UNIVERSIDADE FEDERAL DO PARANÁ

PAULO VITOR DOS SANTOS ZEFERINO

GESTÃO DE VENDAS COM MOBILIDADE

CURITIBA

2011

PAULO VITOR DOS SANTOS ZEFERINO

GESTÃO DE VENDAS COM MOBILIDADE

Dissertação apresentada ao Programa de Pós-Graduação Engenharia de Software da Universidade Federal do Paraná, como requisito para obtenção do título de Especialista em Engenharia de Software.

Orientador: Prof. MSc. Jaime Wojciechowski.

CURITIBA

2011

13-10.

TERMO DE APROVAÇÃO

PAULO VITOR DOS SANTOS ZEFERINO

GESTÃO DE VENDAS COM MOBILIDADE

Monografia apresentada como requisito parcial para a obtenção da titulação de especialista, pelo Curso de Pós-Graduação Lato Sensu em Engenharia de Software, da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador: Professor Jaime Wojciechowski

Curitiba, 30 de SETEMBRO de 2011.

LISTA DE ABREVIATURAS E SIGLAS

Anatel	Agência Nacional de Telecomunicações
API	Interface de Programação de Aplicações
CDC	*Connected Device Configuration* (sem tradução para o português)
CLDC	*Connected Limited Device Configuration* (sem tradução para o português)
ERP	*Enterprise Resource Planning* (sem tradução para o português)
FP	*Foundation Profile*
GPS	Sistema de Posicionamento Global
IDE	Interface de Desenvolvimento Integrado
JME	Java 2 Micro Edition
JSON	Objeto de Notação JavaScript
JSR	Requisição de Especificação Java
JVM	Máquina Virtual Java
LBS	Serviços Baseados em Localização
KVM	*Kilobyte Virtual Machine*
MIDP	Perfil de Informações do Dispositivo Móvel
MMAPI	*Mobile Media API* (sem tradução para o português)
OMG	*Object Management Group* (sem tradução para o português)
PBP	*Personal Basis Profile*
PDA	Assistente Pessoal Digital
PDV	Ponto de Venda
PHP	*Hypertext Preprocessor* (sem tradução para o português)
PP	*Personal Profile*
RFC	*Request for Comments* (sem tradução para o português)
RMS	*Record Management System* (sem tradução para o português)
RUP	*Rational Unified Process*
TI	Tecnologia da Informação
UML	Linguagem de Modelagem Unificada
VM	Máquina Virtual
XML	Linguagem de Marcação Extensível

LISTA DE FIGURAS

SUMÁRIO

RESUMO

O paradigma da mobilidade computacional cresce hoje no mundo em um ritmo muito acelerado, fazendo com que os setores da economia estejam sempre absorvendo os benefícios de tais avanços. Utilizando a tecnologia JME é possível desenvolver aplicações que atingem um grande número de dispositivos móveis existentes no mercado de hoje. E desta maneira, o desenvolvimento de um sistema de *front-end* móvel voltado para a gestão de pontos de venda se torna necessário para o setor, tendo em vista que as tarefas da área exigem constantes movimentações dos envolvidos e as tomadas de decisões devem ocorrer de maneira rápida e efetiva. Uma solução para este problema é a possibilidade do registro e envio das diversas informações coletadas durante as atividades de trabalho para aplicações corporativas (disponíveis na chamada nuvem computacional) através dos dispositivos móveis, independente da localização em que o usuário encontra-se, possibilitando queda no tempo de comunicação entre os envolvidos. A integração do JME com outras tecnologias visando que o processo seja atendido de maneira satisfatória contará com a exploração de recursos como GPS e câmera fotográfica nos aparelhos celulares e *WebServices* para o transporte de dados entre sistemas utilizando o padrão JSON. Trabalhando as tecnologias apresentadas em conjunto, será possível a elaboração de um aplicativo móvel que torne o trabalho dos funcionários de campo e de gestão das empresas mais ágil e eficiente, proporcionando facilidades no processo de coleta/envio de dados e também na disponibilização das informações a quem for necessário. Isso acarretará no aumento de produtividade e principalmente dos lucros das empresas que buscarem a informatização e mobilidade de seus negócios.

Palavras-chave: JME. Computação móvel. Dispositivos móveis. Ponto de Venda.

