

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

ROBSON DA MOTTA

CATÁLOGO PARA DISPOSITIVOS MÓVEIS

MONOGRAFIA DE ESPECIALIZAÇÃO

FRANCISCO BELTRÃO

2014

ROBSON DA MOTTA

CATÁLOGO PARA DISPOSITIVOS MÓVEIS

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. MSc. Géri Natalino Dutra

FRANCISCO BELTRÃO

2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Francisco Beltrão
Diretoria de Pesquisa e Pós-Graduação
Curso de Especialização em Desenvolvimento de Sistemas para Internet
e Dispositivos Móveis

TERMO DE APROVAÇÃO

Dia 20 do mês de novembro de 2014 às: 21 horas, na sala COLIN - anfiteatro do Campus Francisco Beltrão, realizou-se a apresentação pública da monografia pelo estudante Robson Da Motta, intitulada “Catálogo para Dispositivos Móveis.” Finalizada a apresentação e arguição, a Banca Examinadora declarou **aprovada** a monografia do estudante, como requisito parcial para obtenção do título de Especialização em Desenvolvimento e Sistemas para Internet e Dispositivo Móveis.

Professor MSc. Géri Natalino Dutra - UTFPR
(Orientador)

Professor Edson dos Santos Cordeiro - UTFPR
(Convidado)

Professor Andrei Carniel - 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.

Dedico este trabalho aos interessados e motivados ao conhecimento relacionado às tecnologias móveis, que buscam conhecimento através da pesquisa e tempo de estudo.

AGRADECIMENTOS

Os parágrafos que se seguem não atendem a todos as pessoas que de alguma forma estiveram presentes nos melhores momentos durante toda esta trajetória. Para tanto, agradeço aos meus amigos os quais guardo um sentimento especial por cada um. Desde já peço desculpas àquelas pessoas que não estiverem relacionadas, mas que por algum motivo também contribuíram para esta realização.

Agradeço aos meus amigos das aulas de Kung-fu, pelos momentos alegres.

Aos meus colegas de sala.

Aos meus professores, pelo conhecimento e dedicação.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

RESUMO

O presente trabalho aborda a temática do desenvolvimento de um sistema que proporcione uma alternativa a catálogos de produtos, convencionalmente estes catálogos são impressos em grandes quantidades o que gera volumes e custos para as empresas que trabalham com estes tipos de público. O desenvolvimento do sistema propõe a integração de ambiente Internet com móvel o qual é realizado por meio de *Web Service*. A elaboração de ambos os sistemas é por meio da linguagem de programação Java, esta, é utilizada para o desenvolvimento do sistema web e do aplicativo móvel disponível na plataforma Android. O *Web Service* por sua vez realiza a integração através das tecnologias REST e JSON para prover o transporte das informações entre os diferentes ambientes, proporcionando velocidade e simplicidade durante os processos de requisição do aplicativo móvel.

Palavras-chave: Catálogo. Android. REST. Java. JSON.

ABSTRACT

This paper talks about the issue of developing a system that provides an alternative to product catalogs. Conventionally these catalogs are printed in large quantity, which generates volumes and costs for companies that work with these products. The development of the system proposes the integration of the environment of the internet with the mobile, which is conducted through the Webservice. The preparation of both systems is by means of the Java programming language, it is used to development of the web system, and of the mobile application available in Android platform. The Web Service, in turn, performs the integration through the REST and JSON technologies to provide the transport of information between the different environments, providing simplicity and speed during the request procedure of the mobile application.

Keywords: Catalog. Android. REST. Java. JSON.

LISTA DE ILUSTRAÇÕES

FIGURA 1 -	Fluxo do processo de pedido por catálogos.....	14
FIGURA 2 -	Tela inicial do sistema ND mobile.....	20
FIGURA 3 -	Tela de listagem do sistema ND mobile.....	21
FIGURA 4 -	Tela de detalhes do item no sistema ND mobile.....	21
FIGURA 5 -	Função de pesquisa no sistema ND mobile.....	22
FIGURA 6 -	Cadastro de consultores no sistema ND mobile.....	22
FIGURA 7 -	Cadastro de produtos no sistema ND mobile.....	23
FIGURA 8 -	Tela inicial do sistema Magazine Luiza.....	24
FIGURA 9 -	Opções do menu no sistema Magazine Luiza.....	24
FIGURA 10 -	Tela de listagem no sistema Magazine Luiza.....	25
FIGURA 11 -	Tela de detalhes no sistema Magazine Luiza.....	26
FIGURA 12 -	Relacionamento entre os padrões UDDI, SOAP e WSDL.....	29
FIGURA 13 -	Tela inicial do sistema <i>Astah Community</i>	33
FIGURA 14 -	Tela inicial do sistema NetBeans 8.0.....	34
FIGURA 15 -	Tela inicial do sistema MySQL Workbench 6.0 CE.....	35
FIGURA 16 -	Tela inicial do sistema Eclipse com o <i>plugin</i> ADT.....	36
FIGURA 17 -	Exemplo de uma AVD.....	37
FIGURA 18 -	Caso de Uso, Aplicação web.....	40
FIGURA 19 -	Caso de Uso, Aplicação móvel.....	41
FIGURA 20 -	Caso de Uso, Aplicação móvel.....	42
FIGURA 21 -	Diagrama de sequência, realizar pedido.....	43
FIGURA 22 -	Diagrama de Atividade, Aplicação móvel.....	44
FIGURA 23 -	Tela de <i>login</i> do sistema <i>web</i>	45
FIGURA 24 -	Tela inicial do sistema <i>web</i>	46
FIGURA 25 -	Tela de cadastro de entidade do sistema <i>web</i>	47
FIGURA 26 -	Tela de apresentação de informações (entidades cadastradas)....	47
FIGURA 27 -	Controle de paginação do Grid, disponível nas telas de listagem.....	48
FIGURA 28 -	Tela de cadastro de produtos.....	48
FIGURA 29 -	Cadastro de figuras para os produtos pré-cadastrados.....	49
FIGURA 30 -	Exemplo de resposta do método <i>getImagens</i> no recurso do <i>Web Service</i> REST.....	52
FIGURA 31 -	Tela de configurações do aplicativo móvel.....	53
FIGURA 32 -	Tela principal do aplicativo mobile.....	54
FIGURA 33 -	Listagem das categorias no aplicativo móvel.....	56
FIGURA 34 -	Listagem dos produtos da categoria selecionada.....	57
FIGURA 35 -	Tela de detalhes do item consultado.....	58
FIGURA 36 -	Tela de consulta de valores de frete.....	58
FIGURA 37 -	Informações retornadas na consulta de valores de frete.....	59
FIGURA 38 -	Tela para realizar os pedidos.....	62

LISTA DE QUADROS

QUADRO 1 -	Descrição dos verbos usados em <i>Web Service REST</i>	30
QUADRO 2 -	Exemplo de método no recurso do <i>Web Service REST</i>	31
QUADRO 3 -	Requisitos funcionais da aplicação móvel.....	38
QUADRO 4 -	Requisitos não funcionais da aplicação móvel.....	38
QUADRO 5 -	Requisitos funcionais da aplicação <i>web</i>	39
QUADRO 6 -	Requisitos não funcionais da aplicação <i>web</i>	39
QUADRO 7 -	Exemplo de método que retorna uma instância do objeto no formato JSON.....	50
QUADRO 8 -	Exemplo de método no recurso que retorna as imagens dos produtos cadastrados.....	51
QUADRO 9 -	Trecho do código que recupera as imagens no aplicativo <i>mobile</i>	55
QUADRO 10 -	Montagem dos parâmetros de consulta do frete.....	60
QUADRO 11 -	Retorno da requisição das imagens pertencentes aos itens.....	61

LISTA DE SIGLAS

ADT	Android Developer Tools
API	Application Programming Interface
AVD	Android Virtual Device
CEP	Código de Endereçamento Postal
CNPJ	Cadastro Nacional da Pessoa Jurídica
CSS	Cascading Style Sheets
FIFA	Fédération Internationale de Football Association
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IHC	Interação Homem-Computador
JMS	Java Messaging Service
JSF	Java Server Faces
JSON	JavaScript Object Notation
PAC	Prático, Acessível e Confiável
RDW	Responsive Web Designer
REST	Representational State Transfer
SGBD	Sistema de Gerenciamento de Banco de Dados
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	Extensible Markup Language
WSDL	Web Services Description Language

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 PROBLEMA DE PESQUISA.....	13
1.2 HIPÓTESE.....	15
1.3 JUSTIFICATIVA.....	15
1.4 OBJETIVOS.....	16
1.4.1 Objetivo Geral.....	16
1.4.2 Objetivos Específicos.....	16
1.5 METODOLOGIA.....	17
1.6 ESTRUTURA DO TRABALHO.....	18
2 FUNDAMENTAÇÃO TEÓRICA.....	19
2.1 CATÁLOGO.....	19
2.2 APLICATIVOS PARA CATÁLOGO.....	19
2.3 TENDÊNCIAS DE SERVIÇOS NA WEB E EM DISPOSITIVOS MÓVEIS.....	26
2.4 PROGRAMAÇÃO PARA AMBIENTE INTERNET.....	27
2.5 UTILIZAÇÃO DE FRAMEWORK PARA DESENVOLVIMENTO WEB.....	28
2.6 WEB SERVICES REST.....	28
2.6.1 Web Service SOAP.....	29
2.6.2 Web Service REST.....	30
2.7 PROGRAMAÇÃO PARA ANDROID.....	31
3 MATERIAIS E MÉTODOS.....	32
3.1 MATERIAIS.....	32
3.1.1 Astah Community.....	32
3.1.2 NetBeans 8.0	33
3.1.3 MySQL Workbench 6.0. CE	35
3.1.4 Android Developer Tools	36
3.2 MÉTODOS.....	37
3.2.1 Levantamento de Requisitos.....	38
3.2.2 Diagramas do Sistema.....	40
4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS.....	45
4.1 APRESENTAÇÃO DO SISTEMA WEB.....	45
4.2 APRESENTAÇÃO DO SISTEMA MÓVEL.....	52
5 CONSIDERAÇÕES FINAIS.....	63
REFERÊNCIAS.....	64

1 INTRODUÇÃO

A abordagem do trabalho desenvolvido foca o desenvolvimento de um catálogo para dispositivos móveis que pode ser utilizado para quaisquer tipos de produtos. O mesmo se propõe em disponibilizar algumas utilidades como cálculo de frete e lista de pedidos. A escolha do assunto para a elaboração, surgiu após a necessidade pessoal de desenvolver uma aplicação que pudesse se adaptar para as diversas áreas relacionadas a comércio de produtos que atualmente em sua grande maioria é realizada por catálogos impressos em material físico.

Em relação ao público que o sistema proposto no trabalho de especialização é destinado, apesar de não existir uma distribuição proporcional no Brasil e os acessos serem menores em determinadas regiões como norte e nordeste, o número de pessoas que estão acessando a Internet principalmente por *smartphones* ou *tablet's* tem aumentado progressivamente (O GLOBO, 2014).

Pesquisas realizadas pelo Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (Cetic) em 9.514 (nove mil quinhentos e quatorze) Telecentros cadastrados pelo ministério das Comunicações apontam que 75% dos usuários conectados a Internet tem entre 10 a 15 anos de idade, os usuários de 16 até 24 anos somam 77% e 66% representam usuários com 25 a 34 anos, ou seja, mais da metade das pessoas entrevistadas demonstram ter acesso à Internet, e destas 31% iniciam através de celulares (CETIC ,2014).

A prática de recursos como *Responsive Web Designer* (RWD) cujas características são tornar os elementos visuais adaptados ao tamanho da tela do dispositivo em que é acessado, viabiliza o acesso aos mais variados conteúdos em diversos tipos de dispositivos móveis. Este tipo de recurso pode ser obtido usando a programação de aplicações em linguagem específica do sistema operacional como o Android ou através de arquivos com *Cascading Style Sheets* (CSS) usado para aplicar estilos visuais em páginas da Internet.

Disponibilizar elementos visuais e integrações com outras plataformas atrai a atenção dos usuários dos sistemas que são desenvolvidos para aplicativos móveis. Entretanto possibilitar alternativas que ofereçam recursos cotidianos como o sistema proposto pelo trabalho visa complementar alternativas para a melhoria em processos existentes o qual agrega vantagens para os dois lados tanto o usuário quanto o

responsável pelo catálogo, pois a ferramenta entre outros benefícios pode se destacar a agilidade em realizar o processo que antes era feito de forma manual através de caneta e papel.

1.1 PROBLEMA DE PESQUISA

Um catálogo representa uma lista ordenada de itens com suas respectivas descrições, comumente é usado para listar produtos cuja venda é através de consultores credenciados a empresa que fornece os produtos.

Os mais conhecidos entre os catálogos são os para produtos de cosmética e perfumaria entre estes pode ser relacionada algumas das grandes empresas presentes no mercado brasileiro como Natura, AVON e Jequití, ambas apresentam seus produtos através de catálogos impressos em folhas de papel.

O principal problema a ser analisado é: Qual é a importância de um sistema móvel com características de catálogo de produtos em relação aos meios convencionais? Neste sentido pode ser descrito como funciona o fluxo de um processo que é realizado durante a visita de um consultor até a entrega do pedido disponível no catálogo.

Inicialmente o processo ocorre quando um cliente solicita a visita de um consultor para que o mesmo apresente seus produtos que estão disponíveis em um catálogo convencionalmente impresso em folhas de papel, como por exemplo, os produtos disponíveis no catálogo dos consultores da empresa de cosmética AVON.

Ao escolher os produtos do catálogo o cliente informa ao consultor quais produtos está interessado em adquirir, na sequência o consultor realiza o pedido junto ao seu fornecedor, este processo entre a escolha do cliente e o pedido do consultor pode levar mais de um dia para ser realizado.

Quando o pedido do fornecedor é registrado, o mesmo aguarda a entrega do produto no endereço disponibilizado, o que costuma ser a residência do consultor. Após receber a mercadoria o consultor informa ao cliente sobre a encomenda recebida, que é liberada mediante pagamento do cliente.

O processo descrito pode ser visualizado com mais clareza através do fluxo apresentado no diagrama de sequência presente na figura 1.

