móvel.
Fonte: Autoria própria.

5. CONCLUSÃO

Neste trabalho foram abordadas algumas características de um documento de especificação de software, e com base na análise levantada foram encontrados alguns problemas e como solução proposta foi desenvolvido um sistema para gerenciamento desses documentos, a fim de facilitar o acesso pelos envolvidos, sejam eles administradores, analistas ou quaisquer leitores desses documentos.

Esse trabalho teve como foco principal o desenvolvimento de uma aplicação web, no qual foi possível aplicar as técnicas e linguagens de programação para internet e dispositivos móveis, no sistema desenvolvido o acesso a todos os recursos se efetiva através de aplicações que possibilitam a navegação na internet.

Com a utilização da ferramenta *Bootstrap* foi desenvolvido um produto que além de possuir um visual agradável fornece características responsivas, ou seja, a interface se adapta dependendo do tamanho da tela do dispositivo acessado, possibilitando dessa forma a elaboração e produção de uma segunda aplicação para dispositivos móveis com sistema operacional *Android*.

Após a implantação do sistema em ambiente real e com a utilização dessas aplicações o processo de confecção dos documentos de especificação de software e o próprio acesso aos mesmos tornaram-se rápidos e flexíveis, podendo o usuário acessar por diversos dispositivos e onde estiver que possuam acesso à internet. Usuários relataram sentir um maior conforto quanto à utilização dos softwares desenvolvidos, um dos detalhes apontados é a ausência de poluição gráfica, em comparação com outras aplicações antes utilizadas, diversos artefatos gráficos que nunca eram utilizados, poluíam a interface gráfica. Também a partir desse, um padrão pode ser seguido e o espaço em disco utilizado para o armazenamento foi reduzido já que conteúdos antigos inalterados se manterão únicos, diferentemente dos arquivos de texto que independente o conteúdo, a cada nova versão lançada, um novo arquivo é gerado e conseqüentemente o conteúdo anterior é 100% replicado para a nova versão.

A tabela 2 apresenta algumas comparações realizadas após a utilização do sistema desenvolvido.

	Sistema Desenvolvido	Sistema Anteriormente utilizado
Acesso por dispositivos móveis	Sim	Não
Padrão para estrutura do documento	Sim	Não
Requer instalação de executáveis do sistema	Não	Sim
Leitura de documentos em versões anteriores	Sim	Sim
Indicação de documento em edição	Sim	Não
Requisito mínimo	Qualquer navegador instalado	Executável específico instalado
Tamanho ocupado mínimo	Independente quantidade de documentos: ~12KB	A cada documento: ~9KB

Tabela 2 - Comparativo de Sistemas

Fonte: Autoria própria.

O Anexo D representa a impressão de um documento gerado pelo sistema, em utilização no ambiente real.

O sistema desenvolvido possibilita o gerenciamento desses documentos de quaisquer projetos de software, o acesso ao mesmo pode ser feito por endereço web ou pela aplicação móvel, atualmente esta sob o domínio interno, e com levantamentos de estudos referentes à distribuição pública, para possibilitar o uso por outras organizações.

5.1 Problemas Encontrados

Durante o desenvolvimento deste trabalho, algumas dificuldades foram encontradas, como problemas de toque nos menus suspensos do *Twitter Bootstrap* em telas de toque.

Foi desenvolvido o sistema, e em alguns testes foi possível identificar que no aplicativo móvel, o componente de menu, ao expandir, os itens não estavam acessíveis, não respondiam a ação de clique, que deveria, ao tocar nos itens.

Com base em pesquisas foi visto que o problema realmente existe, segundo David Cochran (2012) essa questão já esta sendo vista por uma equipe e o problema poderá ser corrigido em versões futuras do *Bootstrap*.

5.2 Trabalhos Futuros

Como trabalhos futuros, pode-se apontar:

- Desenvolvimento de integração da aplicação web com frameworks de comparação de texto: permitir selecionar ao mínimo duas versões do mesmo documento e em visão vertical conseguir compará-las identificando as diferenças.
- Criação de cadastro de tipos de documento: poder personalizar o corpo do documento e relacionar na criação de um documento um tipo de documento, sendo este o layout do mesmo.
- Exportar documentos: permitir com a integração de alguma ferramenta a exportação do documento para formatos como PDF, DOC ou HTML.
- Rastreabilidade e relação entre documentos: permitir inserir ligações a outros documentos nos conteúdos dos documentos, capaz de efetuar acesso rápido aos documentos citados, por meio de links ou botões de acesso, inseridos ao meio do texto. Automaticamente será possível estabelecer uma rastreabilidade, possibilitando uma análise de impacto nas modificações futuras.

REFERÊNCIAS

ALEX_SENTCHA'S BLOG. **Android – Exchange data with Siemens PLC.** Disponível em: <<http://alexsentcha.wordpress.com/libnodave-exchange-data-with-siemens-plc-2/android-exchange-data-with-siemens-plc/>>. Acesso em: 02 Ago. 2014.

ALSHANETSKY, Iliia. **Usage Statistics.** Disponível em: <<http://webadvent.org/2010/usage-statistics-by-ilia-alshanetsky>> Acesso em: 18 Jul. 2014.

ANDROID_SDK. **SDK.** Disponível em: <<http://developer.android.com/sdk/index.html>>. Acesso em: 08 Dez. 2014.

ANDROID ON BOARD. **Meu primeiro aplicativo Android.** Disponível em: <<http://androiddevbr.wordpress.com/2013/08/13/meu-primeiro-aplicativo-android/>> Acesso em: 16 Jul. 2014.

ARAUJO, Igor. **Vantagens do controle de versão no desenvolvimento ágil.** Disponível em: <<http://blog.myscrumhalf.com/2011/10/vantagens-do-controle-de-versao-no-desenvolvimento-agil/>> Acesso em 31 Jul. 2014.

ASTAH_DOWNLOAD. **Download.** Disponível em: <<http://astah.net/download>>. Acesso em: 08 Dez. 2014.

BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL.** São Paulo: Casa do Código, 2014.