ABSTRACT

The paradigm of computing mobile increases today in the world in a very fast way, making the economy sectors always incorporate the benefits of such advances. Using JME technology it's possible to develop applications that affect a large number of mobile devices on the market in the present day. By the way, the development of a front-end mobile system turned to point of sale's management becomes necessary to the sector, provided that the tasks in this area require constant movement of the involved people and the decisions must occur quickly and effectively. A solution to this problem is the possibility of recording and transmission of the information collected during the work activities for corporate applications (available on the cloud) via mobile devices, independent of the location where the user is, thus allowing shorter communication times between those involved. The integration of JME with other technologies aiming at a satisfactory execution of the process will include the exploitation of resources such as camera and GPS in cell phones and web services for transporting data between systems using the JSON standard. By means of using all the presented technologies, it will be possible to develop a mobile application that will make the work of field employees and management of the companies more agile and efficient, providing facilities in the process of collecting / sending data and also in making the information available to the people who need it. This will result in increased productivity and profits for the companies that seek to computerize and mobility their business.

Keywords: JME. Mobile Computing. Mobile Devices. Point of Sale.

1. INTRODUÇÃO

1.1 DOMÍNIO DO PROBLEMA

Hoje no Brasil existem aproximadamente 208 milhões de aparelhos celulares segundo dados da Agência Nacional de Telecomunicações (Anatel) de fevereiro de 2011 e este expressivo número continua em franca ascensão. Tais números só reforçam a idéia de integração que todo o mundo se encontra atualmente. E o setor de vendas não pode ficar fora do rol de áreas que vem aderindo às soluções móveis em busca de melhores resultados.

Os Pontos de Venda necessitam de soluções inteligentes que permitam uma gestão eficiente das informações a fim de proporcionar um melhor desempenho aos gestores, auditores, promotores, equipes comerciais e de vendas e todos os demais envolvidos no processo, possibilitando uma ampliação das oportunidades de negócio.

As tomadas de decisões nesta área devem ocorrer de maneira muito rápida e eficiente. E para isto ocorrer os gestores devem possuir as informações ao seu alcance em tempo real para ser possível gerar relatórios inteligentes e precisos demonstrando os rumos que a empresa vem seguindo, como os concorrentes encontram-se no mercado, o desempenho de cada funcionário, entre outras situações que podem ser apresentadas dentro de cada organização.

Este ramo de trabalho não se limita a possuir seus funcionários trabalhando em um escritório. Os consultores ficam dispersos pelos mais diversos locais e necessitam sempre enviar as informações que conseguem coletar para um sistema central da empresa. Quando mais rápida e precisa forem o processo das coletas, melhores serão os resultados de todos os envolvidos no processo.

1.2 OBJETIVO

1.2.1 Objetivos Gerais

A aplicação tema deste trabalho objetiva desenvolver uma solução para dispositivos móveis para a realização de auditorias em pontos de venda utilizando a tecnologia *Java Micro Edition*. Ocorrerá a integração do cliente móvel com um sistema servidor através de *Web Services* utilizando o formato JSON para comunicação de dados.

1.2.2 Objetivos Específicos

- Estudo da tecnologia JME;
- Integração com um software desenvolvido em outra tecnologia;
- Estudo do formato JSON com uma alternativa a Linguagem de Marcação Extensível (XML);
- Desenvolvimento de um aplicativo para controle de auditorias em pontos de venda que seja integrado com um software *web*.

1.3 JUSTIFICATIVA

Segundo o conceituado instituto de pesquisas em Tecnologia da Informação (TI) *Gartner*, o mercado de mobilidade dentro das empresas terá um grande crescimento até o final de 2011, onde o número de soluções móveis usadas dentro das organizações em todo o mundo avance a taxas de 30% ao ano.

Com o grande salto que as tecnologias móveis deram com o advento de *smartphones* e *tablets* cabe aos desenvolvedores estarem aptos a fornecer recursos de software de acordo com a demanda de equipamentos que são lançados no mercado. Mesmo não sendo os dispositivos mais modernos nos dias de hoje, os aparelhos celulares que suportam a tecnologia Java ainda são a grande maioria. E visando este mercado, se podemos dizer assim mais popular dentro do mundo dos aparelhos, visamos o desenvolvimento de uma solução móvel para força de vendas voltada para a plataforma *Java Micro Edition*, que apesar de não apresentar recursos tão sofisticados em relação ao *iPhone* ou aparelhos com o sistema

Android, para o aplicativo proposto tais recursos não farão diferença, acarretando em uma relação custo-benefício muito boa.

Com a computação de hoje voltada para o desenvolvimento de soluções que atuem na nuvem (*Cloud Computing*), nada melhor do que criar uma aplicação móvel que se integre com um sistema ERP (*Enterprise Resource Planning*) de uma empresa, por exemplo, e realize transmissão de dados entre os dispositivos.