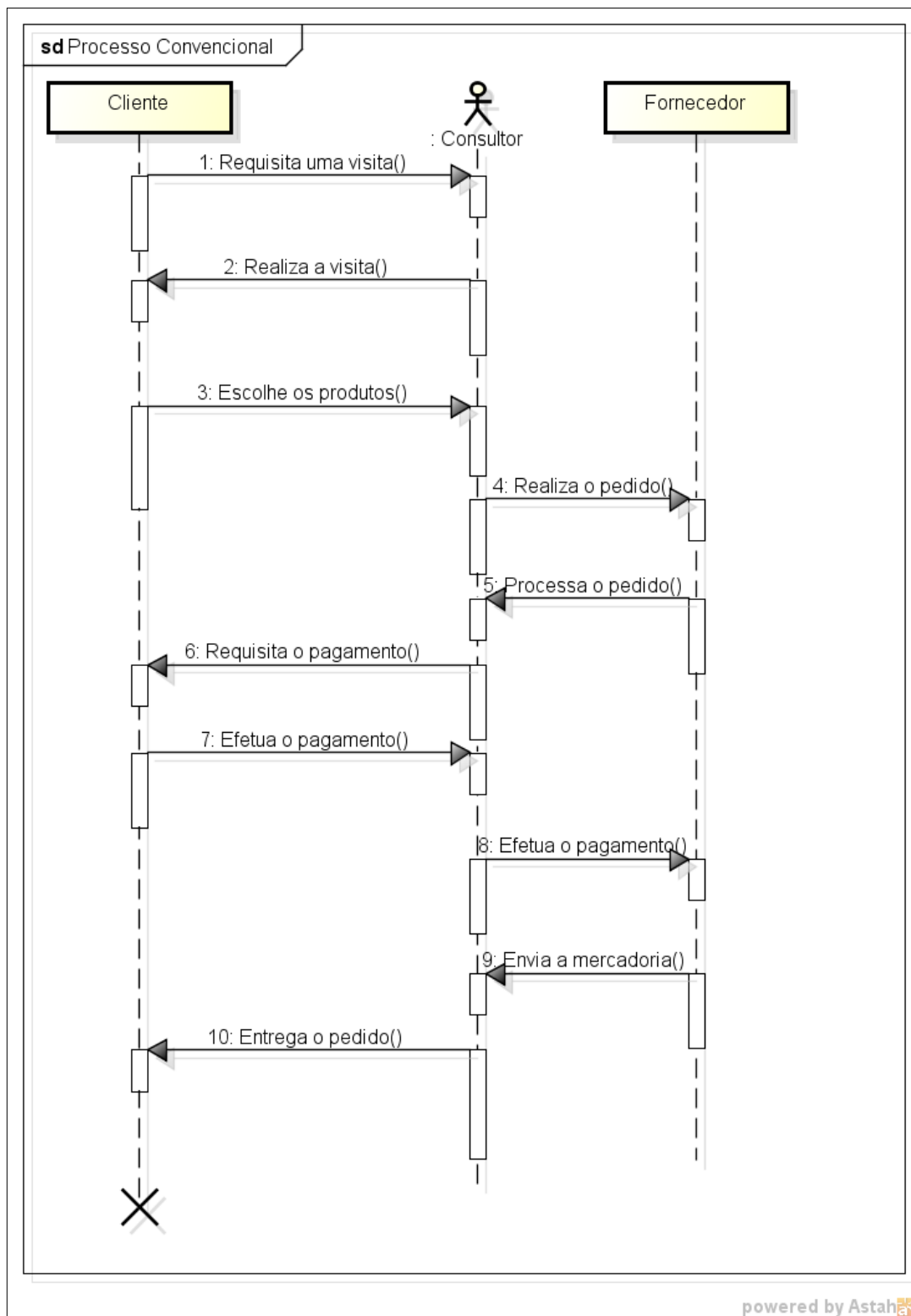


Figura 1: Fluxo do processo de pedido por catálogos.
Fonte: Autoria própria.

Como pode ser observado na figura 1, são vários os passos realizados até o término do processo, uma vez que é necessário o intermédio do consultor para que o cliente receba suas mercadorias.

1.2 HIPÓTESE

Apresentar uma alternativa digital para dispositivos móveis, dispondo de uma aplicação específica que possibilite a expansão das fronteiras alcançadas por meios físicos. Dentre as vantagens vale destacar o aumento dos usuários, popularização do catálogo e a questão ambiental, pois espera-se diminuir a utilização de papel e melhorar ainda mais a velocidade e qualidade da informação, tornando viável o sistema proposto.

Através deste projeto será respondidas algumas destas questões levantadas como a alternativa para os meios convencionais existentes e a velocidade no fluxo do processo realizado, podendo assim saber quais as reais aplicabilidades do mesmo.

1.3 JUSTIFICATIVA

No segundo trimestre de 2013 dos 8,3 milhões de *smartphones* comercializados, 90% rodavam sistema operacional Android (o restante do mercado fica principalmente com iOS da Apple, Windows Phone e BlackBerry OS). (UOL TEC, 2014).

Partindo do problema de pesquisa, a justificativa para o desenvolvimento do projeto de especialização visa disponibilizar um catálogo para dispositivos móveis, no qual o usuário possa consultar seus produtos sem a necessidade de esperar a visita de um vendedor em sua residência.

Em relação aos aplicativos similares disponíveis para uso, será citado dois produtos são eles: ND Mobile e o aplicativo da loja Magazine Luiza.

Sobre o aplicativo ND Mobile pode ser descrito que possui as funções de um catálogo, onde o mesmo apresenta como particularidade o visual dinâmico, que é personalizado de acordo com o usuário do sistema, estas opções de personalização são realizadas através do painel de controle acessando o site do sistema.

Sobre o aplicativo da loja Magazine Luiza, suas funcionalidades são específicas para a venda dos produtos da loja virtual, possui as características de

um catálogo, porém com algumas particularidades, o usuário pode localizar as lojas mais próximas de onde ele se encontra no momento em que está acessando o aplicativo. Outros detalhes sobre os aplicativos será apresentado no capítulo 2.

O diferencial da aplicação proposta no trabalho é realizar a função de um catálogo de forma simples, com auxílio a consulta de valores e custo de frete, além trabalhar com múltiplos catálogos no mesmo aplicativo, ficando a critério do usuário escolher qual deseja acessar.

Sobre tudo o aplicativo dispensa a utilização de um consultor para realizar o processo de pedidos, pois a popularização dos *smartphones* aumenta o alcance de usuários que podem acessar o aplicativo e utilizar as funcionalidades trazendo benefícios para quem deseja adquirir produtos, assim como para quem os vende.

1.4 OBJETIVOS

Nesta seção serão apresentados o objetivo geral e os objetivos específicos.

1.4.1 Objetivo Geral

Para atingir o objetivo geral são necessários os seguintes: Desenvolver uma aplicação com as características de um catálogo de produtos variados, que integre dois ambientes distintos (móvel e *web*) utilizando tecnologia REST (*Representational State Transfer*) e sistema operacional Android.

1.4.2 Objetivos Específicos

Para atingir os objetivos específicos do projeto, deve ser realizada uma sequência de passos, as quais contemplam o levantamento de requisitos a fim de descrever as funcionalidades desejadas para os produtos (sistema *web* e móvel), a elaboração dos diagramas de UML (*Unified Modeling Language*), para exemplificar

de forma visual as rotinas dos sistemas e por fim a construção dos sistemas usando as tecnologias necessárias para o desenvolvimento.

Entre as características necessárias para as aplicações são destacados as seguintes funcionalidades: Cadastro de informações pelo sistema *web*, listagem dos itens cadastrados tanto no sistema *web* quanto no sistema móvel, a troca de informações entre os dois sistemas, bem como o controle de acesso ao sistema *web* por meio de *login*.

1.5 METODOLOGIA

Para cada tipo pesquisa realizada, diferentes métodos de pesquisa são aplicados os quais são divididos nos grupos de Natureza, Abordagem, Objetivo e Procedimentos técnicos.

O presente trabalho abrange os grupos de Natureza e Procedimentos técnicos.

A pesquisa qualitativa no ponto de vista da Natureza não se preocupa com representatividade numérica. Os dados analisados não são numéricos por isso não podem ser quantificados uma vez que sua finalidade é explicar o porquê das coisas e como são feitas. (NEVES, 1996).

No ponto de vista dos procedimentos técnicos segundo Gil (1991), aborda a pesquisa bibliográfica e estudo de caso, por se tratar de pesquisas bibliográficas em artigos e materiais publicados na Internet.

O presente trabalho caracteriza-se como um estudo de caso, pois, é aplicado um estudo profundo em um ou mais objetos a fim de dispor um amplo e detalhado conhecimento sobre o mesmo, uma vez que apresenta um produto que pode ser aplicado em situações específicas do estudo.

Dispondo das informações apresentadas, o trabalho adota uma abordagem qualitativa, aplicando as técnicas de estudo de caso e pesquisas bibliográficas.

1.6 ESTRUTURA DO TABALHO

O trabalho está dividido em cinco capítulos, são eles: Introdução; Fundamentação teórica; Materiais e métodos; Apresentação e discussão dos resultados e por fim as Considerações finais.

Na Introdução é presenta de forma introdutória o assunto relacionado a catálogos que é o tema abordado na monografia.

No capítulo que se refere a Fundamentação teórica, o objetivo é familiarizar o leitor ao assunto relacionado ao trabalho, abordando uma breve descrição das tecnologias utilizadas na elaboração dos aplicativos propostos na monografia.

No capítulo que aborda os materiais e métodos utilizados no desenvolvimento do projeto, são relacionados os requisitos funcionais e não funcionais do sistema, modelagens e diagramas dos processos bem com os softwares utilizados para a realização do mesmo.

A Apresentação dos resultados expõe detalhes dos resultados obtidos no processo de desenvolvimento das aplicações *web* e móvel exemplificando rotinas de forma técnica através de trechos de código.

Nas Considerações finais, estão descritos quais resultados foram obtidos ao término do desenvolvimento do trabalho, além de apresentar o ponto de vista do autor em relação à conclusão dos resultados.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será abordada uma breve descrição dos temas relacionados ao trabalho proposto. Inicialmente será descrito sobre o tema catálogos, na sequência a apresentação de aplicativos similares ao proposto no trabalho, bem como alguns dos temas relacionados ao desenvolvimento do projeto.

2.1 CATÁLOGOS

Segundo o dicionário Aurélio, catálogo é uma lista ordenada de itens. Apresenta informações e descreve sobre cada item presente na lista (FERREIRA, 2011, p. 192).

Sua utilização em dispositivos móveis tem bastante aceitação, uma vez que as características que o dispositivo móvel apresenta como tamanho e mobilidade contribuem para a adesão deste tipo de aparelho para uso.

Aplicativos com tais características já são comuns no mercado, a funcionalidade de um catálogo em um dispositivo móvel proporciona a interação entre o usuário e a aplicação, diferente do que ocorre nos métodos convencionais (tópico 1.1 do trabalho).

Sobre as utilizações dos aplicativos móveis para uso de catálogos, será apresentado alguns aplicativos existentes, que estão disponíveis de forma gratuita para o sistema operacional Android.

2.2 APLICATIVOS PARA CATÁLOGOS

Para o desenvolvimento do aplicativo proposto no trabalho, seus requisitos levam em consideração algumas das funcionalidades básicas, são elas: proporcionar uma lista ordenada com itens e suas respectivas informações bem como a possibilidade de realizar pedidos dos mesmos. Aplicativos para esse fim já são encontrados, alguns são disponibilizados de forma gratuita e outros não.

Nesta sessão será relacionado dois aplicativos que apresentas características semelhantes as que o sistema desenvolvido oferece e ambos são para o mesmo

sistema operacional. No capítulo das referências estão dispostos os *link's* para obter os aplicativos.

Inicialmente será apresentado o aplicativo ND mobile, desenvolvido para os sistemas operacionais Android e iOS. A aplicação disponibilizada para o sistema Android dispõe de recursos como busca de registros, listagem de itens por categorias e proporciona a visualização das imagens cadastradas de forma ampliada.

Na figura 2 apresenta tela inicial do aplicativo no dispositivo.



Figura 2: Tela inicial do sistema ND mobile.
Fonte: (NDM, 2014).

A listagem dos itens no catálogo é disposta de acordo com o tamanho da tela que o dispositivo móvel apresenta o que pode variar entre um item ou até oito itens na mesma tela. A figura 2 apresenta os itens em um dispositivo cuja dimensão da tela é sete polegadas.



Figura 3: Tela de listagem do sistema ND mobile.
Fonte: (NDM, 2014).

A visualização dos detalhes de cada item é apresentada na figura 4 a seguir.



Figura 4: Tela de detalhes do item no sistema ND mobile.
Fonte: (NDM, 2014).

Como é possível observar na figura 4, as informações do item são disponibilizadas em uma única caixa de texto para o usuário do aplicativo. O recurso visual empregado é bastante interessante no ponto de vista estético, pois aproveita ao máximo a proporção da tela do dispositivo, mantendo a imagem do item a vista do usuário.

A consulta dos itens é realizada através de uma caixa de texto onde a informação digitada realiza o filtro no aplicativo, a fim de listar apenas as informações consultadas.

Na figura 5 a seguir é possível ver com detalhes a função de pesquisas.



Figura 5: Função de pesquisa no sistema ND mobile.
Fonte: (NDM, 2014).

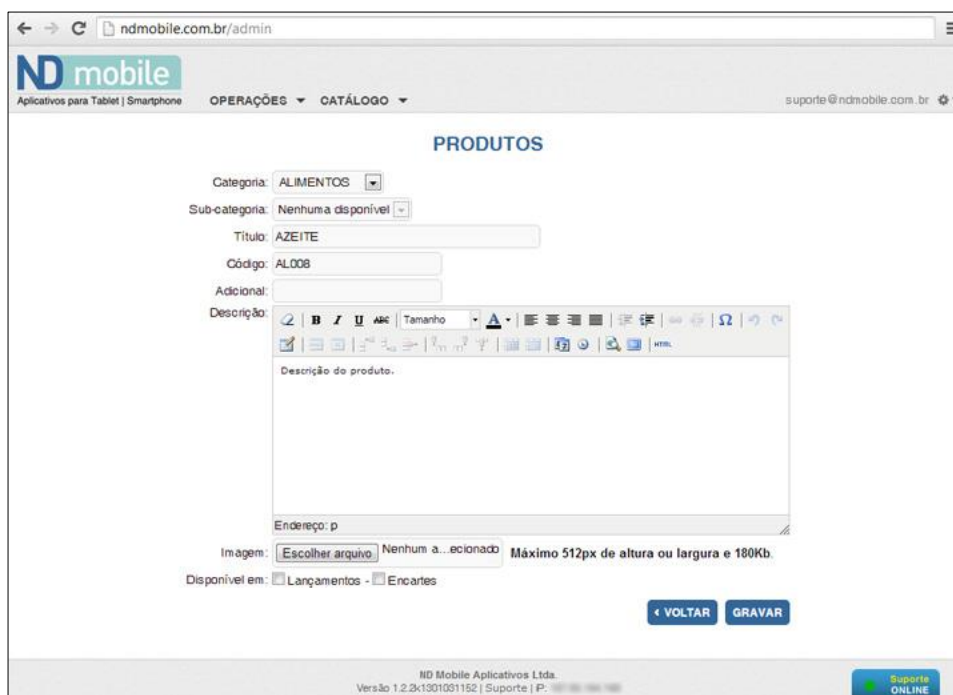
Para cadastrar as informações do catálogo o consultor possui acesso ao painel de controles do sistema, este painel está disponível no site do desenvolvedor do aplicativo uma vez que o consultor esteja devidamente cadastrado no sistema.

A figura 6 apresenta a tela de cadastro de consultores no painel de controles do sistema *web* ND mobile.

Figura 6: Cadastro de consultores no sistema ND mobile.
Fonte: (NDM, 2014).

O acesso as configurações do painel de controle, são controladas através de *login*, e as informações dispostas são exibidas no aplicativo móvel. Para o gerenciamento dos produtos estão disponíveis as opções de edição, cadastro e exclusão das informações.

A figura 7 apresenta as informações necessárias para inclusão do produto no catálogo.



A imagem mostra a interface de administração do sistema ND mobile, especificamente a tela de cadastro de produtos. O navegador indica o endereço ndmobile.com.br/admin. O cabeçalho contém o logo 'ND mobile', o texto 'Aplicativos para Tablet | Smartphone', menus para 'OPERAÇÕES' e 'CATÁLOGO', e o e-mail de suporte 'suporte@ndmobile.com.br'. O título principal da página é 'PRODUTOS'. O formulário de cadastro contém os seguintes campos: 'Categoria:' com o menu suspenso 'ALIMENTOS'; 'Sub-categoria:' com o menu suspenso 'Nenhuma disponível'; 'Título:' com o texto 'AZEITE'; 'Código:' com o texto 'AL008'; 'Adicional:' com um campo de texto vazio; 'Descrição:' com uma área de texto rica em formato (WYSIWYG) contendo o texto 'Descrição do produto.'; 'Endereço:' com o texto 'p'; 'Imagem:' com o botão 'Escolher arquivo', o texto 'Nenhum a...eclonado' e a especificação 'Máximo 512px de altura ou largura e 180Kb'; e 'Disponível em:' com as opções 'Lançamentos' e 'Encartes'. Na base do formulário, há dois botões: 'VOLTAR' e 'GRAVAR'. Na barra de rodapé, há o texto 'ND Mobile Aplicativos Ltda. Versão 1.2.2k1301031152 | Suporte | P...' e um botão 'Suporte ONLINE'.