BLASCHEK, José Roberto. **Gerência de Requisitos: O principal problema dos projetos de software.** Disponível em: <<http://www.bfpug.com.br/islig-rio/Downloads/Ger%C3%Aancia%20de%20Requisitos-o%20Principal%20Problema%20dos%20Projetos%20de%20SW.pdf>>. Acesso em: 28 Jan. 2014.

BOOTSTRAP. **About.** Disponível em: <<http://getbootstrap.com/about/>> Acesso em: 20 Jul. 2014.

BOOTSTRAP_COMPONENTES. **Componentes.** Disponível em: <<http://getbootstrap.com/components/>>. Acesso em: 08 Dez. 2014.

BOOTSTRAP_GET. **Get.** Disponível em: <<http://getbootstrap.com/>>. Acesso em: 08 Dez. 2014.

CHROME_DOWNLOAD. **Download.** Disponível em: <<http://www.google.com/intl/pt-BR/chrome/browser/>>. Acesso em: 08 Dez. 2014.

COCHRAN, David. **Fix Twitter Bootstrap's Dropdown Menus in Touch Screens.** Disponível em: <<http://alittlecode.com/fix-twitter-bootstraps-dropdown-menus-in-touch-screens/>> Acesso em: 15 de Agosto de 2014.

DEITEL, Paul; DEITEL, Harvey. **Java: Como Programar**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

DIAS, Eduardo Campos. **O atual mercado dos sistemas operacionais para dispositivos móveis**. Disponível em: <<http://dificuldadezero360.blogspot.com.br/2014/02/o-atual-mercado-dos-sistemas.html>> Acesso em: 08 Dez. 2014.

D3NOOB.ORG. **Getting started using Bootstrap with d3.js**. Disponível em: <<http://www.d3noob.org/2013/07/getting-started-using-bootstrap-with.html>> Acesso em: 20 Jul. 2014.

ECLIPSE_DOWNLOAD. **Download**. Disponível em: <<https://www.eclipse.org/downloads/>>. Acesso em: 08 Dez. 2014.

EQUIPE BAIXAKI. **Notepad++**. Disponível em: <<http://www.baixaki.com.br/download/notepad-.htm>>. Acesso em: 03 Ago. 2014.

FERRARI, Fabrício Augusto. **Crie banco de dados em MySQL**. São Paulo: Digerati Books, 2007.

FREITAS, Daniel Tannure Menandro de. **Análise Comparativa entre Sistemas de Controle de Versões**. 46p. Monografia, UFJF, Juiz de Fora, Brasil, 2010.

GITHUB. **Consulta**. Disponível em: <<https://github.com/search?q=desc&q=stars%3A%3E1&s=stars&type=Repositories>> Acesso em: 18 Jul. 2014.

GOOGLE. **Saiba mais sobre o navegador Google Chrome**. Disponível em: <<http://www.google.com/intl/pt-BR/chrome/browser/features.html>>. Acesso em: 02 Ago. 2014.

HIGA, Paulo. **Android ultrapassa 50% de market share; Samsung vende mais**. Disponível em: <<http://tecnoblog.net/97184/android-market-share/>>. Acesso em: 15 Jul. 2014.

LECHETA, Ricardo R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com Android SDK**. 3. ed. São Paulo: Novatec Editora, 2013.

MORAES, Janaína Bedani Dixon. **Engenharia de Software 2 - Técnicas para levantamento de Requisitos**. Disponível em: <<http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151>>. Acesso em: 04 Nov. 2014.

MURTA, Leonardo Gresta Paulino. **Gerência de configuração no desenvolvimento baseado em componentes**. 213p. Tese de Doutorado, COPPE, UFRJ, Rio de Janeiro, Brasil, 2006.

NOTEPAD_DOWNLOAD. **Download.** Disponível em: < <http://notepad-plus-plus.org/download/> >. Acesso em: 08 Dez. 2014.

ORACLE. **MySQL – O Banco de Dados de Código Aberto Mais Popular do Mundo.** Disponível em: <<http://www.oracle.com/br/products/mysql/overview/index.html>> Acesso em: 19 Jul. 2014.

PHP.NET. **História do PHP.** Disponível em: < http://php.net/manual/pt_BR/history.php.php> Acesso em: 17 Jul. 2014.

XCARECROWS.COM. **The Rational RequisitePro interface.** Disponível em: <http://www.xcarecrows.com/eclipse/doc/Xcarecrows4SMV/current_version/html/userGuide/RequisitePro%20-%20interface.html> Acesso em: 20 Jul. 2014.

WAMP_DOWNLOAD. **Download.** Disponível em: < <http://www.wampserver.com/en/#download-wrapper> >. Acesso em: 08 Dez. 2014.

WILDT, Daniel de Freitas; LACERDA, Guilherme Silva de. **Conhecendo o eXtreme Programming (XP).** Disponível em: <<http://pt.slideshare.net/dwildt/conhecendo-o-extreme-programming>> Acesso em: 01 Ago. 2014.

WTHREEX.COM. **Conceitos: Gerenciamento de Requisitos.** Disponível em: <http://www.wthreex.com/rup/portugues/process/workflow/requirem/co_rm.htm> Acesso em: 31 Jul. 2014.

Anexo A – Dicionário de dados.

Coluna	Tipo	Nulo	Omissão	Comentários
Idator	Int	Não		PK Auto_Increment
nomeator	varchar(50)	Não		
Tipo	char(1)	Não	P	P pessoa H hardware S software

Quadro 16 – Tabela ator: Responsável por armazenar os dados dos atores dos documentos.
Fonte: Autorial Própria

Coluna	Tipo	Nulo	Omissão	Comentários
iddocator	int	Não		PK Auto_Increment
idator	int	Não		FK com ator.idator
iddoc	int	Não		FK com docs.iddoc
idversao	int	Não		
idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		

Quadro 17 - Tabela docator: Responsável por registrar as alterações de atores dos documentos.
Fonte: Autorial Própria

Coluna	Tipo	Nulo	Omissão	Comentários
idcapa	Int	Não		PK Auto_Increment
iddoc	Int	Não		FK com docs.iddoc
idversao	Int	Não		
titulo	varchar(100)	Não		
idusuario	Int	Não		FK com usuario.idusuario
dtalt	datetime	Não		