Dentro da área de vendas os resultados positivos são mais do que fundamentais e por ser um mercado de grande oscilação, todas as decisões tomadas vão afetar o desempenho final para melhor ou pior. Quanto maior for a quantidade e qualidade de informações que os profissionais que gerem uma empresa possuírem, maiores serão os acertos possíveis em relação aos negócios e tendências do setor.

Pelo fato de os consultores de empresas de força de vendas estarem em constante movimento atrás dos clientes nas mais diversas localidades de uma cidade, estado ou país, a comunicação acaba ficando mais complicada entre as pontas do negócio. E justamente por isso, uma solução móvel iria potencializar e muito o trabalho de todos.

Integrando a aplicação móvel com um sistema gerencial que a empresa venha a possuir como *back-end*, as informações poderão estar ao acesso da gerência a todo o tempo possibilitando a extração dos mais diversos relatórios a respeito daquilo que se fizer necessário em determinado momento. Informações a respeito de como se encontram cada Ponto de Venda, cada promoção que esteja ocorrendo, dados de produtos dos clientes e também da concorrência. Sem falar na possibilidade de conseguir saber se cada funcionário está realizando suas atividades de maneira correta independente da distância que cada um esteja.

Referente aos funcionários operacionais do sistema (consultores, promotores, auditores, entre outros) os benefícios também são grandes. As coletas de dados serão realizadas de maneira muito mais prática e ágil devido ao tamanho dos equipamentos que serão carregados, os erros humanos, custos e necessidades de comunicação serão reduzidos, pois a entrada dos dados não será mais via papel, por exemplo. As ações nos Pontos de Venda ocorrerão de maneira muito mais interativa entre coletores e clientes, tornando o contato entre as partes mais agradável e fortalecendo os laços existentes.

Todos os benefícios apresentados serão gerados por um modelo computacional na área de mobilidade que visa auxiliar as empresas em gerar um grande retorno e melhores resultados às organizações do setor.

1.4 ESCOPO

O escopo da solução deste trabalho compreende o estudo da arquitetura JME, a utilização da tecnologia JSON para comunicação entre cliente e servidor e o uso de algumas Interfaces de Programação de Aplicações (APIs) opcionais do JME como *Mobile Media API* (MMAPI) e *Location*. O escopo da aplicação estará direcionando para a elaboração de um sistema cliente móvel. O servidor responsável por prover e ser abastecido com informações será desenvolvido com a tecnologia PHP.

O PHP está hoje entre as cinco linguagens de programação mais utilizadas do mundo, sendo voltada para a *web* (TIOBE, 2011). No mercado corporativo de hoje o PHP possui uma aceitação enorme devido a fatores como proporcionar um desenvolvimento rápido das soluções, uma curva de aprendizado curta e um valor final a ser desembolsado pelos empresários menor do que outras tecnologias. Este último fator, normalmente é o que acaba norteando as decisões da escolha da tecnologia para a elaboração de um *software*, principalmente em empresas de pequeno e médio porte. Pelo fato de encontrar muitos sistemas desenvolvidos com PHP no mercado e buscando uma simulação baseada no mundo corporativo, este trabalho visa integrar duas tecnologias diferentes (JME e PHP), utilizando ainda JSON para o transporte de dados entre plataformas ao invés do popular padrão XML.

As seguintes funcionalidades serão contempladas no desenvolvimento da aplicação:

- *Login*/*logout*: para que o usuário realize o acesso ao sistema ele deverá inserir os seus dados que foram cadastrados na plataforma *web*. De acordo com cada usuário autenticado no sistema é que as campanhas serão apresentadas no aparelho;
- Sincronismo de dados: de acordo com o usuário que estiver autenticado no sistema, o aplicativo listará as campanhas que estiverem vinculadas a um determinado promotor onde o mesmo terá a opção de realizar o *download* das campanhas que desejar. O sincronismo irá funcionar também no

processo de envio das auditorias realizadas para a plataforma *web*. Os questionários já finalizados ou ainda parciais poderão ser enviados pelo usuário ao sistema *web*;