Figura 7: Cadastro de produtos no sistema ND mobile.
Fonte: (NDM, 2014).

As funções presentes no sistema ND mobile são semelhantes as funcionalidades do aplicativo proposto no trabalho de especialização, porém seu diferencial em relação ao aplicativo ND mobile é que o mesmo dispõe de recursos de cálculo de frete, a possibilidade de trabalhar com múltiplos catálogos no mesmo aplicativo e possibilidade de realizar pedidos usando o aplicativo.

A seguir será relacionado as funcionalidades do segundo aplicativo analisado, o aplicativo da loja Magazine Luiza.

A figura 8 apresenta a tela inicial da aplicação Magazine Luiza.



Figura 8: Tela inicial do sistema Magazine Luiza.
Fonte: (MLAPP, 2014).

As funções do aplicativo, estão presentes no menu da aplicação localizado no canto superior esquerdo do aplicativo e possui como ícone três barras alinhadas horizontalmente como mostra na figura 9.

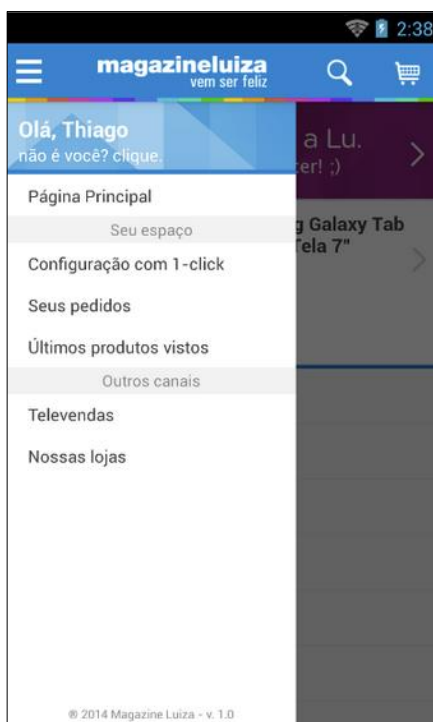


Figura 9: Opções do menu no sistema Magazine Luiza.
Fonte: (MLAPP, 2014).

As opções do menu, estão associadas a conta do usuário no *site* da loja virtual. Como é possível observar as funcionalidades são específicas para a integração com o *site*.

A visualização dos produtos disponíveis no catálogo é disposta por categorias como mostra a figura 10.



Figura 10: Tela de listagem no sistema Magazine Luiza.
Fonte: (MLAPP, 2014).

Ao selecionar um determinado item para visualizar os detalhes do produto, o usuário é redirecionado para a tela de detalhes, onde o mesmo pode contar com recursos de vídeo e imagem sobre o produto listado.

A figura 11 na sequência mostra a tela de detalhes do produto no aplicativo.

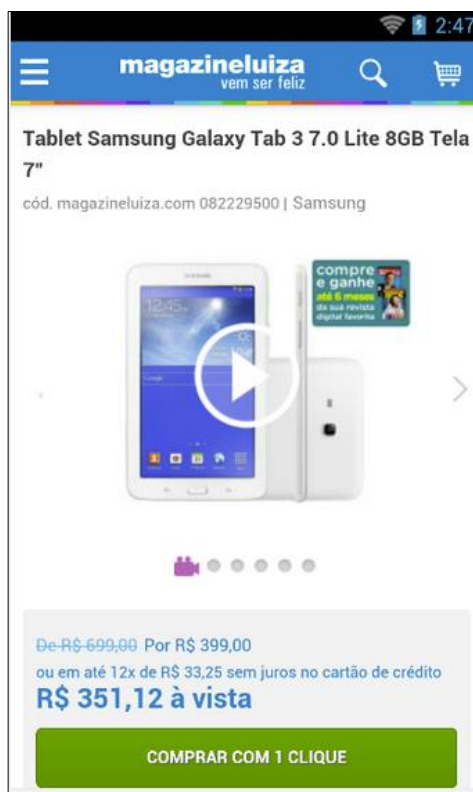


Figura 11: Tela de detalhes no sistema Magazine Luiza.
Fonte: (MLAPP, 2014).

Comparando com o aplicativo proposto, o sistema desenvolvido para a loja Magazine Luiza apresenta como diferencial a funcionalidade de proporcionar ao usuário várias formas de pagamento dos produtos como boleto bancário ou cartão de crédito.

Sobre tudo vale apenas destacar que a aplicação é específica para a loja, o que não ocorre no aplicativo proposto, o qual dispõe como uma das características a possibilidade de trabalhar com mais de um catálogo no mesmo aplicativo.

2.3 TENDÊNCIAS DE SERVIÇOS NA WEB E EM DISPOSITIVOS MÓVEIS

Cada vez mais são criadas aplicações e ferramentas para dispositivos móveis que atendam as necessidades dos usuários, uma vez que cresce o número de pessoas conectadas à Internet através de *smartphones* ou *tablet's*. Por este motivo, as empresas estão explorando cada vez mais este mercado, proporcionando soluções que antes existiam apenas para ambiente *desktop* (TENDENCIA, 2014).

Serviços de diversas naturezas podem ser encontrados para dispositivos móveis, entre os mais usados estão os aplicativos bancários para gerenciamento das contas bancária, lojas virtuais onde proporcionam compra e venda de produtos ou até mesmo *disk pizza*.

Pesquisas realizadas apontam que o Brasil é o país que mais possui pessoas conectadas a Internet, entre estes 97,8% do tempo é usando redes sociais (COMSCORE, 2014). Partindo deste princípio o ministério da fazenda criou uma aplicação para serviços de atendimento móvel de emergência (SAMU), o qual integra redes sociais e está disponível para a plataforma móvel e disponibilizou para os usuários a partir do período da Copa da FIFA (*Fédération Internationale de Football Association*) no Brasil. (COPA, 2014).

Apesar de tudo ainda é possível encontrar nichos de mercado que ainda não foram totalmente explorados. Como a tecnologia está em constante evolução, ainda é possível encontrar serviços que podem ser automatizados ou atualizados.

Na próxima sessão será apresentado de maneira sucinta o desenvolvimento em ambiente para Internet e suas características.

2.4 PROGRAMAÇÃO PARA AMBIENTE INTERNET

Quando se pensa em desenvolvimento *web*, muitos obstáculos que são encontrados no desenvolvimento de sistemas *desktop* não existem, tais como os recursos de *hardware* para manter um sistema estável sem demonstrar lentidão ou baixa capacidade de armazenamento de informações. Isto se deve ao fato que as empresas que oferecem serviços de hospedam de domínios dispõem de recursos suficientes para manter uma aplicação seja ela robusta ou não.

Aplicações no ambiente *web* possuem riqueza em relação ao visual, isto se deve muitas vezes ao uso de *frameworks* ou bibliotecas de código, responsáveis por tornar o *layout* mais atrativo e prender a atenção do usuário. Entre os *frameworks* mais utilizados destaca-se o *Primefaces*, que foi utilizado para a elaboração do projeto proposto no trabalho. O *framework* é gratuito e disponível para desenvolvedores da linguagem *Java*.

Na sessão seguinte será abordado com mais detalhes o tema *framework*, descrevendo algumas de suas características e usabilidades.

2.5 UTILIZAÇÃO DE FRAMEWORK PARA DESENVOLVIMENTO WEB

Quando se deseja elaborar um projeto que necessite de uma análise mais aprofundada, certamente uma das primeiras tarefas antes de escrever o código fonte é estudar quais *frameworks* existem para atender as necessidades do projeto. Os *frameworks* são ferramentas especializadas em determinado escopo, ou seja, possuem um domínio de uma aplicação em particular, porém os mesmos permitem a extensão de funcionalidades abstratas. (DSC, 2014)

Entre os *frameworks* existentes, muitos são de código aberto, e recebem a contribuição da comunidade adequada ao uso de *softwares OpenSource* afim de melhorar cada vez mais as ferramentas e funcionalidades do *framework*, além de incentivar o crescimento da mesma (COOPERATI, 2014).

Atualmente os *frameworks* os mais conhecidos que trabalham com a linguagem de programação *Java* destaca-se o *Hibernate*, que proporciona um fácil mapeamento entre banco de dados com modelo relacional e aplicação e o *Primefaces* o qual será abordado com mais detalhes na sequência.

Os *frameworks* não são exclusivamente *back-end*¹ em um projeto, existem os responsáveis por auxiliar no *front-end*², tais como o *Primefaces* usado no desenvolvimento do sistema web deste projeto entre outros.

Na próxima seção será abordado de maneira sucinta o tema *Web Service*.

2.6 WEB SERVICE

A necessidade de transferência de informações entre sistemas distintos é resolvida com o uso de protocolos de transporte de informações. Os *web services* são componentes que permitem a integração entre sistemas construídos em plataformas diferentes tornando a comunicação entre os mesmos possível.

Sobre os tipos de *Web Service* são destacados dois tipos de comunicação os quais será relacionado de maneira sucinta nas próximas sessões.

1 - *back-end* : A lógica do framework não é visível ao usuário, as funcionalidades são internas ao sistema.

2 - *front-end* : A lógica do framework é visível ao usuário, as funcionalidades são aplicadas no visual do sistema.

2.6.1 Web Service SOAP

O *Simple Object Access Protocol* (SOAP) é uma alternativa para integração entre aplicações distintas o qual transmite mensagens em formato XML. A estrutura montada para a comunicação é dividida em três etapas: o envelope, onde estão está definido o conteúdo da mensagem transmitida; o conjunto de regras e codificações sobre os tipos de dados; e os procedimentos para as chamadas e respostas aos métodos disponíveis no *Web Service* (Erl, 2014).

As especificações do serviço estão disponibilizadas no arquivo *Web Services Description Language* (WSDL), além destas especificações o mecanismo responsável por descrever e suplementar os serviços está nas especificações *Universal Description, Discovery and Integration* (UDDI) A figura 12 apresenta o esboço de como é realizado este processo.

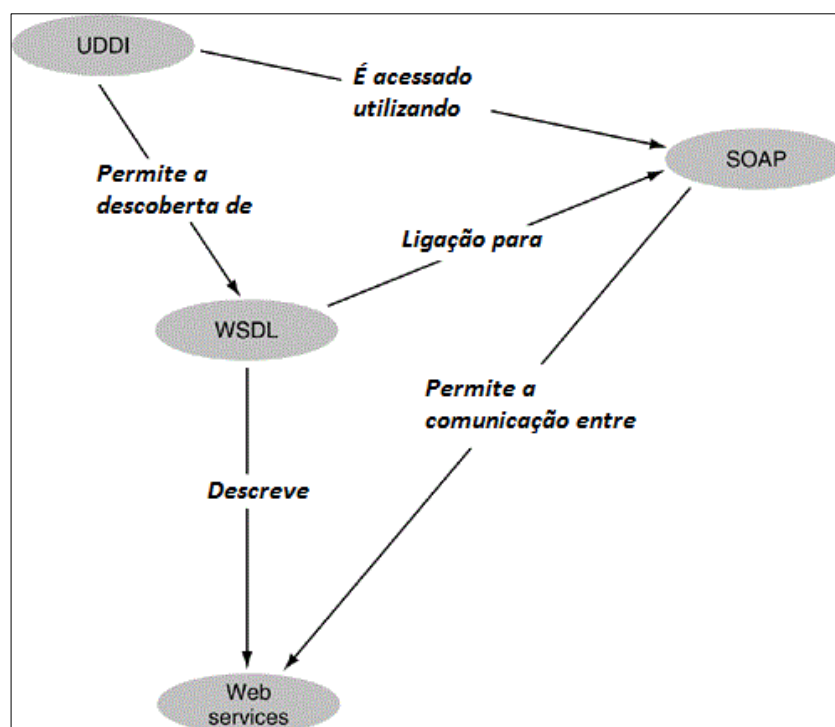


Figura 12: Relacionamento entre os padrões UDDI, SOAP e WSDL.
Fonte: Autoria própria.

A tecnologia SOAP apresenta vantagens e desvantagens para o uso, dentre as vantagens está o transporte generalizada das informações, esta não depende apenas de um tipo específico de protocolo de transporte, as mensagens podem ser

enviadas por SMTP, HTTP ou até mesmo JMS (*Java Messaging Service*) (Erl, 2014).

Como desvantagens da tecnologia destaca-se o volume de informações produzidas durante o processo de envio, por disponibilizar as informações no formato XML é necessário um grande volume de dados para transportar, o que proporciona um desempenho menor se aplicado em transporte de um grande volume de informação.

2.6.2 Web Service REST

O conceito REST surgiu a partir da tese de doutorado no ano de 2000 por Roy Fielding o qual estudava a arquitetura HTTP (*Hypertext Transfer Protocol*), suas principais características são o conceito de trabalhar com recursos, onde o mesmo possui uma URL (*Uniform Resource Locator*) única para sua identificação, sua comunicação é realizada usando o protocolo HTTP e seus recursos podem ser relacionados e possuir múltiplas formas. (Fielding, 2000).

A semântica dos recursos REST são predefinidas com as palavras (*GET, PUT, DELETE, POST e HEAD*) no cabeçalho de cada implementação do recurso.

No quadro 1 é possível observar a relação de cada verbo e suas respectivas aplicabilidades no *WebService* REST.

Ação	Descrição
GET	Usado quando deseja recuperar informações
POST	Usado para atualizar as informações no servidor
PUT	Usado para gravar as informações recebidas na URL
DELETE	Usado para apagar as informações no servidor
HEAD	Usado para recuperar os meta dados de um recurso dado a URL

Quadro 1: Descrição dos verbos usados em Web Service REST.

Fonte: A autoria própria.

Para exemplificar a sintaxe aplicação de uma funcionalidade em um recurso REST o quadro 2 apresenta um exemplo do termo na declaração de um método.

```

42
43     @GET
44     @Path("/listar")
45     @Produces("application/json")
46     public Response getEntidades() {
47         List<ICatalogo> list = DAO.getEntidadeDAO(Conexao.getConexao()).findAll();
48         JSONArray array = new JSONArray();
49         for (ICatalogo entidade : list) {
50             array.put(((Entidade) entidade).toJSON());
51         }
52         return Response.ok(array.toString().encoding("UTF-8").build());
53     }
54

```

Quadro 2: Exemplo de método no recurso do Web Service REST.
Fonte: Autoria própria.

Como é possível observar no quadro 2 o método escrito usa o termo *GET* (linha 43 do quadro 2), o qual é usado para funções que retornam informações em suas requisições.

Os recursos disponíveis no *WebService* do sistema desenvolvido produzem como resposta aos métodos dos recursos, textos na notação JSON (*JavaScript Object Notation*) este formato foi adotado por ser leve, e simples de manipular na aplicação mobile que realiza as requisições.

2.7 PROGRAMAÇÃO PARA ANDROID

A programação para dispositivos móveis é bastante limitada se comparada a um ambiente *desktop* ou *web*. Apesar desta limitação o sistema *Android* é desenvolvido e baseado na plataforma operacional *Linux*, e disponibiliza uma série de API (*Application Programming Interface*) para desenvolvedores que possuem familiaridade na linguagem de programação *Java*.

Uma das principais características do sistema *Android* é a compatibilidade, ou seja, as novas versões do sistema operacional são capazes de executar aplicações desenvolvidas nas versões anteriores da plataforma, isto faz com que desenvolvedores não se preocupem com a possibilidade de seus aplicativos pararem de funcionar quando o sistema for atualizado. Outro ponto interessante é a segurança sobre o acesso dos aplicativos, ou seja, uma aplicação só pode acessar os dados do dispositivo mediante a permissão do usuário e nunca pode acessar dados de outra aplicação. (Six, Jeff, 2012, p17).