Quadro 18 - Tabela doccapa: Responsável por registrar as alterações da capa dos documentos.
Fonte: Autorial Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Iddescricao	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Sim	NULL	
Descrição	varchar(1000)	Não		
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		

Quadro 19 - Tabela docdescricao: Responsável por registrar as alterações da descrição dos documentos.
Fonte: Autorial Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Iddiagrama	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Não		
Titulo	varchar(100)	Não		
Status	char(1)	Não	A	A ativo E excluido
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		
iddiagramapai	int	Sim	0	

Quadro 20 - Tabela docdiagrama: Responsável por registrar as alterações dos diagramas dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idfluxo	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Não		
Titulo	varchar(100)	Não	Sem Titulo	
Descrição	varchar(1000)	Não	Sem Descrição	
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		
Tipo	char(1)	Não	B	
Situação	char(1)	Não	A	A ativo E excluido
Idfluxopai	int	Sim	0	

Quadro 21- Tabela docfluxo: Responsável por registrar as alterações dos fluxos dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Iddochistorico	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Não		
Descrição	varchar(100)	Não	Criação/Alteração deste Documento	
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		

Quadro 22 - Tabela dochistorico: Responsável por registrar as alterações do histórico dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
idposcondicao	int	Não		PK Auto_Increment
iddoc	int	Não		FK com docs.iddoc
ldversao	int	Não		
Poscondicao	varchar(1000)	Não		
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		

Quadro 23 - Tabela docposcondicao: Responsável por registrar as alterações da as pós-condições dos documentos.

Fonte: Aatoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idprecondicao	int	Não		PK Auto_Increment
iddoc	int	Não		FK com docs.iddoc
ldversao	int	Não		
Precondição	varchar(1000)	Não		
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		

Quadro 24 - Tabela docprecondicao: Responsável por registrar as alterações das pré-condições dos documentos.

Fonte: Aatoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
idpropbasica	int	Não		PK Auto_Increment
iddoc	int	Não		FK com docs.iddoc
ldversao	int	Não		
status	char(1)	Não		A - ativo - Inativo
idpasta	int	Não		FK com pastas.idpasta
idusuario	int	Não		FK com usuario.idusuario
dtalt	datetime	Não		

Quadro 25 - Tabela docpropbasicas: Responsável por registrar as alterações das propriedades básicas dos documentos.

Fonte: Aatoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idprototipo	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Não		
Titulo	varchar(100)	Não		
Status	char(1)	Não	A	A ativo E excluído
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		
idprototipopai	int	Sim	0	

Quadro 26 - Tabela docprototipo: Responsável por registrar as alterações dos protótipos dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idreferencia	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Não		
Titulo	varchar(100)	Não		
Descrição	varchar(1000)	Não		
Status	char(1)	Não	A	A ativo E excluído
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		
idreferenciapai	int	Sim	0	

Quadro 27 - Tabela docreferencia: Responsável por registrar as alterações das referências dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idrequisito	int	Não		PK Auto_Increment
Iddoc	int	Não		FK com docs.iddoc
Idversao	int	Não	0	
Titulo	varchar(100)	Não		
Descrição	varchar(1000)	Não		
Situação	char(1)	Não	A	A ativo E excluído
Idusuario	int	Não		FK com usuario.idusuario
Dtalt	datetime	Não		
idrequisitopai	int	Sim	0	

Quadro 28 - Tabela docrequisito: Responsável por registrar as alterações dos requisitos dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Iddoc	int	Não		PK Auto_Increment
idusuariocriador	int	Não		FK com usuario.idusuario
Dtcriacao	datetime	Não		
idusuarioresevado	int	Sim	NULL	FK com usuario.idusuario
Dtreserva	datetime	Sim	NULL	
Idversao	int	Não	0	

Quadro 29 - Tabela docs: Responsável por registrar os documentos e controlar as reservas dos mesmos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idimagem	int	Não		PK Auto_Increment
caminhoimagem	varchar(100)	Não		
Iddotipo	int	Sim	NULL	
Tipo	char(1)	Sim	U	U usuario

Quadro 30 - Tabela imagens: Responsável por registrar as alterações das imagens dos documentos.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idmensagem	int	Não		PK Auto_Increment
Titulo	varchar(50)	Não		
Idusuariorigem	int	Não		FK com usuario.idusuario
idusuariodestino	int	Não		FK com usuario.idusuario
Data	datetime	Não		
Status	char(1)	Não	P	P pendente - L lido
Texto	varchar(500)	Não		

Quadro 31 - Tabela mensagens: Responsável por registrar as mensagens trocadas entre os usuários.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
Idpasta	int	Não		PK Auto_Increment
Nomepasta	varchar(50)	Não		
Idpastapai	int	Não		
Idusuario	int	Não		FK com usuario.idusuario
Idprojeto	int	Não		FK com projetos.idprojeto
Status	char(1)	Não	A	A ativo inativo

Quadro 32 - Tabela pastas: Responsável por registrar as pastas.

Fonte: Autoria Própria

Coluna	Tipo	Nulo	Omissão	Comentários
idprojeto	Int	Não		PK Auto_Increment
nomeprojeto	varchar(50)	Não		
idusuario	Int	Não		FK com usuario.idusuario
data	Date	Não		

**Quadro 33 - Tabela projetos: Responsável por registrar os projetos.
Fonte: Autoria Própria**

Coluna	Tipo	Nulo	Omissão	Comentários
idusuario	Int	Não		PK Auto_Increment
nome	varchar(50)	Não		
nivel	Int	Não	3	1 admin 2 analista 3 leitor 4 bloqueado
projeto default	Int	Não		FK com projetos.idprojeto
usuarioo	varchar(50)	Não		
senhaa	varchar(50)	Não		
email	varchar(100)	Sim	NULL	
imagem	varchar(100)	Sim	NULL	
tamANHosenha	Int	Sim	NULL	

**Quadro 34 - Tabela usuario: Responsável por registrar os usuários.
Fonte: Autoria Própria**

Coluna	Tipo	Nulo	Omissão	Comentários
idrecuperaacesso	Int	Não		PK Auto_Increment
idusuario	Int	Não		FK com usuario.idusuario
codigounico	varchar(5)	Não		
dtalt	datetime	Não		

**Quadro 35 - Tabela usuario_recuperaacesso: Responsável por registrar as solicitações de recuperação de acesso dos usuários.
Fonte: Autoria Própria**

Anexo B – Listas de códigos desenvolvimento web

A seguir são apresentados alguns dos arquivos criados, e algumas listas de códigos, tidos como principais para o funcionamento do sistema:

conecta.php: Arquivo que proporciona a conexão com o banco de dados. Nele ficam relacionados informações de acesso aos dados, como o próprio banco de dados, endereço eletrônico, usuário e senha.