- Campanhas: esta funcionalidade da aplicação listará todas as campanhas que foram “baixadas” da interface *web* por meio da funcionalidade de Baixar Campanhas. Ao selecionar uma determinada campanha, o usuário deverá executar a(s) auditoria(s) referente(s) aos itens que selecionou. As auditorias realizadas serão referentes a produtos;
- Armazenamento (persistência) de dados: com a realização do sincronismo de dados e da resolução das auditorias, a aplicação irá armazenar os dados localmente no dispositivo utilizando-se da API *Record Management System* (RMS). Este recurso permite o funcionamento da aplicação *off-line* caso não seja possível a conexão com a *internet* por algum motivo, desde que os dados necessários já estejam gravados no aparelho utilizado. Quando as auditorias estiverem finalizadas e forem enviadas ao servidor *web*, automaticamente ocorrerá a exclusão de determinados registros, uma vez que alguns aparelhos não possuem grande capacidade de armazenamento. Será possível também excluir manualmente auditorias desde que elas não tenham sido concluídas e enviadas (como dito anteriormente ocorre a exclusão automática nestes casos);
- Integração com câmera: esta função do aplicativo poderá ser utilizada apenas nos equipamentos que dispuserem de câmera fotográfica. Este recurso possibilitará aos usuários fotografarem determinados produtos durante sua auditoria. As fotos tiradas também serão enviadas para a plataforma *web* quando ocorrer um sincronismo de dados;
- Integração com Sistema de Posicionamento Global (GPS): assim como a integração com câmera, esta funcionalidade estará disponível apenas para alguns dispositivos que possuem as funcionalidades de GPS, será possível armazenar a localização onde uma determinada auditoria foi realizada, permitindo desta maneira que tanto os usuários quanto gestores saibam se a auditoria foi realizada no local certo.

Assim como delimitamos os limites de atuação da aplicação neste momento, também é necessário identificar aspectos que não serão abordados no projeto, tais como:

- Customização de leiaute do sistema *web*: não será desenvolvida uma interface customizada da aplicação *web* para que ela seja acessível através de navegadores dos dispositivos móveis;
- Portabilidade para outras tecnologias: a aplicação será desenvolvida apenas para funcionar em dispositivos que suportem a tecnologia Java;
- Conexão com a *internet*: para que a solução possa ser executada os dispositivos deverão possuir acesso a *internet*, seja por meio de redes *wi-fi* ou via pacote de dados. Como já foi descrito, a *internet* será fundamental para realizar o sincronismo de dados (envio e recebimento) do *software* móvel com o sistema *web* da empresa. Durante a realização das auditorias o acesso a *web* não se faz necessário com os dados sendo mantidos no próprio aparelho.

1.5 ORGANIZAÇÃO DO TRABALHO

O trabalho será dividido em sete capítulos que serão distribuídos da seguinte maneira:

- O segundo capítulo são abordadas teoricamente todas as tecnologias relevantes e utilizadas para a implementação do *software*, como informações sobre a tecnologia JME, algumas de suas APIs, a notação JSON e outras ferramentas para o desenvolvimento;
- O capítulo três é referente aos dados de análise e modelagem da solução proposta utilizando os conceitos da UML;
- O capítulo quatro apresenta um estudo de caso demonstrando uma experiência do uso do sistema desenvolvido;
- No quinto capítulo são descritas as conclusões obtidas com o trabalho, quais dificuldades foram encontradas, os pontos positivos e sugestões de trabalhos futuros que podem ser elaborados;
- Os dados do sexto capítulo são as referências utilizadas para a elaboração do trabalho;

- E o capítulo sete é onde se encontram os apêndices utilizados para elaboração do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Todas as tecnologias e ferramentas que foram fundamentais para o desenvolvimento deste trabalho serão descritas a seguir:

2.1 JAVA MICRO EDITION

Com o surgimento de uma grande gama de dispositivos móveis (telefones celulares, *pagers*, PDAs, entre outros) que apresentavam pequena capacidade de processamento e interfaces com poucos recursos gráficos em relação a computadores *desktop*, por exemplo, a Sun Microsystem lança em 1999 a plataforma Java 2 Micro Edition para os dispositivos móveis.

O JME consiste em um grupo de especificações que visa disponibilizar toda a tecnologia Java e as suas vantagens, como uma grande comunidade de desenvolvedores, uma grande empresa mantendo a tecnologia, a força de mercado que o Java possui e um forte sistema de segurança já muito experimentado.

Configurações, perfis e APIs são os três pilares que fundamentalmente compõe o JME. Basicamente esta divisão da plataforma ocorre devido à diversidade existente entre os aparelhos. Através da escolha de configurações, perfis e APIs diferentes podem ser construídos aplicativos que utilizam alguns recursos que determinados aparelhos possuem enquanto outros não.

As figuras a seguir ilustram a divisão da tecnologia Java e a estrutura do JME.