3 MATERIAIS E MÉTODOS

No presente capítulo, serão abordados os requisitos levantados a partir da análise do processo realizado por consultores de catálogos de cosméticos e perfumaria. Também será apresentado as modelagens desenvolvidas para o sistema proposto no trabalho de especialização, bem como os softwares utilizados para o desenvolvimento dos produtos propostos.

3.1 MATERIAIS

Para o desenvolvimento dos aplicativos foram utilizados softwares específicos para auxiliar na elaboração dos diagramas de UML, e codificação dos sistemas *web* e móvel.

Inicialmente será apresentada a ferramenta utilizada para a modelagem do sistema, e seus diagramas conforme os requisitos levantados para cada situação.

3.1.1 Astah Community

A ferramenta Astah Community surgiu após a descontinuidade da sua antecessora de nome *Project Jude* para projetos UML a qual integra uma comunidade de software livre.

Dentre os destaques que a ferramenta dispõe está a iteratividade com o software além de apresentar modelos que facilitam as a construção dos diagramas, possibilitando que o analista possa desenvolver seu próprio padrão de modelagem.

A ferramenta possui como principais característica ser completamente desenvolvida em *Java* e seu uso intuitivo. A figura 13 apresenta a interface do sistema.

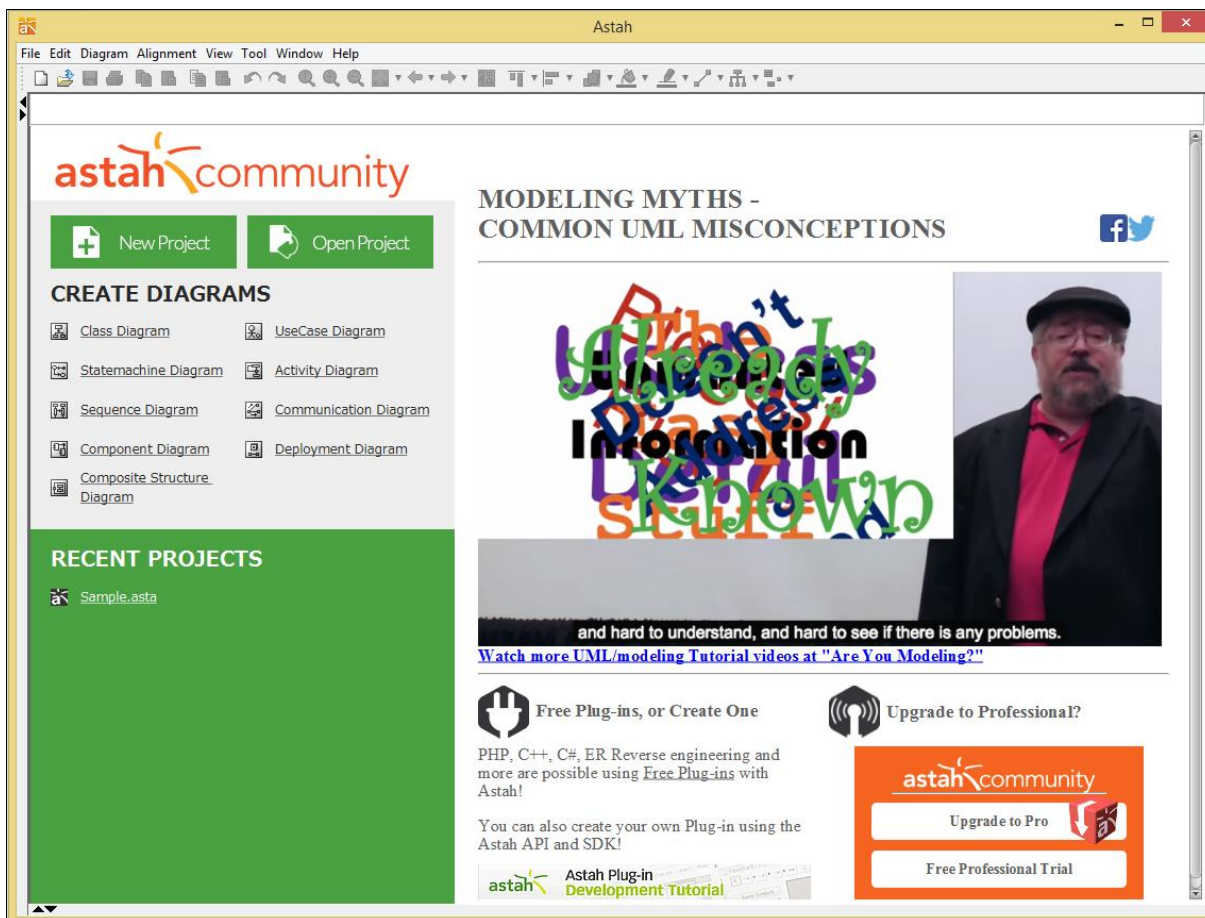


Figura 13: Tela inicial do sistema Astah.

Fonte: Autoria própria.

Uma vez que os projetos criados na ferramenta podem ser relacionados entre si, a relação entre os diagramas bem como as dependências podem tornar o projeto mais prático e de fácil entendimento, uma vez que dividindo em varias partes o modelo fica mais claro e compreensível.

Na sequência será abordado a ferramenta utilizada para o desenvolvimento do sistema *web*.

3.1.2 NetBeans 8.0

A ferramenta *NetBeans* foi escolhida para desenvolvimento da aplicação *web* por ser robusta e já dispor de uma série de funcionalidades pré-configuradas para o desenvolvimento de aplicações entre elas as aplicações *web*. Para isto a

mesma disponibiliza *containers* de servidores para depurar e simular o ambiente externo.

A linguagem de programação em que se desenvolve é *Java*, nas aplicações *web*, é empregado o uso de *frameworks* que realizam a comunicação entre a linguagem HTML e o código *Java*, entre os *frameworks* nativos destaca-se o (*Java Server Faces*) onde o mesmo dispõe de recursos ricos para aplicações de pequeno ou grande porte.

A figura 14 apresenta a interface do software.

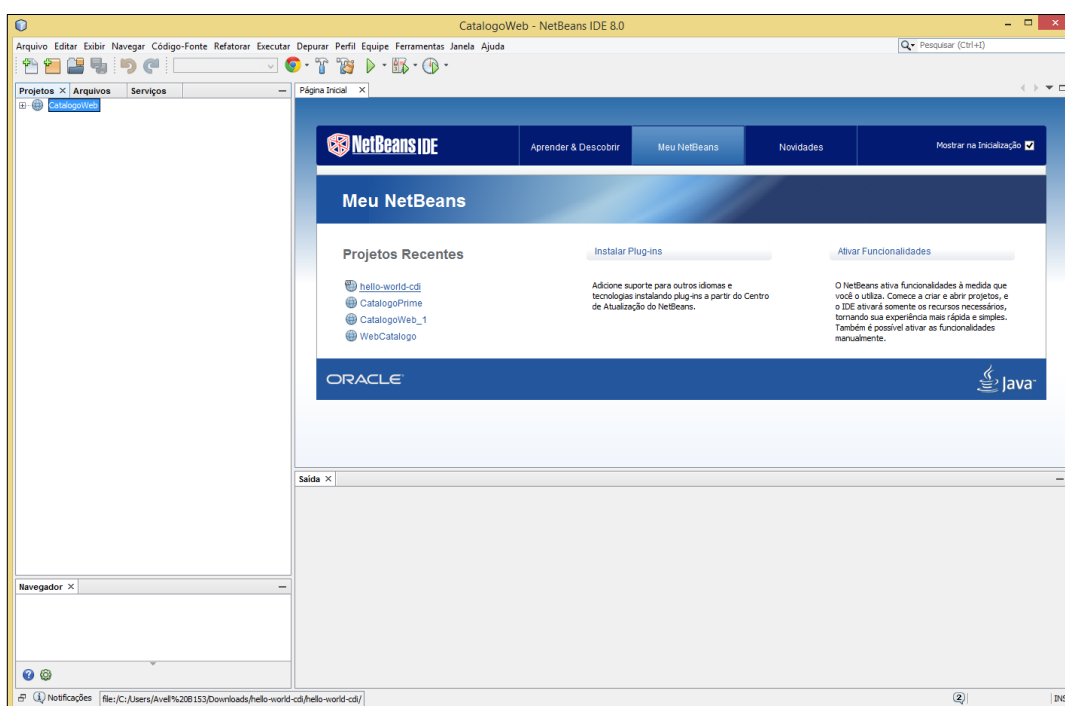


Figura 14: Tela inicial do sistema NetBeans 8.0.

Fonte: Autoria própria.

Com a interface de desenvolvimento, é possível realizar a construção de projetos com rapidez e praticidade, pois a mesma apresenta recursos que auxiliam o usuário no desenvolvimento, bem como dispõe de componentes para uso nos projetos, além de recursos avançados para construção personalizada de ferramentas personalizadas.

Na sequência será descrito sobre a ferramenta utilizada na construção e modelagem do banco de dados utilizado no sistema *web*.

3.1.3 MySQL Workbench 6.0 CE

Seu Para o armazenamento das informações do sistema *web* o bando de dados escolhido para tal finalidade foi o MySQL e para o seu gerenciamento a ferramenta SGBD (Sistema de Gerenciamento de Banco de Dados) adequada é o MySQL Workbench, pois é específica para o gerenciamento de banco de dados MySQL.

A ferramenta é gratuita para o uso, e apresenta diversos recursos para a construção, modelagem e gerenciamento de banco de dados. Dentre os recursos disponíveis está a possibilidade de gerar o modelo do banco de dados usando engenharia reversa.

Esta funcionalidade permite criar um modelo lógico a partir de um *script* em linguagem SQL (*Structured Query Language*) ou vice-versa. A figura 15 apresenta a interface inicial do sistema.

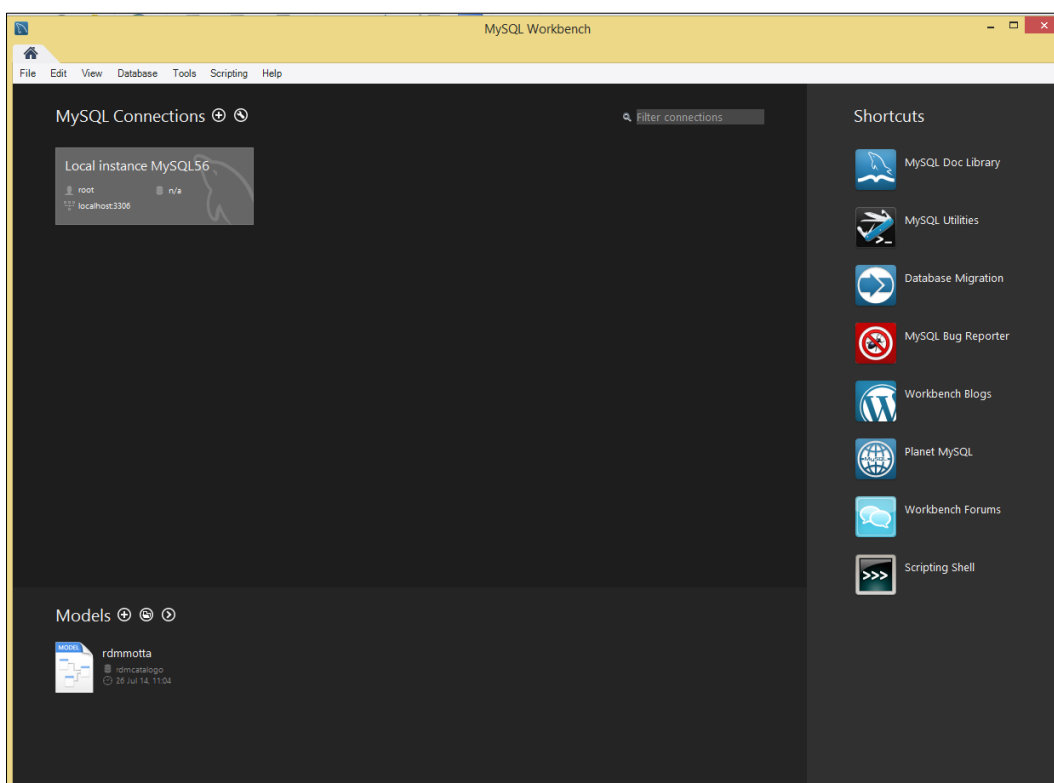


Figura 15: Tela inicial do sistema MySQL Workbench 6.0 CE.

Fonte: Autoria própria.

Na sequência será apresentado a ferramenta usada no desenvolvimento da aplicação móvel.

3.1.4 Android Developer Tools

A ferramenta *Android Developer Tools* (ADT) é um plugin com ricos recursos para o desenvolvimento de aplicações para Android, além de estar integrada a IDE Eclipse, apresenta como recurso a AVD (*Android Virtual Device*) para emular o sistema operacional Android no computador.

O uso da AVD auxilia nos testes de depuração quando o desenvolvedor não optar por testar a aplicação diretamente no dispositivo físico.

A figura 16 apresenta a *Integrated Development Environment* (IDE) Eclipse com a ferramenta ADT configurada para o uso.

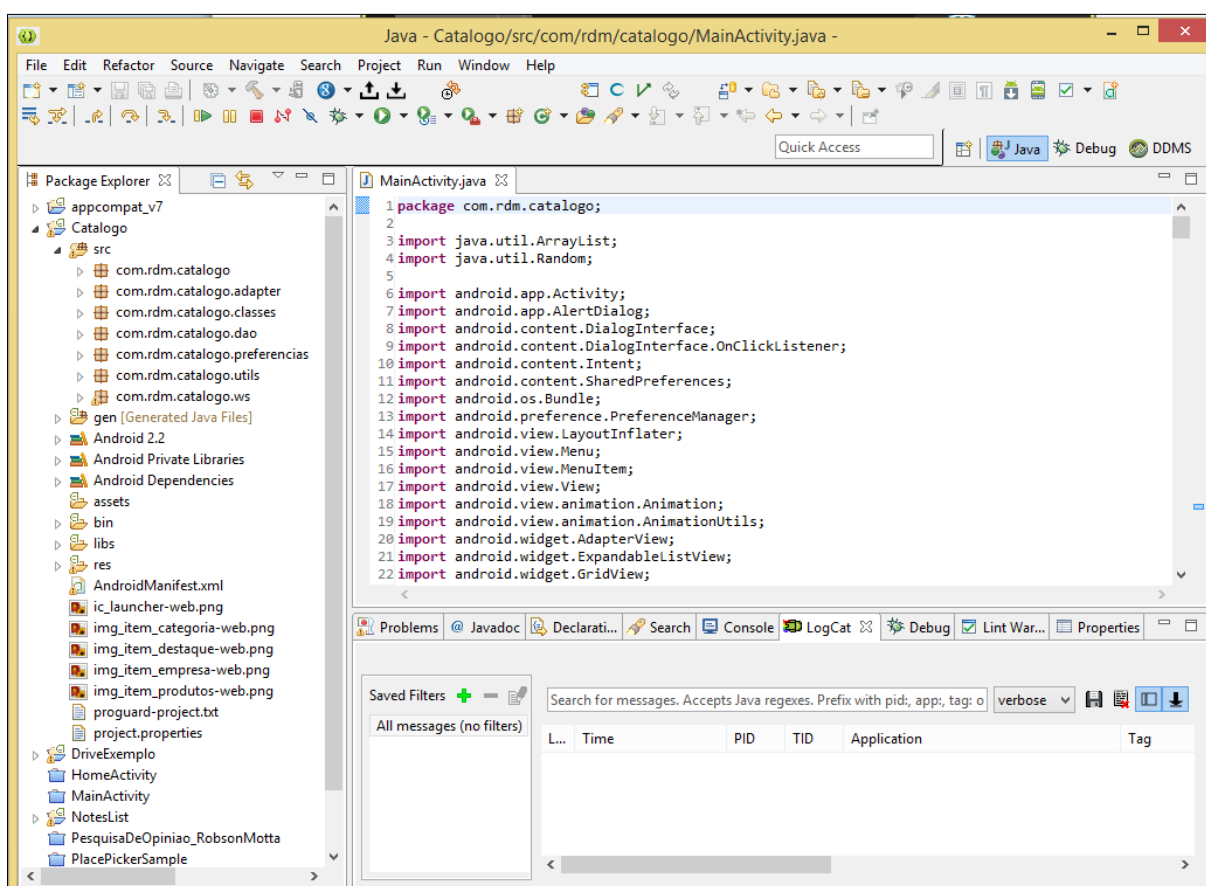


Figura 16: Tela inicial do sistema Eclipse com o *plugin* ADT

Fonte: Autoria própria.