Listagem 1 - Listagem de código PHP para conexão com banco de dados.

```

1.  <?php
2.
3.  $host    = "localhost";    //ip ou endereço do servidor
4.
5.  $usuario = "usuario";     //usuário
6.
7.  $senha   = "senha";       //senha
8.
9.  $dbase   = "geresdb";     //banco que será usado nesta conexão
10.
11. $con = mysql_connect($host,$usuario,$senha);    //Efetua a conexão com o BD
12.
13. $db = mysql_select_db($dbase, $con) OR DIE(mysql_error());
    //Seleciona a base de dados
14.
15. mysql_query("SET NAMES 'utf8'", $con);
16.
17. if (!$con) {
18.     echo "Erro de conexão ao banco de dados!";
19.     exit();
20. }
21. ?>

```

Fonte: Autoria própria.

gravafoto.php: Arquivo responsável pelo carregamento da imagem para o servidor. Recebe a imagem do formulário de envio, monta uma nova nomenclatura de imagem e move a mesma para o destino programado.

Listagem 2 - Listagem de trecho de código PHP para envio de imagens para o servidor.

```

22.  //[...]
23.  //Recebe a imagem do formulario
24.  $foto = $_FILES["foto"];
25.
26.  // Pega extensão da imagem
27.  preg_match("/\.(gif|bmp|png|jpg|jpeg){1}$/i", $foto["name"], $ext);
28.
29.  // Gera um nome para a imagem
30.  $nome_imagem = $idregistro.'-'. $idimagem.'--'.$tipo.'" . $ext[1];
31.
32.  // Caminho de onde ficará a imagem
33.  $caminho_imagem = "../.../fotos/" . $nome_imagem;
34.
35.  // Faz o upload da imagem para seu respectivo caminho
36.  move_uploaded_file($foto["tmp_name"], $caminho_imagem);
37.  //[...]

```

Fonte: Autoria própria.

docscorpo.php: Arquivo responsável pela estrutura e organização do corpo do documento de especificação de software.

Listagem 3 - Listagem de trecho de código PHP para captura dos dados de cada tópico do corpo do documento.

```

38.  //[...]
39.  //Instancia objeto de regras.
40.  include('docsregra.php');
41.  $docsregra = new docsregra;
42.
43.  //>>>>>>>> Propriedades basicas
44.  $d_propbasicas = $docsregra -> of_get_dados_propbasicas($con, $iddoc,
$dpastaselecionada, $uslogado, $leitura, $idversao_propbasica);
45.  $idpastaselecionada = $d_propbasicas['idpasta'];
46.
47.  //>>>>>>>> Capa
48.  $d_capa = $docsregra -> of_get_dados_capa($con, $iddoc, $uslogado, $leitura,
$idversaoler);
49.
50.  //>>>>>>>> historico
51.  $d_historico = $docsregra -> of_get_dados_historico($con, $iddoc, $uslogado, $leitura);
52.
53.  $idversaodoc = 0;
54.  $sql = 'SELECT max(idversao) as idversao FROM docs WHERE docs.iddoc = '.$iddoc;
55.  $exec_sql = mysql_query($sql, $con);
56.  while($res = mysql_fetch_array($exec_sql)){
57.      $idversaodoc = $res['idversao'];
58.  }
59.
60.  //>>>>>>>> Nome breve descricao
61.  $d_nomedescricao = $docsregra -> of_get_dados_nomedescricao($con, $iddoc, $uslogado,
$leitura, $idversaoler);
62.
63.  //>>>>>>>> Pre condicoes
64.  $d_precondicao = $docsregra -> of_get_dados_precondicao($con, $iddoc, $uslogado,
$leitura, $idversaoler);
65.
66.  //>>>>>>>> Pos Condicoes
67.  $d_poscondicao = $docsregra -> of_get_dados_poscondicao($con, $iddoc, $uslogado,
$leitura, $idversaoler);
68.
69.  //>>>>>>>> Atores
70.  $d_atores = $docsregra -> of_get_dados_atores($con, $iddoc, $uslogado, $leitura,
$idversaoler);
71.
72.  //>>>>>>>> Fluxos
73.  $d_fluxos = $docsregra -> of_get_dados_fluxos($con, $iddoc, $uslogado, $leitura,
$idversaoler);
74.
75.  //>>>>>>>> Requisitos
76.  $d_requisitos = $docsregra -> of_get_dados_requisitos($con, $iddoc, $uslogado,
$leitura, $idversaoler);
77.
78.  //>>>>>>>> Diagramas
79.  $d_diagramas = $docsregra -> of_get_dados_diagramas($con, $iddoc, $uslogado, $leitura,
$idversaoler);
80.
81.  //>>>>>>>> Prototipos de interface
82.  $d_prototipos = $docsregra -> of_get_dados_prototipos($con, $iddoc, $uslogado,
$leitura, $idversaoler);
83.
84.  //>>>>>>>> Referencias
85.  $d_referencias = $docsregra -> of_get_dados_referencias($con, $iddoc, $uslogado,
$leitura, $idversaoler);
86.  //[...]

```

Fonte: Autoria própria.

docsregra.php: Arquivo responsável por manter regra de busca das informações dos documentos. Identificando o usuário, modo de leitura ou edição e até a versão a ser apresentada.