Uma vez adicionado o *plugin* ao sistema Eclipse, o mesmo passa a disponibilizar ao usuário a construção de aplicações específicas para o sistema Android, além de prover os recursos necessários disponibiliza para o usuário o

gerenciamento da AVD. A figura 17 apresenta a AVD onde a mesma emula o sistema operacional Android na versão 4.2.



Figura 17: Exemplo de uma AVD.

Fonte: Autoria própria.

Todas as funcionalidades que um aparelho físico dispõe são encontradas na AVD como a mesma simula um dispositivo real, o teclado físico também está disponível para o uso, assim como as funcionalidades de telas sensíveis ao toque.

Na próxima sessão será relacionado os métodos realizados para obter o resultado desejado no produto, listando os diagramas e requisitos dos sistemas.

3.2 MÉTODOS

Nesta sessão será relacionado os requisitos levantados para a elaboração dos aplicativos propostos no trabalho, a análise levantada para os requisitos tem como base a ação realiza em um processo fictício realizado entre um cliente e um consultor de um catálogo.

3.2.1 Levantamento de Requisitos

Na sequência será abordado os requisitos levantados para o desenvolvimento dos sistemas móvel e *web*. Estão relacionados apenas os requisitos que são classificados como desejáveis ou obrigatórios para a aplicação.

Requisitos Funcionais: Aplicativo móvel, requisitos do sistema.		
Funcionalidade	Descrição	Tipo
F.1	Controle por empresa: Disponibiliza os itens do catálogo de acordo com a empresa selecionada.	Desejável
F.2	Apresentar as informações: Listar os itens disponíveis no catálogo de forma ordenada.	Obrigatório
F.3	Consulta de frete: Disponibilizar o valor do frete para o endereço desejado	Desejável
F.4	Pedido: Proporciona a opção de realizar o pedido do produto que está presente no catálogo	Obrigatório
F.5	Transferência de dados: Sincronizar as informações do aplicativo com o sistema <i>web</i>	Obrigatório

Quadro 3: Requisitos funcionais da aplicação móvel.

Fonte: Autoria própria.

Na sequência será apresentado no quadro 4 os requisitos não funcionais do aplicativo móvel.

Requisitos não funcionais: Aplicativo móvel		
Funcionalidade	Descrição	Tipo
NF.1	Seleção de empresa: Apresentar através de uma janela, as empresas cadastradas no sistema <i>web</i> , uma vez escolhido a empresa, o catálogo deve ser montado com as informações da empresa escolhida.	Desejável

(continua)

NF.2	Itens em destaque: Selecionar de forma randômica os produtos cadastrados e	Desejável
-------------	--	-----------

	apresentar em tela para o usuário.	
NF.3	Catálogo: Utilizar os componentes de lista do dispositivo para apresentar as informações cadastradas no catálogo	Obrigatório
NF.4	Calculo do frete: Para o endereço desejado através do serviço disponibilizado no <i>web service</i> dos correios, calcular e apresentar o valor ao usuário.	Obrigatório
NF.5	Realizar pedido: Enviar as informações do produto escolhido para o <i>web service</i> desenvolvido para a aplicação utilizando protocolo de comunicação HTTP.	Obrigatório

Quadro 4: Requisitos não funcionais da aplicação móvel.

Fonte: Autoria própria.

No quadro 5 estão relacionados os requisitos funcionais para a aplicação *web*.

Requisitos funcionais: Aplicativo web, requisitos do sistema.		
Funcionalidade	Descrição	Tipo
F.1	Controle de <i>Login / Logout</i> do sistema	Obrigatório
F.2	Cadastro de empresas	Obrigatório
F.3	Cadastro de Produtos	Obrigatório
F.4	Listar informações	Desejável

Quadro 5: Requisitos funcionais da aplicação web.

Fonte: Autoria própria.

No quadro 6 é listado os requisitos não funcionais do sistema *web*.

Requisitos não funcionais: Aplicativo web, requisitos do sistema		
Funcionalidade	Descrição	Tipo
NF.1	<i>Login</i> : Criptografar as informações e comparar com os registros na base	Segurança /Obrigatório
NF.2	Sincronismo: Converter as informações da base de dados em formato JSON	Obrigatório
NF.3	Imagens: Serializar as imagens associadas aos produtos para a transferência das informações entre os sistemas.	Obrigatório

(continua)

NF.4	Recebimento do pedido: Gerar um registro para o pedido contendo as informações recebidas do aplicativo móvel	Obrigatório
-------------	--	-------------

Quadro 6: Requisitos não funcionais da aplicação web.

Fonte: Autoria própria.

Para realizar os requisitos levantados o uso da UML auxilia no processo de compreensão e documentação das rotinas funcionais e não funcionais do produto proposto. Na próxima sessão será apresentado os diagramas desenvolvidos para o projeto.

3.2.2 Diagramas do Sistema

Para a documentação do projeto, alguns diagramas de UML foram desenvolvidos, dentre os diagramas de UML disponíveis para o desenvolvimento, foi adotado o diagrama de sequência, diagrama de estado, diagrama de atividade e caso de uso.

Inicialmente será relacionado as rotinas realizadas no sistema web posteriormente as operações disponíveis para o aplicativo móvel . A figura 18 relaciona as ações do usuário no sistema web.

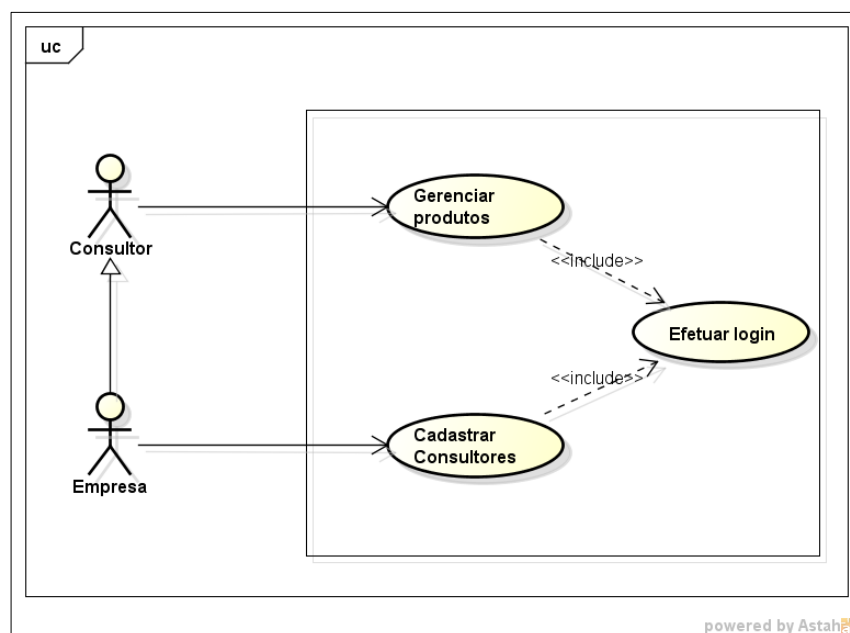


Figura 18: Caso de Uso, Aplicação web.

Fonte: Autoria própria.

Na figura 18 pode ser observada a relação entre o “Consultor” e “Empresa” através do relacionamento de herança, está ligação é para atender a necessidade de tornar o sistema capaz de suportar múltiplos catálogos.

As ações possíveis no sistema *web* são apenas para o gerenciamento das informações, logo, não é possível realizar operações sem estar devidamente registrado no sistema através do *login*.

O gerenciamento dos itens do catálogo está representado no diagrama de caso de uso na figura 19 a seguir.

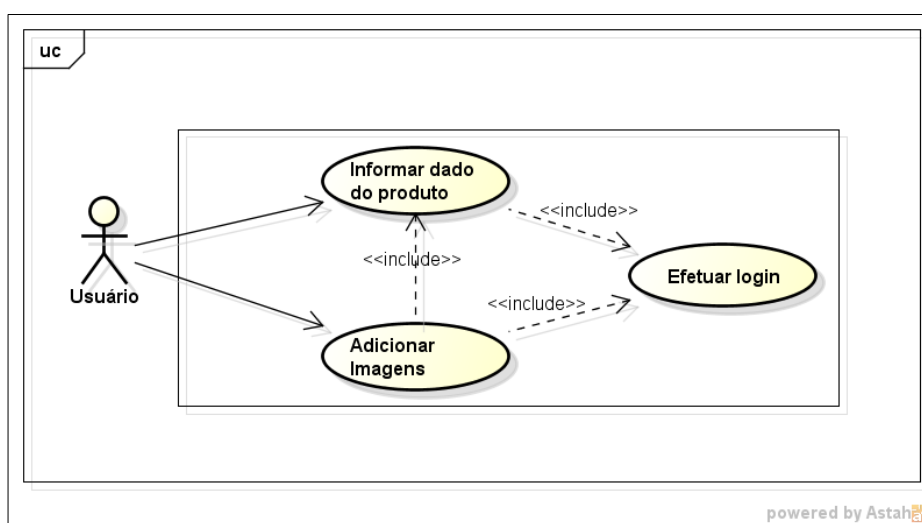


Figura 19: Caso de Uso, Aplicação móvel.

Fonte: Autoria própria.

A maneira como foi documentado o processo de cadastro das informações do produto no qual a funcionalidade de associar imagens aos itens do catalogo está separado da rotina de informar os dados do produto, se deve ao fato de que, durante o desenvolvimento da aplicação, ao utilizar o *framework Primefaces*, o mesmo não apresentou suporte ao gerenciamento de múltiplos objetos em uma mesma sessão.

Na sequência será abordado os diagramas desenvolvidos para a aplicação móvel. Para descrever de forma simples como funciona o uso do aplicativo móvel a figura 20 em seguida relaciona as operações que o usuário da aplicação pode realizar.

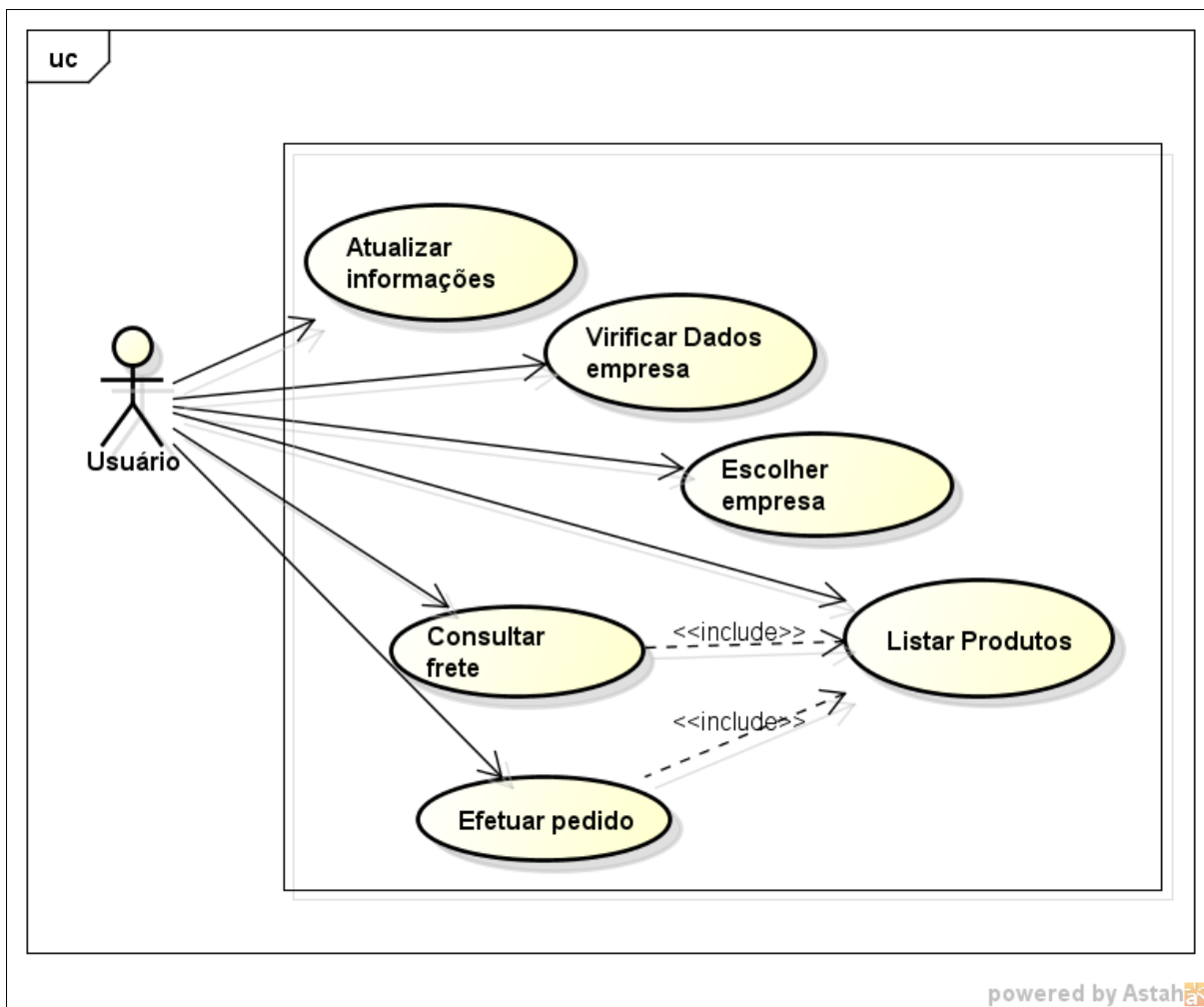


Figura 20: Caso de Uso, Aplicação móvel.
Fonte: Autoria própria.

As funcionalidades disponíveis ao usuário da aplicação móvel são relativamente bem simples, nas funções disponíveis o usuário tem a possibilidade de consultar as informações do catálogo de forma rápida, sendo necessário apenas que o dispositivo móvel tenha realizado pelo menos uma vez a sincronização das informações com o *Web Service*.

Para descrever de forma visual como funciona o processo de realizar pedido o diagrama de sequência presente na figura 21 demonstra de forma simplificada o que ocorre no aplicativo móvel.

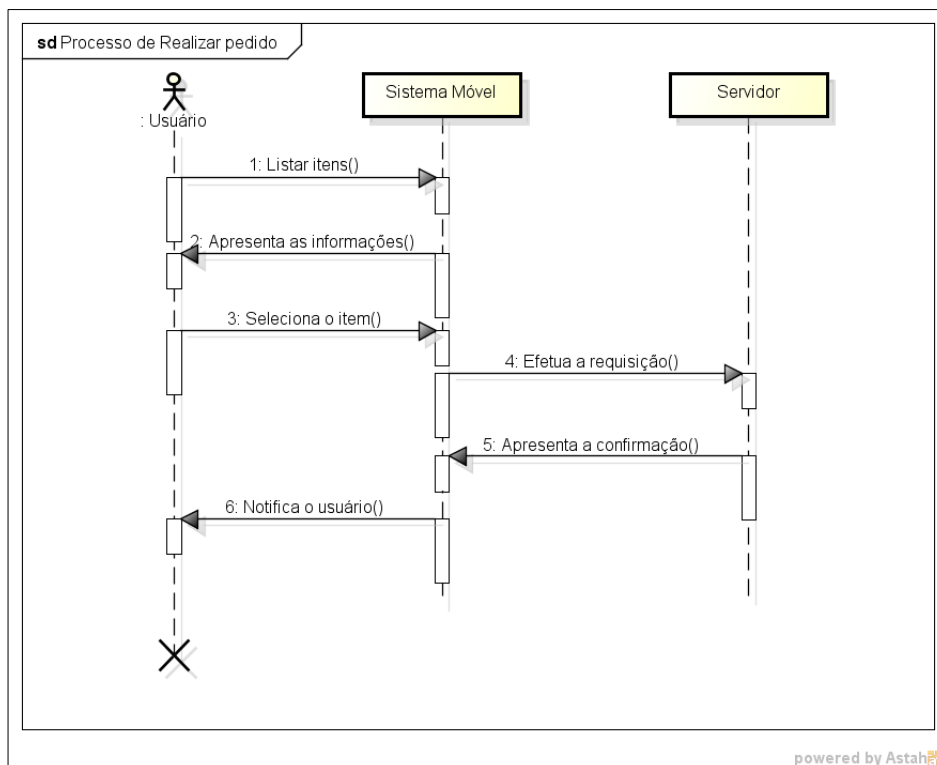


Figura 21: Diagrama de sequência, realizar pedido.

Fonte: Autoria própria.

Os passos necessários para realizar o pedido de um ou mais itens do catálogo é apresentado na figura 21, onde o usuário primeiramente lista as informações do catálogo, logo em seguida seleciona um item para ter acesso as informações detalhadas do mesmo, onde é disponibilizado ao usuário a opção de realizar o pedido do item, uma vez escolhido realizar o pedido o sistema realiza a requisição ao servidor da aplicação, onde está localizado o *Web Service*, este por sua vez retorna a confirmação de que o pedido foi realizado com sucesso.

O fluxo geral de como o sistema móvel funciona é apresentado através do diagrama de atividade na figura 22, o qual descreve as ações realizadas pelo usuário do sistema até o término do fluxo.

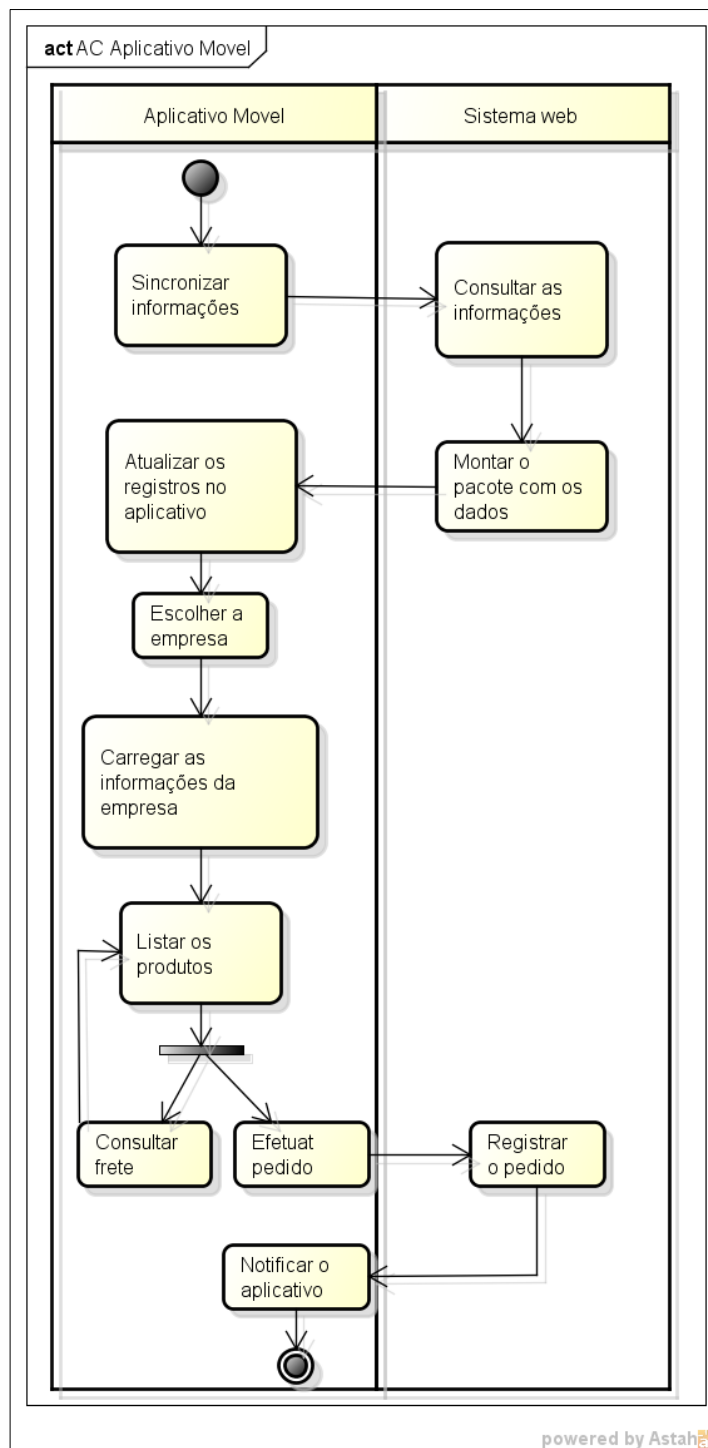


Figura 22: Diagrama de Atividade, Aplicação móvel.
Fonte: Autoria própria.

O termino do fluxo é quando o usuário recebe a confirmação de que seu pedido foi realizado, a aplicação permite que o usuário possa realizar a consulta do frete para todos os itens contidos no catálogo, sem que necessariamente realize o pedido de um item.


4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Este capítulo apresenta de forma sucinta os resultados alcançados no trabalho desenvolvido. Inicialmente será abordado o sistema *web* destacando algumas partes importantes durante o desenvolvimento e na sequência o sistema móvel.

4.1 APRESENTAÇÃO DO SISTEMA WEB

O sistema *web* apresenta um *layout* limpo, sem muitos efeitos visuais, de forma que não o torne poluído para o usuário que operar. Os recursos do *framework Primefaces* dispõem de diversos tipos de temas visuais, o escolhido para a aplicação foi o “Cupertino” como pode ser observado nas figuras que apresentam o sistema.

A fim de realizar o controle por usuário, inicialmente o sistema solicita que o usuário efetue seu acesso por *login*. A figura 23 apresenta a tela de *login*.



A imagem mostra uma captura de tela de um navegador web. No topo, a barra de endereços indica 'localhost:8096/CatalogoWeb/'. Abaixo, há uma barra de tarefas com ícones para 'Aplicativos', 'Entrada - muitopoc...' e 'Bem-vindo ao Face...'. O conteúdo principal da página começa com o título 'Catalogo WEB. Sistema para catalogos' em um cabeçalho cinza. Abaixo, há um menu de navegação com 'Início', 'Cadastro' e 'Listar'. Os campos de entrada são 'Usuário' e 'Chave'. Há botões para 'Login' e 'Limpar'.

Figura 23: Tela de *login* do sistema *web*.

Fonte: Autoria própria

As informações que são requisitadas como credenciais para o acesso são o usuário e chave, onde o campo usuário corresponde ao valor informado no campo

CNPJ (Cadastro Nacional da Pessoa Jurídica) no cadastro da entidade (figura 25) apresentado mais adiante.

A figura 24 a seguir mostra a tela inicial da aplicação com ênfase no menu de cadastros onde são apresentados as opções disponíveis.



Figura 24: Tela inicial do sistema web.

Fonte: A autoria própria.

A figura 24 apresenta as opções de cadastros que estão disponíveis no sistema, bem como a sequência que deve ser seguida para cadastrar as informações pela primeira vez. No cadastro de entidades estão às informações pertinentes a entidade e que são relevantes ao catálogo.

Na figura 25 pode ser observado com detalhes o cadastro de uma entidade.

Figura 25: Tela de cadastro de entidade do sistema web.

Fonte: A autoria própria.

Na aba “Complemento” visualizada na figura 25, estão disponíveis para cadastro outras informações relevantes como endereço, cidade entre outros.

A figura 26 apresenta um exemplo de saída de dados, o qual é possível visualizar os dados cadastrados da entidade.

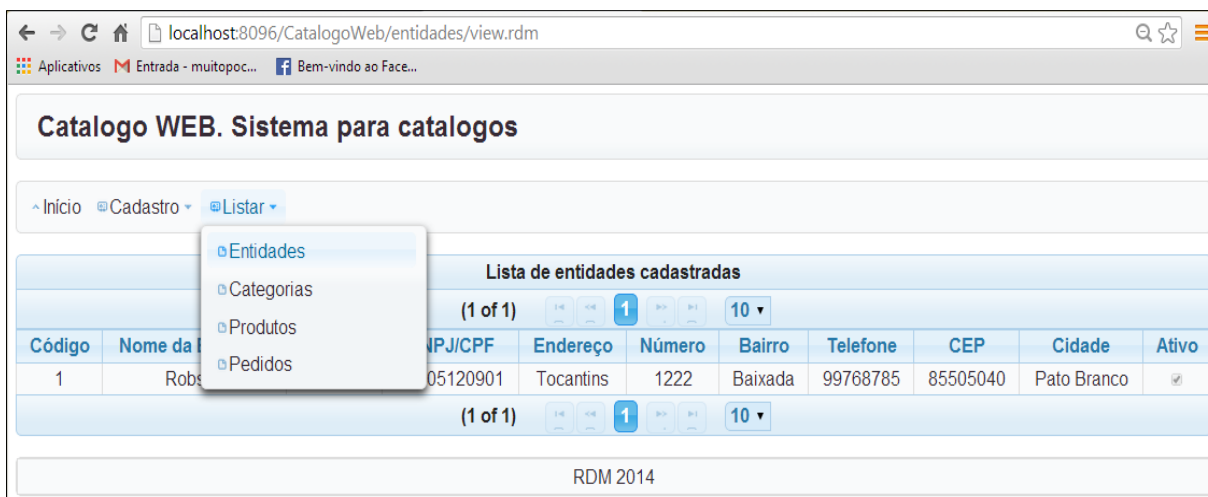


Figura 26: Tela de apresentação de informações (entidades cadastradas).

Fonte: Autoria própria.

A figura 26 demonstra como são exibidas as informações ao usuário do sistema, na representação os dados do cadastro de entidades são listados no *grid*, que pode ser expandido conforme o número de registros cadastrados.

Como padrão as telas de listagem apresentam as informações dispostas em *grid*, o qual possui um sistema de paginação próprio, onde o mesmo realiza o controle para apresentar as informações visuais. Na figura 27 apresenta o recurso descrito anteriormente.

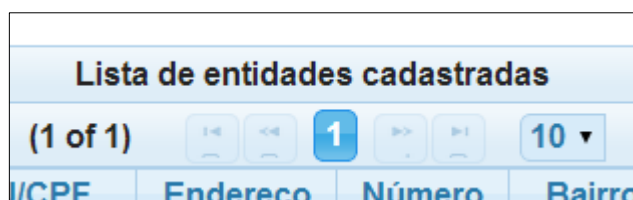
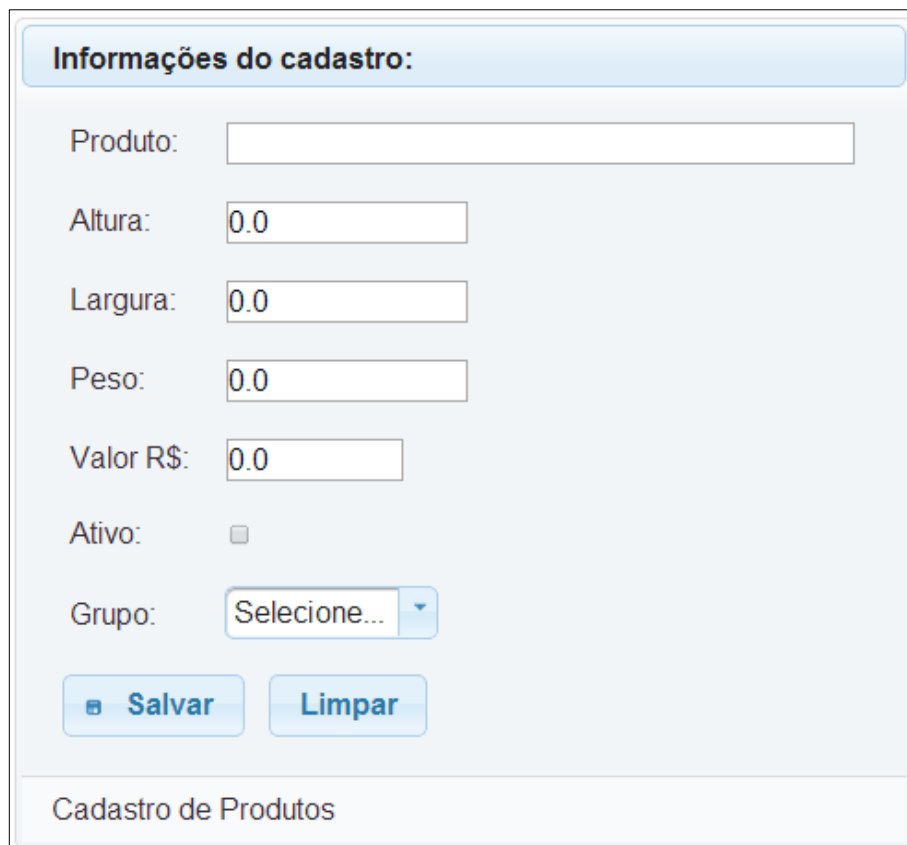


Figura 27: Controle de paginação do *Grid*, disponível nas telas de listagem.

Fonte: Autoria própria.

Observando a figura 27 no lado esquerdo está o controle de páginas apresentadas no *grid*, no centro está exibindo o número da paginação que estão sendo listadas as informações (no exemplo é exibido à página um) e ao lado direito o número dez, indica o máximo de registros que serão apresentados na página.

Na sequência, na figura 28, exibe-se a tela de cadastro de um produto do catálogo contendo algumas informações relevantes para o sistema *web*.



A imagem mostra uma interface web para o cadastro de produtos. O formulário é intitulado "Informações do cadastro:" e contém os seguintes campos:

- Produto: campo de texto vazio.
- Altura: campo de texto com o valor "0.0".
- Largura: campo de texto com o valor "0.0".
- Peso: campo de texto com o valor "0.0".
- Valor R\$: campo de texto com o valor "0.0".
- Ativo: opção de checkbox desativada.
- Grupo: menu suspenso com o texto "Selecione..." e uma seta para baixo.

Abaixo dos campos, há dois botões: "Salvar" (com um ícone de disquete) e "Limpar".

Na base da interface, há uma barra de navegação com o texto "Cadastro de Produtos".

Figura 28: Tela de cadastro de produtos.

Fonte: A autoria própria.

Como são observadas na figura 28, algumas informações como peso altura e largura estão disponíveis para o cadastro, isto se deve ao fato da necessidade de obter a medida do pacote para calcular o valor do frete do produto.

Ao terminar de cadastrar um ou mais produtos, o usuário pode optar por adicionar imagens para que possa ser exibido no aplicativo de catálogo móvel. Este procedimento é realizado no menu de cadastros selecionando a opção de "figuras", como mostra a figura 29 na sequência.