Listagem 4 - Listagem de trecho de código PHP para captura de dados da capa dos documentos.

```

87.     class docsregra{
88.         //[...]
89.         function of_get_dados_capa($con, $iddoc, $idusuario, $leitura, $idversao){
90.             $ret = array('idcapa' => 'NULL', 'iddoc' => $iddoc, 'idversao' => 0, 'titulo' =>
'Sem titulo', 'idusuario' => $idusuario, 'dtalt' => date('Y-m-d H:i:s'));
91.
92.             if($leitura){
93.                 $s_idversao = '';
94.                 if($idversao > 0){
95.                     $s_idversao = ' and doccapa.idversao <= '.$idversao;
96.                 }
97.                 $sql = 'SELECT
98.                     t.idcapa, t.iddoc, t.idversao, t.titulo, t.idusuario, t.dtalt
99.                 FROM
100.                    doccapa t
101.                WHERE
102.                    t.iddoc = '.$iddoc.' and t.idversao =
103.                    (SELECT max(idversao) as idversao FROM doccapa WHERE doccapa.iddoc =
104.                    '.$iddoc.$s_idversao.'));
105.                }else{
106.                    $sql = 'SELECT
107.                        t.idcapa, t.iddoc, t.idversao, t.titulo, t.idusuario, t.dtalt
108.                    FROM
109.                        doccapa t
110.                    WHERE
111.                        t.iddoc = '.$iddoc.' and t.idversao =
112.                        (SELECT
113.                            CASE
114.                                WHEN EXISTS (SELECT 1 FROM doccapa dc WHERE dc.iddoc = '.$iddoc.' and dc.idversao =
115.                                0 and dc.idusuario = '.$idusuario.') THEN
116.                                    0
117.                                ELSE
118.                                    (SELECT max(idversao) as idversao FROM doccapa WHERE doccapa.iddoc =
119.                                    '.$iddoc.')
120.                                END as idversao)';
121.                }
122.                $exec_sql = mysql_query($sql, $con);
123.                while($res = mysql_fetch_array($exec_sql)){
124.                    $ret['idcapa'] = $res['idcapa'];
125.                    $ret['iddoc'] = $res['iddoc'];
126.                    $ret['idversao'] = $res['idversao'];
127.                    $ret['titulo'] = $res['titulo'];
128.                    $ret['idusuario'] = $res['idusuario'];
129.                    $ret['dtalt'] = $res['dtalt'];
130.                }
131.            }
132.            return $ret;
133.        }
134.    }

```

Fonte: Autoria própria.

pastaslistas.php: Arquivo responsável por recuperar todas as pastas cadastradas e apresentá-las em formato de árvore. Na mesma também é possível recuperar os documentos conforme parâmetros passados.

Listagem 5 - Listagem de trecho de código PHP recursivo para captura e montagem de apresentação das pastas e documentos.

```

134.  //[...]
135.  function of_imprimirfilhos($idpastapai,$rpc, $nomepai, $opcoes, $rpdocs){
136.      $ret = '';
137.      for($l = 1; $l <= count($rpc)+1; $l++){
138.          if($rpc[$l]['idpastapai'] == $idpastapai){
139.
140.              $ls_docs = of_apresentadocs($rpc[$l]['idpasta'], $rpdocs);
141.              $lb_temdocs = (trim($ls_docs) <> '');
142.
143.              $ret = $ret.<li '.iif(($rpc[$l]['temfilho'] == 1 or $lb_temdocs == 1),
iif($rpc[$l]['status'] == 'I', '', 'class="parent_li"),'').'style="display: none;">;
144.
145.              $ret = $ret.<span '.iif($rpc[$l]['status'] == 'I', 'style="background:
rgb(216, 216, 216);" title="Inativo", ').'><i class="'.iif(($rpc[$l]['temfilho'] == 1 or
$lb_temdocs == 1), 'icon-folder-close','icon-folder-open')."></i>
'. $rpc[$l]['nomepasta'].iif($rpc[$l]['status'] == 'I', ' (Inativo)', ').'
['.$rpc[$l]['nomeprojeto'].']'</span>;
146.              $ret = $ret.iif($opcoes == 1, ' <a
href="'. $javascript:carrega('php/cad/cadastro.php?cad=pasta&id=".$rpc[$l]['idpasta']. "&tipo=vi
sualizar");'."><span title="Editar" style="background: rgb(242, 240, 255);"><i class="icon-
pencil"></i></span></a><a
href="'. $javascript:carrega('php/cad/cadastro.php?cad=pasta&idpastapai=".$rpc[$l]['idpasta']. "
&tipo=visualizar");'."><span title="Criar nova pasta em: '.$rpc[$l]['nomepasta']. "
style="background: rgb(242, 240, 255);"><i class="icon-plus-sign"></i></span></a><a
href="'. $javascript:carrega('php/docs/docs.php?novo=s&idpasta=".$rpc[$l]['idpasta']. "',
'CADASTRAR Novo Documento');'."><span title="Criar novo documento de especificação de
software em: '.$rpc[$l]['nomepasta']. " style="background: rgb(242, 240, 255);"><i
class="icon-star"></i></span></a>',
147.              ' <a href="#"
onClick="'. $javascript:of_alterapasta(".$rpc[$l]['idpasta']. "',
'.'. $rpc[$l]['nomepasta']. "')'." data-dismiss="modal" style="cursor:pointer;"><span
title="Selecionar esta pasta" style="background: rgb(242, 240, 255);"><i class="icon-chevron-
left"></i> Selecionar</span></a>');
148.              if(($rpc[$l]['temfilho'] == 1 or $lb_temdocs == 1) and ($rpc[$l]['status'] ==
'A')){
149.                  $ret = $ret.<ul>;
150.                  //se tiver arquivos apresentar aqui
151.                  // $ret = $ret.<li style="display: none;"><span><i class="icon-file"></i><a
href="#"> teste de arquivo requisito x </a></span></li>;
152.                  if($opcoes <> 2 and $lb_temdocs == 1){
153.                      $ret = $ret.$ls_docs;
154.                  }
155.
156.                  $ret = $ret.of_imprimirfilhos($rpc[$l]['idpasta'], $rpc,
$rpc[$l]['nomepasta'], $opcoes, $rpdocs);
157.                  $ret = $ret.</ul>;
158.              }
159.              $ret = $ret.</li>;
160.          }
161.      }
162.      return $ret;
163.  }
164.  //[...]

```

Fonte: Autoria própria.

gravabanco.php: Arquivo responsável por armazenar os dados dos formulários ao banco de dados. O mesmo é capaz de montar dinamicamente códigos de inserção e atualização de dados ao banco. O reuso e a manutenção são as principais vantagens na utilização deste arquivo, normalmente sem a utilização de frameworks como este, um arquivo de gravação e atualização de dados eram gerados para cada formulário, cada um contendo um script SQL para manipulação dos dados ao banco. Desta forma apenas um arquivo exerce essas funções.

Listagem 6 - Listagem de código PHP, framework para inserção e atualização de dados com o banco de dados.

```

165.  //[...]
166.  $stemfoto = false;
167.  if(!empty($_POST["temfoto"])){
168.      if($_POST["temfoto"] == 'tem'){
169.          $stemfoto = true;
170.      }
171.  }
172.
173.  if($stemfoto){
174.      if($_POST[$_POST['pk']] <> 'NULL'){
175.          include("gravaimagem.php");
176.          $retimagens = of_gravaimagens($_FILES, $_POST[$_POST['pk']], $_POST['tabela'],
$con);
177.          if($retimagens <> 1){
178.              echo $retimagens;
179.              //Continua gravacao dos demais registros no banco
180.          }
181.      }
182.  }
183.
184.  $stemsenha = false;
185.  if(!empty($_POST["md5"])){
186.      if($_POST["md5"] <> ''){
187.          $stemsenha = true;
188.      }
189.  }
190.  if($stemsenha){
191.      $camposenha = $_POST["md5"];
192.  }
193.
194.  //recebe os dados do formulario
195.  $atualiza = $_POST['atualiza']; //T ou F
196.  $pk = $_POST['pk'];
197.  $tabela = $_POST['tabela'];
198.  $cols = $_POST['colunas']; //col1,col2,col3
199.  $colunas = explode(",", $cols);
200.  //recebe o array dos campos e valores
201.  $parm = $_POST;
202.
203.  if($stemsenha){
204.      if(strpos($cols,$camposenha) === false){
205.          $stemsenha = false;
206.      }
207.  }
208.  if($stemsenha){
209.      $colunas[count($colunas) + 1] = 'tamanhosenha';
210.      $parm['tamanhosenha'] = strlen($parm[$camposenha]);
211.  }
212.
213.  $camposconcat = '`\`idresposta`, `iduser`, `texto`, `data`, `idsuporte`
214.  $valoresconcat = '`\`//NULL, "'.$nome."', "'.$resp.'"
215.  $filtro = '`\`;
216.
217.  foreach ($colunas as $key => $val) {
218.      if($atualiza == 'F'){
219.          //insert campos
220.          if($camposconcat <> ''){
221.              $camposconcat = $camposconcat.", `".$val."`;
222.          }else{
223.              $camposconcat = "`".$val."`;
224.          }
225.
226.          if($val == $pk){
227.              $valor = $parm[$val];
228.          }else{
229.              $valor = "'.$parm[$val]."'";
230.          }
231.
232.          if($stemsenha){
233.              if($val == $camposenha){
234.                  $valor = 'md5('.$valor.')';

```

```

235.     }
236.   }
237.
238.   //insert valores
239.   if($valoresconcat <> ''){
240.     $valoresconcat = $valoresconcat.", ".$valor;
241.   }else{
242.     $valoresconcat = $valor;
243.   }
244. }else{
245. //update
246. if($val <> $pk){
247.   $valor = "".$parm[$val]."";
248.
249.   if($temsenha){
250.     if($val == $camposenha){
251.       $valor = 'md5('.$valor.')';
252.     }
253.   }
254.
255.   if($valoresconcat <> ''){
256.     $valoresconcat = $valoresconcat.", `".$val."` = ".$valor;
257.   }else{
258.     $valoresconcat = "`".$val."` = ".$valor;
259.   }
260. }
261. }
262. }
263.
264. //Monta o filtro aqui
265. $filtro = "`".$pk."` = ".$parm[$pk];
266.
267.
268. if($atualiza == 'T'){
269.   echo of_atualizadado($tabela, $valoresconcat, $filtro, $con);
270. }else{
271.   echo of_inseredado($tabela, $camposconcat, $valoresconcat, $con);
272. }
273.
274. function of_inseredado($stab, $campos, $valores, $con){
275.   $lb_gravado = -1;
276.
277.   $sql = "INSERT INTO ".$stab." (".$campos.") VALUES (".$valores.");";
278.
279.   $res = mysql_query($sql, $con);
280.
281.   if(!$res){
282.     $lb_gravado = -1;
283.   }else{
284.     $lb_gravado = 1;
285.   }
286.
287.   return $lb_gravado;
288. }
289.
290. function of_atualizadado($stab, $camposevalores, $filtro, $con){
291.   $lb_gravado = -1;
292.
293.   $sql = "UPDATE ".$stab." SET ".$camposevalores." WHERE ".$filtro.";";
294.
295.   $res = mysql_query($sql, $con);
296.
297.   if(!$res){
298.     $lb_gravado = -1;
299.   }else{
300.     $lb_gravado = 1;
301.   }
302.
303.   return $lb_gravado;
304. }
305. ?>

```

Fonte: Autoria própria.

Cabeçalho de formulário dinâmico: A seguir é apresentado um trecho de código, em PHP e HTML, básico para o envio de dados para o servidor. Ao utilizar o arquivo citado anteriormente, algumas variáveis de formulário devem ser citadas para o correto funcionamento, são elas: atualiza (valor pode ser “T” ou “F”, respectivamente identificam se o registro será atualizado ou inserido novo), tabela (identificando o nome da tabela), colunas (identifica as colunas que serão afetadas, dispostas separadamente por vírgula), pk (identifica a chave da tabela), “variável com nome da pk” (identifica o valor da chave, utilizado para filtrar dados) e temfoto (identifica se o formulário deve manter fotos).