The screenshot shows a web browser window with the address bar displaying 'localhost:8096/CatalogoWeb/produtos/figuras.rdm'. The page title is 'Catalogo WEB. Sistema para catalogos'. The navigation menu includes 'Início', 'Cadastro', and 'Listar'. The main content area is titled 'Adicionar figuras:' and contains a form with the following elements:

- A 'Produto' dropdown menu with the text 'Selecione...'.
- A text area labeled 'Descrição da imagem:' with the placeholder text 'Informe uma descrição para a imagem..'.
- An 'Escolher arquivo' button next to the text 'Nenhum arquivo selecionado'.
- A 'Gravar Imagens' button.

Below the form, there is a section titled 'Cadastro de figuras para produtos' and a footer with the text 'RDM 2014'.

Figura 29: Cadastro de figuras para os produtos pré-cadastrados.

Fonte: Autoria própria.

A figura 29 apresentada anteriormente exibe a tela de cadastro de figuras para os produtos pré-cadastrados no sistema, onde é possível adicionar uma pequena descrição para cada imagem carregada na aplicação.

Todas as classes da aplicação, implementam o método “toJSON” o qual converte os valores em uma instância do objeto JSON, possibilitando a comunicação entre a aplicação móvel e *web* mais simplificada.

O quadro 7 a seguir exemplifica a implementação do método em uma classe do sistema.

```
80      @Override
81      public JSONObject toJSON() {
82          JSONObject json = new JSONObject();
83          try {
84              json.put("cidade_id", idCidade);
85              json.put("cidade_nome", nome);
86              json.put("cidade_uf", uf);
87              json.put("cidade_codigo", codigo);
88          } catch (JSONException ex) {
89              return null;
90          }
91          return json;
92      }
93  }
```

Quadro 7: Exemplo de método que retorna uma instancia do objeto no formato JSON.

Fonte: Autoria própria.

De acordo com a documentação disponível no site “*json.org*” a notação para o objeto *JSON* deve possuir uma chave de identificação e um valor para a mesma, como mostra no quadro 7, o sistema adota o seguinte critério para as chaves usadas (nome da tabela do banco de dados e nome da coluna referente à tabela, separando as palavras com o símbolo “_”).

Outro trecho do código que é interessante destacar é o método responsável por retornar as imagens dos produtos, o método é implementado no *Web Service REST* e pode ser acessível através da URL “*http://localhost:8089/CatalogoWeb/api/Imagens/Produto?id=1*”, onde (*localhost:8089*) pode ser substituído pelo endereço onde o serviço estiver hospedado e o parâmetro “*id=1*” pelo respectivo código do produto cadastrado no banco de dados.

No quadro 8 na sequência apresenta o trecho codificado para converter as imagens em um objeto na notação *JSON* para posteriormente ser retornado na requisição do método do *Web Service*.

```

50  @GET
51  @Path("/produto")
52  @Produces("application/json")
53  public Response getImagens(@QueryParam("id") @DefaultValue("0") String id) {
54
55      Produto produto = (Produto) DAO.getProdutoDAO(Conexao.getConexao()).findObjectforID(Integer.valueOf(id));
56      List<ICatalogo> list = DAO.getImagemProdutoDAO(Conexao.getConexao()).findAll(produto);
57
58      JSONArray array = new JSONArray();
59      for (ICatalogo imagem : list) {
60          JSONObject object = new JSONObject();
61
62          try {
63              object.put("imagensproduto_id", ((ImagemProduto) imagem).getIdImagens());
64              object.put("imagensproduto_id_produto", ((ImagemProduto) imagem).getIdProduto());
65              object.put("imagensproduto_id_entidade", ((ImagemProduto) imagem).getIdEntidade());
66
67              String path = ((ImagemProduto) imagem).getSrcPath();
68
69              BufferedImage img = null;
70              try {
71                  FileImageInputStream is = new FileImageInputStream(new File(path));
72                  String imgSTR = "";
73                  byte[] buffer = new byte[(int) is.length()];
74                  int bytesRead = 0;
75                  while (true) {
76                      bytesRead = is.read(buffer);
77                      if (bytesRead > 0) {
78                          imgSTR = Base64.encode(buffer);
79                      } else { break; }
80                  }
81                  object.put("objeto_imagens", imgSTR);
82              } catch (IOException ex) { }
83
84          } catch (JSONException ex) { }
85          array.put(object);
86      }
87      return Response.ok(array.toString()).encoding("UTF-8").build();
88  }
89

```

Quadro 8: Exemplo de método no recurso que retorna as imagens dos produtos cadastrados.

Fonte: Autoria própria.

Os arquivos de imagens não são salvos “fisicamente” no banco de dados o que realmente fica armazenado é o diretório onde as mesmas foram copiadas. No quadro 8, após selecionar as informações o método busca a imagem obtida no diretório (linha 67 do quadro 4) e armazena em *buffer* para posteriormente converter um uma cadeia de caracteres *String* através da classe *Base64* (linhas 75 até 80 do quadro 8).

A figura 30 apresenta o formato do retorno após a chamada do método do *Web Service* para listar as imagens do produto.

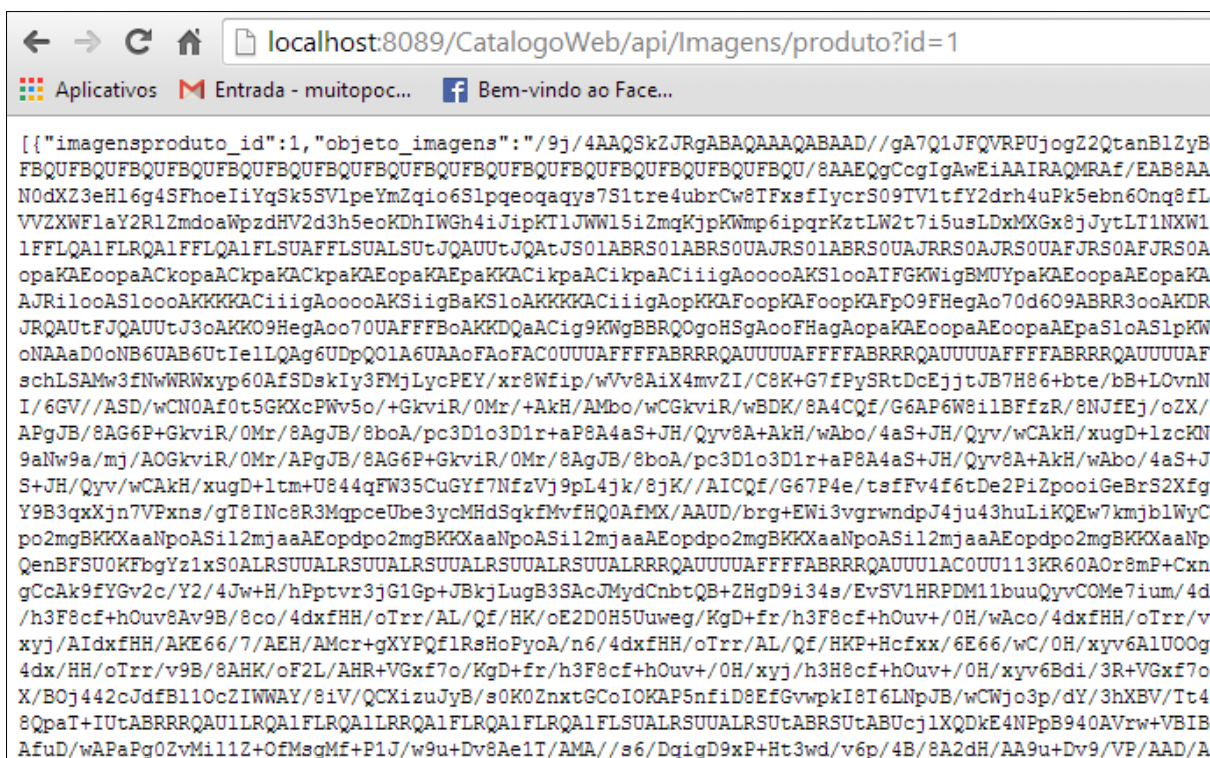


Figura 30: Exemplo de resposta do método `getImagens` no recurso do Web Service REST.

Fonte: Autoria própria.

Observando a figura 30 nota-se que os *bits* da imagem estão convertidos em texto. Quando o dispositivo móvel recebe a resposta da requisição, o mesmo inicia o processo reverso apresentado no quadro 4, a fim de obter uma cópia da imagem assim pode armazenar a mesma para que possa ser visualizada pelo aplicativo.

4.2 APRESENTAÇÃO DO SISTEMA MÓVEL

O sistema para dispositivos móveis apresenta os recursos necessários propostos para o desenvolvimento do projeto (vide capítulo 3). Disponibiliza uma interface simples e de fácil interpretação como são propostas nas normas de IHC.

Na sequência serão apresentadas as telas do aplicativo e juntamente alguns trechos de código que são relevantes para a compreensão do funcionamento do mesmo. A primeira figura que será abordada é a tela de configurações do aplicativo, tal qual pode ser armazenado as informações do endereço do servidor REST onde o mesmo realiza as chamadas para popular os dados ou registrar os pedidos do catálogo.

A figura 31 que segue demonstra a tela de configuração.



Figura 31: Tela de configurações do aplicativo móvel.
Fonte: Autoria própria.

Como é observado na figura 31 está disponível para cadastro as informações necessárias para a conexão com o servidor além de uma opção para sempre visualizar a tela ao abrir o aplicativo.

Na figura 32 pode ser observada a tela principal do aplicativo, onde o mesmo é dividido em três partes, cada qual separada em uma aba na região superior da tela do dispositivo.



Figura 32: Tela principal do aplicativo móvel.

Fonte: Autoria própria.

Como pode ser visto na figura 32 as abas estão divididas entre destaque, produtos e empresa, as quais serão abordadas na sequência. Também é possível observar que algumas imagens de produtos estão dispostas na tela, estas figuras são apresentadas de maneira aleatória e estão com a finalidade de exibir alguns dos produtos em destaque.

O quadro 9 na sequência apresenta parte do código responsável por selecionar as figuras para exibição no aplicativo.

```

44⇒ public ImagensProduto listar(int id) {
45     ImagensProduto imagensProduto = new ImagensProduto();
46     ;
47     List<Bitmap> imagens = new ArrayList<Bitmap>();
48
49     String[] columns = { Columns.ID, Columns.IDENTIDADE, Columns.IDPRODUTO,
50         Columns.SRCIMAGEM };
51     try {
52         SQLiteDatabase db = getDbInstance().getReadableDatabase();
53
54         Cursor c = db.query(DBHelper.TABELA_PRODUTOIMAGENS, columns,
55             Columns.IDPRODUTO + " = ?",
56             new String[] { String.valueOf(id) }, null, null, null);
57         try {
58             while (c.moveToNext()) {
59
60                 imagensProduto
61                     .setId(c.getInt(c.getColumnIndex(Columns.ID)));
62                 imagensProduto.setIdEntidade(c.getInt(c
63                     .getColumnIndex(Columns.IDENTIDADE)));
64                 imagensProduto.setIdProduto(c.getInt(c
65                     .getColumnIndex(Columns.IDPRODUTO)));
66
67                 String src = c.getString(c
68                     .getColumnIndex(Columns.SRCIMAGEM));
69                 Bitmap bMap = BitmapFactory.decodeFile(src);
70                 imagens.add(bMap);
71             }
72             imagensProduto.setImagens(imagens);
73
74         } finally {
75             if (db.isOpen()) {
76                 db.close();
77             }
78             if (c != null) {
79                 c.close();
80             }
81         }
82     } catch (Exception e) {
83     }
84
85     return imagensProduto;
86 }
87

```

Quadro 9: Trecho do código que recupera as imagens no aplicativo móvel.

Fonte: Autoria própria.

Como pode ser observado na linha 67 do quadro 9, através do diretório onde está salvo a imagem no dispositivo que é recuperado do banco de dados usando do método estático “*decodeFile*” (linha 69 do quadro 9) uma imagem é carregada e armazenada na lista (linha 70 no quadro 9), para posteriormente ser apresentada no aplicativo.

A forma como é apresentado as categorias de cada produto no aplicativo está disposta em um *grid*, onde o mesmo apresenta o nome e uma imagem para melhor visualização.

A figura 33 apresenta um esboço de como é apresentado ao usuário.

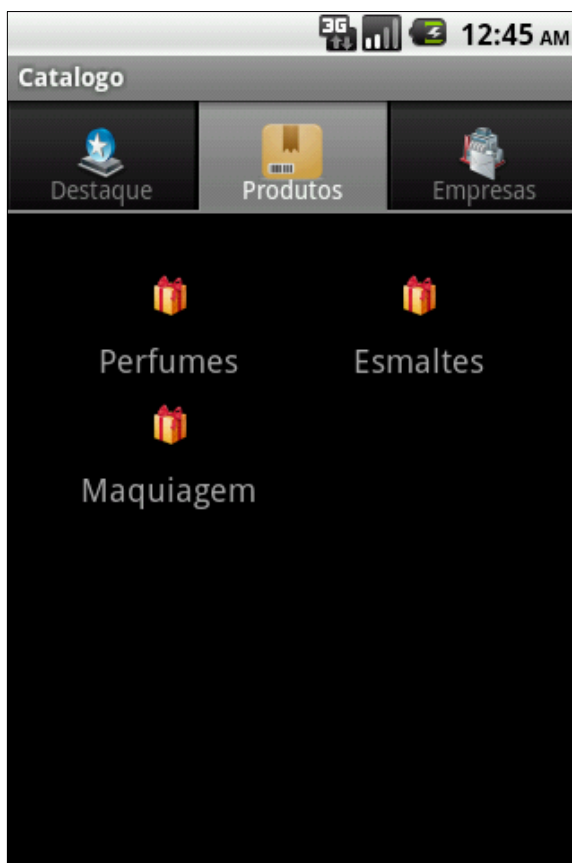


Figura 33: Listagem das categorias no aplicativo móvel.

Fonte: Autoria própria.

Como é observado na figura 33 cada categoria pré-cadastrada no sistema *web* é apresentada no aplicativo móvel e a mesma redireciona para a visualização dos respectivos produtos relacionados.

Na figura 34 é possível visualizar um exemplo de listagem de produtos da categoria “Esmaltes” visualizados na figura 33 anteriormente os quais são ordenados e apresentados por ordem de cadastro no sistema.

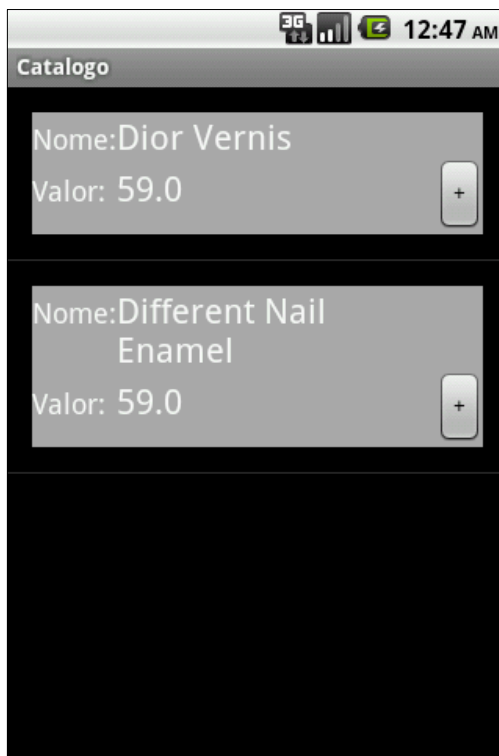


Figura 34: Listagem dos produtos da categoria selecionada.

Fonte: Autoria própria.