Listagem 7 - Listagem de trecho de código PHP e HTML do cabeçalho básico de um formulário para envio de dados para o servidor.

```

306.  //[...]
307.  <form action="php/gravabanco.php" name="form<?php echo $cad; ?>" id="form<?php echo
    $cad; ?>" <?php echo iif(($temimagem), 'enctype="multipart/form-data" accept-charset="UTF-
    8"', ''); ?> method="POST">
308.      <input type="hidden" value="<?php echo iif(($id == 0) or ($cad == 'mensagem'),
    'F', 'T'); ?>" name="atualiza">
309.      <input type="hidden" value="<?php echo $tabela; ?>" name="tabela">
310.      <input type="hidden" value="<?php echo $colunas; ?>" name="colunas" id="colunas">
311.      <input type="hidden" value="<?php echo $pk; ?>" name="pk">
312.      <input type="hidden" value="<?php echo iif(($id == 0) or ($cad == 'mensagem'),
    'NULL', $id); ?>" name="<?php echo $pk; ?>">
313.      <?php
314.          if($temimagem){
315.              ?>
316.                  <input type="hidden" value="tem" name="temfoto">
317.              <?php
318.                  }
319.              ?>

```

Fonte: Autoria própria.

Anexo C – Listas de códigos desenvolvimento móvel.

A seguir são apresentados alguns códigos em listas, tidos como principais para o funcionamento do sistema móvel:

activity_principal.xml: arquivo XML responsável pela interface gráfica, neste foi adicionado o componente nativo chamado *WebView* com preenchimento cheio da tela.

A listagem a seguir representa o arquivo XML que foi gerado:

Listagem 8 - Listagem de código XML do arquivo activity_principal.xml.

```

320. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
321.     xmlns:tools="http://schemas.android.com/tools"
322.     android:layout_width="fill_parent"
323.     android:layout_height="fill_parent"
324.     tools:context="centenarojonas.geres.PrincipalActivity" >
325.
326.     <WebView
327.         android:id="@+id/wvNavegador"
328.         android:layout_width="fill_parent"
329.         android:layout_height="fill_parent" />
330.
331. </RelativeLayout>

```

Fonte: Autoria própria.

PrincipalActivity.java: Com papel de *controller*, esse arquivo é responsável pela interação do usuário com o aplicativo. Nesse arquivo são configurado o navegador e iniciada a navegação.

A listagem a seguir representa trechos de códigos Java desse arquivo que fazem todo o controle da aplicação:

A listagem a seguir representa o arquivo JAVA que foi gerado:

Listagem 9 - Listagem de código JAVA do arquivo ActivityPrincipal.java.

```

332. //[...]
333. public class PrincipalActivity extends Activity {
334.
335.     public WebView wvNavegador;
336.     public String urlGeres;
337.
338.     @Override
339.     protected void onCreate(Bundle savedInstanceState) {
340.         super.onCreate(savedInstanceState);
341.         setContentView(R.layout.activity_principal);
342.
343.         wvNavegador = (WebView)findViewById(R.id.wvNavegador);
344.         urlGeres = "http://172.30.6.45/geres2";
345.         setConfiguracoesNavegador();
346.         carregaPagina();
347.     }
348.
349.     public void setConfiguracoesNavegador(){
350.         wvNavegador.getSettings().setJavaScriptEnabled(true);
351.         wvNavegador.setWebViewClient(new InsideWebViewClient());

```

```

352.     }
353.
354.     public void carregaPagina(){
355.         wvNavegador.loadUrl(getUrlGeres());
356.     }
357.
358.     private class InsideWebViewClient extends WebViewClient {
359.     @Override
360.         public boolean shouldOverrideUrlLoading(WebView view, String url) {
361.     view.loadUrl(url);
362.         return true;
363.     }
364.     }
365.
366.     public String getUrlGeres(){
367.         SharedPreferences sp = getSharedPreferences("centenarojonas.geres_preferences",
Context.MODE_PRIVATE);
368.         if(sp.getBoolean("enablePreference", false)){
369.             urlGeres = sp.getString("endereco", "http://192.168.5.3/geres2");
370.         }else{
371.             urlGeres = "http://192.168.5.3/geres2";
372.         }
373.
374.         return urlGeres;
375.     }
376.
377.     @Override
378.     public boolean onCreateOptionsMenu(Menu menu) {
379.         getMenuInflater().inflate(R.menu.principal, menu);
380.         return true;
381.     }
382.
383.     @Override
384.     public boolean onOptionsItemSelected(MenuItem item) {
385.         int id = item.getItemId();
386.         if (id == R.id.action_settings) {
387.             Intent i = new Intent(this, ConfigActivity.class);
388.             startActivity(i);
389.             return true;
390.         }
391.         if(id == R.id.paginaInicial){
392.             wvNavegador.loadUrl(getUrlGeres());
393.         }
394.         return super.onOptionsItemSelected(item);
395.     }
396. }

```

Fonte: Autoria própria.

AndroidManifest.xml: Arquivo de configuração do aplicativo, ícones, nome da aplicação, *activities*, definições de segurança, controle de versão, configurações mínimas e permissões estão denifidas nele. Uma das principais configurações necessárias para o funcionamento dessa aplicação é a permissão de usuário para acesso a rede de internet, que é atribuída a este arquivo através do código: “<uses-permission android:name="android.permission.INTERNET" />”

A listagem a seguir representa o arquivo XML que foi gerado:

Listagem 10 - Listagem de código XML do arquivo activity_principal.xml.

```

397. <?xml version="1.0" encoding="utf-8"?>
398. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
399.     package="centenarojonas.geres"
400.     android:versionCode="1"
401.     android:versionName="1.0" >
402.

```

```

403. <uses-sdk
404.     android:minSdkVersion="7"
405.     android:targetSdkVersion="7" />
406.
407. <uses-permission android:name="android.permission.INTERNET" />
408.
409. <application
410.     android:allowBackup="true"
411.     android:icon="@drawable/ic_launcher"
412.     android:label="@string/app_name"
413.     android:theme="@style/AppTheme" >
414. <activity
415.     android:name=".PrincipalActivity"
416.     android:label="@string/app_name" >
417. <intent-filter>
418. <action android:name="android.intent.action.MAIN" />
419.
420. <category android:name="android.intent.category.LAUNCHER" />
421. </intent-filter>
422. </activity>
423. <activity
424.     android:name=".ConfigActivity"
425.     android:label="@string/title_activity_config" >
426. </activity>
427. </application>
428.
429. </manifest>

```

Fonte: Autoria própria.

Para esse projeto foram utilizadas opções para possibilitar o usuário configurar endereço de servidor, para isso foi utilizados os seguintes recursos: para a parte visual foi adicionado uma interface *PreferenceScreen*, capaz de interagir com o usuário e armazenar os dados alterados pelo mesmo. Na *ActivityPrincipal.java*, ou seja, no *controller* foi utilizado o recurso de *SharedPreferences* para capturar as configurações informadas pela interface de tipo *PreferenceScreen*.

A listagem a seguir representa o arquivo XML da pagina de *PreferenceScreen*:

Listagem 11 - Listagem de código XML do arquivo *activity_config.xml*.

```

430. <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
431. >
432.     <PreferenceCategory
433.         android:title="Configuração de navegação" >
434.         <CheckBoxPreference
435.             android:key="enablePreference"
436.             android:title="Utilizar Endereço especificado"
437.             android:summary="Desmarque para utilizar o padrão"
438.             android:defaultValue="true"/>
439.
440.         </PreferenceCategory>
441.         <PreferenceCategory android:title="Endereço">
442.         <EditTextPreference
443.             android:key="endereco"
444.             android:dependency="enablePreference"
445.             android:title="Endereço"
446.             android:summary="Informe qual endereço utilizar."
447.             android:dialogTitle="Informe qual endereço utilizar."
448.             android:defaultValue="http://192.168.5.3/geres2/" />
449.         </PreferenceCategory>
450.     </PreferenceScreen>

```

Fonte: Autoria própria.

A listagem a seguir representa o trecho de código responsável por capturar a configuração do usuário no *controller*:

Listagem 12 - Listagem de código JAVA para captura de configuração.

```
451. public String getUrlGeres(){
452.     SharedPreferences sp = getSharedPreferences("centenarojonas.geres_preferences",
Context.MODE_PRIVATE);
453.     if(sp.getBoolean("enablePreference", false)){
454.         urlGeres = sp.getString("endereco", "http://192.168.5.3/geres2");
455.     }else{
456.         urlGeres = "http://192.168.5.3/geres2";
457.     }
458.
459.     return urlGeres;
460. }
```

Fonte: Autoria própria.

Anexo D – Exemplo de documento impresso no sistema em ambiente real.

Propriedades Básicas

Status: Ativo
Pasta: Produto

Capa

Definir Ponto de Retirada

Projeto / Sistema: *Projeto 2*

Jonas Centenaro
Versão: 1

Figura 59 - Exemplo de documento impresso no sistema em ambiente real, Figura 1 de 4.

Histórico de Revisões

(1) Revisões				
Data	Versão	Descrição	Usuário	
2014-12-20 03:24:34	1	Criação deste Documento	Jonas Centenaro	Visualizando

Nome / Breve Descrição

Este caso de uso descreve a seleção (inclusão) e exclusão de pontos de retirada para um determinado produto.

Pré-Condições

Estar logado no sistema com um usuário que possua direito de acesso ao cadastro de Produtos. Estar com o direito de processo 52 – Bloqueios no Cadastro de Produtos configurado para "Inserir", "Alterar", "Excluir" e "Consultar".

Pós-Condições

Apresentação de mensagem Inclusão/ exclusão de pontos de retirada efetuada com sucesso;

Atores

 Sistema Software.

Figura 60 - Exemplo de documento impresso no sistema em ambiente real, Figura 2 de 4.

 **Usuario Pessoa.**

Fluxo de Eventos

Fluxo Básico

Definir Ponto de Retirada

1. Usuário: Acessa o Cadastro de Produtos/Aba Logística opção "Pontos de Retirada"; 2. Sistema: Exibe a seguinte mensagem de alerta: "A alteração do ponto de retirada vale para todos os produtos da grade! [OK]"; 3. Usuário: Aciona a opção "Incluir" padrão do Sistema; 4. Sistema: Disponibiliza os campos para preenchimento dos dados principais; 5. Usuário: Seleciona um Ponto de Retirada e solicita gravação; 6. Sistema: Retorna que os dados foram gravados com sucesso;

Fluxo Alternativo

Excluir Ponto de Retirada

1. Usuário: Aciona a opção "Excluir" padrão do Sistema; 2. Sistema: Carrega o(s) ponto(s) de retirada(s) informado(s); 3. Usuário: Solicita a exclusão; 4. Sistema: Solicita confirmação para excluir o registro; 5. Usuário: Confirma a exclusão; 6. Sistema: Informa que o registro foi excluído com sucesso e retorna ao passo 4 do Fluxo Básico;

Requisitos

Requisitos de Interface

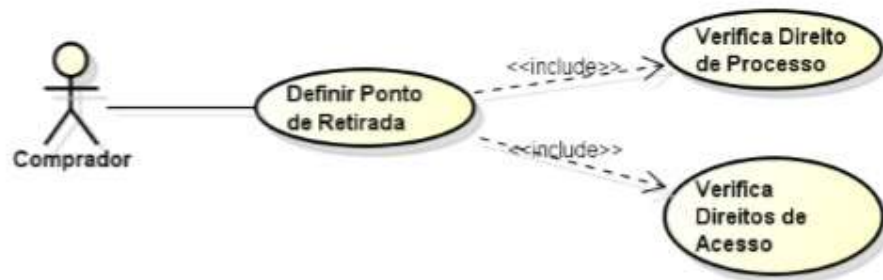
- Grupo Produto: - Código: Através deste campo o usuário poderá visualizar o código do produto atual;
- Descrição do Produto: Através deste campo o usuário poderá visualizar a descrição do produto atual;
- Ponto de Retirada: Através deste campo o usuário poderá selecionar um Ponto de Retirada (previamente cadastrado em "Cadastros/ Pontos de Retirada") para relacionar com o produto que está sendo cadastrado;

Requisitos Cancelados

Figura 61 - Exemplo de documento impresso no sistema em ambiente real, Figura 3 de 4.

Diagramas

Diagrama pre-condições



Protótipos de Interface



Sem Imagem

Referências



Sem Imagem

Figura 62 - Exemplo de documento impresso no sistema em ambiente real, Figura 4 de 4.