A listagem apresenta como informação o nome e o valor de cada produto, e para cada item da lista está disponível um botão com o símbolo “+” onde o mesmo redireciona para outra tela, a qual disponibiliza mais detalhes de cada item.

A figura 35 exhibe o resultado obtido ao pressionar o botão citado anteriormente.



Figura 35: Tela de detalhes do item consultado.

Fonte: Autoria própria.

Na figura 35 entre as informações disponíveis para visualização como as imagens relacionadas ao item, há dois botões: um para a consulta do frete para determinado endereço e respectivamente um botão para realizar o pedido do item.

Na sequência será apresentada a figura 36 com a tela de consulta do valor do frete para o item.

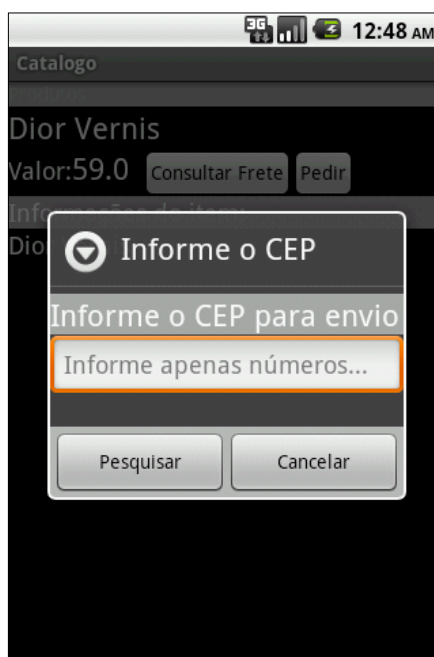


Figura 36: Tela de consulta de valores de frete.

Fonte: Autoria própria.

Quando informado o número do CEP (Código de Endereçamento Postal) para pesquisa, o sistema realiza a consulta através do site dos correios e retorna as informações para apresentar ao usuário, como pode ser observado na figura 37.

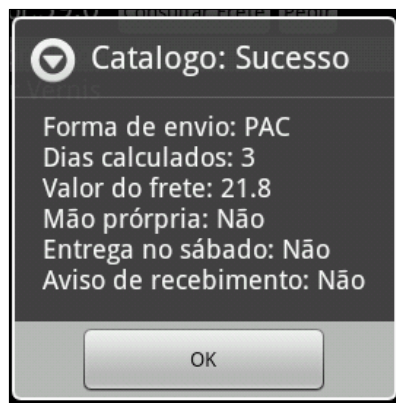


Figura 37: Informações retornadas na consulta de valores de frete.

Fonte: Autoria própria.

Entre as informações relevantes recuperadas na figura 37, está sendo apresentado o valor do frete para o endereço solicitado, o tipo de envio da mercadoria que por padrão foi adotado a forma PAC (sigla para Prático, Acessível e Confiável) o qual é um serviço de encomenda econômica, disponibilizado pelos correios, além de apresentar o valor do serviço. Todas as informações são apenas para conhecimento do usuário sobre o produto consultado.

Na sequência é apresentado o quadro 10 com um trecho do código onde é montado os parâmetros de consulta ao site dos correios afim de obter o valor do frete para o item desejado.

```

233 public String prepareUrlFrete() {
234     // Monta a URL com as informações necessárias;
235     String cepOrigem = produto.getEmpresa().getCep();
236     String vlrPeso = String.valueOf(produto.getPeso());
237     String formaPacote = "1"; // Tipos: 1 para caixa / pacote e 2 para
238         // rolo/prisma;
239     int comprimento = 16; // Comprimento não pode ser inferior a 16 cm;
240     String vlrComprimento = String.valueOf(comprimento);
241     String vlrAltura = String.valueOf(produto.getAltura());
242     String vlrLargura = String.valueOf(produto.getLargura());
243     String maoPropria = "N";
244     String vlrDeclarado = String.valueOf(produto.getValor());
245     String avisoRecebimento = "N";
246     String valorDiametro = "0";
247     String tipoRetorno = "xml";
248     StringBuilder strParametros = new StringBuilder();
249
250     strParametros.append("sCepOrigem=" + cepOrigem)
251         .append("&sCepDestino=" + getCepDestino())
252         .append("&nVlPeso=" + vlrPeso)
253         .append("&nCdFormato=" + formaPacote)
254         .append("&nVlComprimento=" + vlrComprimento)
255         .append("&nVlAltura=" + vlrAltura)
256         .append("&nVlLargura=" + vlrLargura)
257         .append("&sCdMaoPropria=" + maoPropria)
258         .append("&nVlValorDeclarado=" + vlrDeclarado)
259         .append("&sCdAvisoRecebimento=" + avisoRecebimento)
260         .append("&nCdServico=" + getCodigoFormaEnvio()) //41106 PAC sem contrato
261         .append("&nVlDiametro=" + valorDiametro)
262         .append("&StrRetorno=" + tipoRetorno);
263     return URL_WS + strParametros.toString();
264
265 }

```

Quadro 10: Montagem dos parâmetros de consulta do frete.

Fonte: Autoria própria.

Como é observado no quadro 10, entre as linhas 250 e 260 estão sendo valorizados os parâmetros para adicionar a URL do *Web Service* dos correios (ws.correios.com.br/calculador/calcprecoprazo.asmx) o qual retorna as informações em formato de XML (*Extensible Markup Language*) para poder ser processada pelo sistema e apresentado ao usuário.

No quadro 11 na sequência, apresenta parte do código responsável por se conectar ao *Web Service* e interpretar as informações recebidas.

```

50  @GET
51  @Path("/produto")
52  @Produces("application/json")
53  public Response getImagens(@QueryParam("id") @DefaultValue("0") String id) {
54
55      Produto produto = (Produto) DAO.getProdutoDAO(Conexao.getConexao()).findObjectforID(Integer.valueOf(id));
56      List<ICatalogo> list = DAO.getImagemProdutoDAO(Conexao.getConexao()).findAll(produto);
57
58      JSONArray array = new JSONArray();
59      for (ICatalogo imagem : list) {
60          JSONObject object = new JSONObject();
61
62          try {
63              object.put("imagensproduto_id", ((ImagemProduto) imagem).getIdImagens());
64              object.put("imagensproduto_id_produto", ((ImagemProduto) imagem).getIdProduto());
65              object.put("imagensproduto_id_entidade", ((ImagemProduto) imagem).getIdEntidade());
66
67              String path = ((ImagemProduto) imagem).getSrcPath();
68
69              BufferedImage img = null;
70              try {
71                  FileInputStream is = new FileInputStream(new File(path));
72                  String imgSTR = "";
73                  byte[] buffer = new byte[(int) is.length()];
74                  int bytesRead = 0;
75                  while (true) {
76                      bytesRead = is.read(buffer);
77                      if (bytesRead > 0) {
78                          imgSTR = Base64.encode(buffer);
79                      } else { break; }
80                  }
81                  object.put("objeto_imagens",imgSTR);
82              } catch (IOException ex) { }
83
84          } catch (JSONException ex) { }
85          array.put(object);
86      }
87      return Response.ok(array.toString().encoding("UTF-8").build());
88  }
89

```

Quadro 11: Retorno da requisição das imagens pertencentes aos itens.

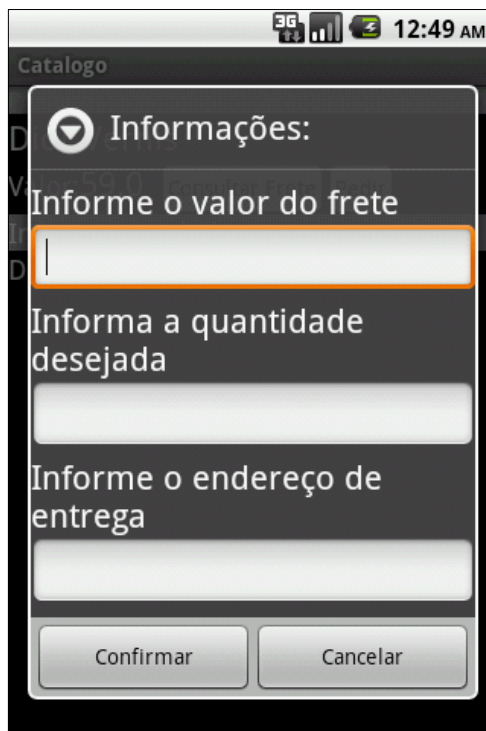
Fonte: Autoria própria.

No quadro 11 apresentado anteriormente, pode ser observado entre as linhas 55 e 56 a consulta no banco de dados das informações enviadas como parâmetros na chamada da requisição, logo após entre as linhas 58 e 86 o objeto JSON é construído para ser retornado ao aplicativo móvel.

Outro ponto que é importante destacar é a possibilidade de realizar os pedidos com os itens cadastrados no aplicativo. Para tanto foi implementado uma tela onde é possível informar os dados necessários para a requisição de um pedido no aplicativo.

Para que possa ser registrado o pedido é necessário informar o valor de frete para o endereço, a quantidade desejada do produto e uma descrição do endereço de entrega.

Na figura 38, apresenta a tela visualizada ao pressionar o botão “Pedir” relacionado na figura 35 anterior.



The image shows a mobile application interface with a dialog box titled "Informações:". The dialog box contains three input fields: "Informe o valor do frete" (highlighted with an orange border), "Informe a quantidade desejada", and "Informe o endereço de entrega". At the bottom of the dialog box are two buttons: "Confirmar" and "Cancelar". The background of the application is dark, and the status bar at the top shows the time as 12:49 AM and the signal strength as 3G.

Figura 38: Tela para realizar os pedidos.
Fonte: Autoria própria.

Após a confirmação do pedido pelo usuário através do botão "Confirmar" presente na figura 38, o usuário do aplicativo poderá interagir com o aplicativo podendo realizar novos pedidos.

5 CONSIDERAÇÕES FINAIS

Como resultados pode se considerar que os requisitos desejados nas duas aplicações tanto *web* quanto móvel, apresentaram suas devidas funcionalidades, podendo, portanto atender as necessidades propostas para ambos os ambientes de desenvolvimento.

Quando aderido o uso de um aplicativo com as características de um catálogo eletrônico como o desenvolvido no trabalho, as empresas que disponibilizam por meios físicos seus catálogos de produtos demonstram suas preocupações com diversos fatores que envolvem políticas ambientais e/ou econômicas.

Em relação à hipótese apresentada no trabalho, podemos entender que a maneira como o projeto foi desenvolvido e utilizado em conjunto com mídias existentes contribui para que sua popularização alcance nas massas, entre as diferentes plataformas do sistema operacional Android existentes atualmente para dispositivos móveis.

REFERÊNCIAS

[CETIC] Disponível em <<http://www.cetic.br/media/docs/publicacoes/2/tic-centros-publicos-de-acesso-2013.pdf>> Acessado em 20 de Outubro de 2014.

[COOPERATI] Disponível em <<http://www.cooperati.com.br/2012/05/10/porque-usar-linux-ou-o-open-source/>> Acessado em 26 de Agosto de 2014.

[COMSCORE] Disponível em <http://www.brainstorm9.com.br/wp-content/uploads/2014/05/2014_Brazil_Digital_Future_in_Focus_PT.pdf> Acessado em 07 de Outubro de 2014.

[COPA] Disponível em <<http://www.copa2014.gov.br/pt-br/noticia/ministerio-da-saude-lanca-aplicativo-para-tablet-e-smartphone-que-sera-usado-na-copa>> Acessado em 20 de Abril de 2014.

[DSC] Disponível em <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>>. Acessado em 15 de Abril de 2014.

Erl, Tomas. Introdução às tecnologias Web Services: SOA, SOAP, WSDL e UDDI. Disponível em <<http://www.devmedia.com.br/introducao-as-tecnologias-web-services-soa-soap-wsdl-e-uddi-parte1/2873>> Acessado em 02 de Dezembro de 2014.

FERREIRA, Aurélio Buarque de Holanda. Aurélio Júnior: dicionário escolar da língua portuguesa. Curitiba: Positivo 2011.

Fielding, Roy Thomas (2000), Architectural Styles and the Design of Network-based Software Architectures, Tese de Doutorado, California, USA

GIL Antonio Carlos, 1946 - **Como elaborar projetos de pesquisa/Antonio Carlos Gil**. - 4. ed. - São Paulo: Atlas, 2002.

NEVES, José Luis. Pesquisa qualitativa: Características, uso e possibilidades. **Caderno de Pesquisas em Administração**, São Paulo, v. 1, n. 3, 1996. Disponível em <<http://www.ead.fea.usp.br/cad-pesq/arquivos/C03-art06.pdf>>. Acesso em 27 jan 2014.

[O GLOBO] Disponível em <<http://oglobo.globo.com/sociedade/tecnologia/numero-de-internautas-no-brasil-alcanca-percentual-inedito-mas-acesso-ainda-concentrado-13027120>> Acessado em 20 de Outubro de 2014.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de informação Gerenciais: estratégias, táticas operacionais**. 8. Ed., São Paulo: Atlas, 1992.

Six, Jeff, 2012 - Segurança de aplicativos Android/ Jeff Six ; [tradução Eduardo Kraszczuk]. São Paulo : Novatec Editora ; Sebastopol, CA : O'Reilly, 2012

[TENDENCIA] Disponível em
<<http://blogbrasil.comstora.com/blog/bid/386298/6-tend%C3%A2ncias-de-mercado-focadas-em-profissionais-de-TI>>. Acessado em 15 de Abril de 2014.

[UOL TEC] Disponível em
<<http://tecnologia.uol.com.br/noticias/redacao/2013/08/28/brasil-venda-de-smartphones-cresce-110-no-ano-e-supera-celulares-comuns.htm>>. Acessado em 27 Janeiro 2014.

Link's para download dos aplicativos descritos no trabalho:

[NDM] ND mobile, 2014. Disponível em
<<http://ndmobile.com.br/catalogo/screenshots.html>> Acessado em 23 de novembro de 2014.

[MLAPP] Magazine Luiza, 2014. Disponível em
<<https://play.google.com/store/apps/details?id=com.luizalabs.mlapp>> Acessado em 23 de Novembro de 2014.



DECLARAÇÃO DE AUTORIA

Autor¹: Robson Da Motta

CPF¹: 069.051.209-01 Código de matrícula¹: 1490460

Telefone¹: (46) 9976-8785 e-mail¹: muitopoco@gmail.com

Curso/Programa de Pós-graduação: Especialista em Desenvolvimento de Sistemas para Internet e Dispositivos Móveis

Orientador: Géri Natalino Dutra

Co-orientador: _____

Data da defesa: 20/11/2014

Título/subtítulo: Catálogo para dispositivos móveis

Tipo de produção intelectual: () TCC² (X) TCCE³ () Dissertação () Tese

Declaro, para os devidos fins, que o presente trabalho é de minha autoria e que estou ciente:

- dos Artigos 297 a 299 do Código Penal, Decreto-Lei nº 2.848 de 7 de dezembro de 1940;
- da Lei nº 9.610, de 19 de fevereiro de 1998, sobre os Direitos Autorais,
- do Regulamento Disciplinar do Corpo Discente da UTFPR; e
- que plágio consiste na reprodução de obra alheia e submissão da mesma como trabalho próprio ou na inclusão, em trabalho próprio, de idéias, textos, tabelas ou ilustrações (quadros, figuras, gráficos, fotografias, retratos, lâminas, desenhos, organogramas, fluxogramas, plantas, mapas e outros) transcritos de obras de terceiros sem a devida e correta citação da referência.

Assinatura do Autor¹

Local e Data

¹ Para os trabalhos realizados por mais de um aluno, devem ser apresentados os dados e as assinaturas de todos os alunos.

² TCC – monografia de Curso de Graduação.

³ TCCE – monografia de Curso de Especialização.