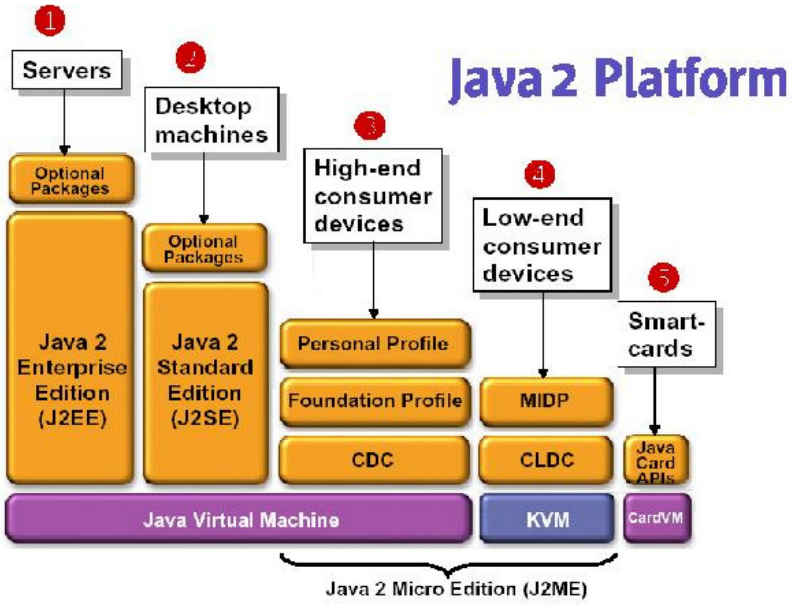


Figura 1 – A plataforma Java
Fonte: http://www.linux.ime.usp.br/~cef/mac499-03/monografias/murakami/monografia.html

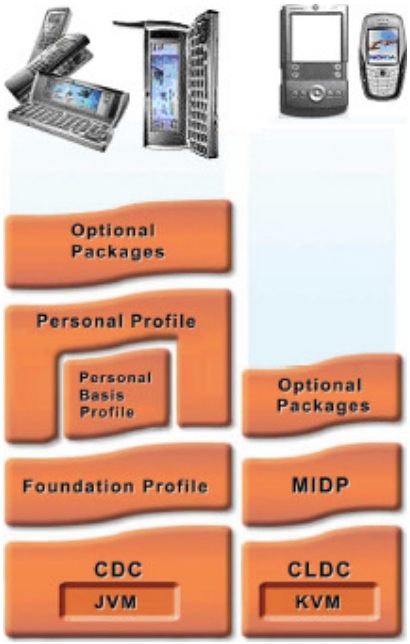


Figura 2 – A tecnologia JME e seus componentes
Fonte: http://www.devmedia.com.br/articles/viewcomp.asp?comp=120

A seguir serão abordados temas referentes à composição do JME, no que diz respeito às configurações, perfis e APIs.

2.1.1 Configurações

Uma configuração é o que define quais os mínimos recursos necessários para um determinado grupo de dispositivos com características parecidas. As configurações determinam qual é a máquina virtual (VM) Java e quais APIs estarão disponíveis para uso no desenvolvimento para os aparelhos que a utilizem.

Hoje estão definidas as configurações *Connected Device Configuration* (CDC) que engloba aparelhos de maior capacidade computacional (alguns PDAs, televisões e GPS, por exemplo) e a *Connected Limited Device Configuration* (CLDC) para equipamentos de menor capacidade, como é o caso de telefones celulares. Este trabalho irá tratar da configuração CLDC, uma vez que o *software* proposto é voltado para telefones celulares.

A máquina virtual que a CLDC possui é uma versão simplificada e muito menor do que a utilizada na edição *Java Server* Edition (JSE) e assim muitos recursos do Java e da VM do JSE não estão disponíveis na edição móvel, cujo nome é KVM (como pode ser visto na figura 3).

O uso restrito de *threads*, a não existência da finalização de objetos e nem reflexão e a pequena quantidade de classes de exceções para tratamento de erros são algumas das limitações imposta a CLDC. Isto sem falar que a versão 1.1 da CLDC trouxe funcionalidades que não havia na versão 1.0, como o suporte a cálculos com ponto flutuante.

2.1.2 Perfis

Mais conhecidos pelo termo em inglês *profiles*, os perfis atuam de maneira mais específica do que as configurações, complementando uma determinada configuração ao adicionar APIs para atender detalhes de cada dispositivo, trabalhando em mais alto nível definindo interface com o usuário, persistência de dados, conexão de dados e segurança.

Dentro do JME existem alguns perfis que englobam a CDC, como FP, PBP e PP e outros que se ligam à CLDC, principalmente o Perfil de Informações do Dispositivo Móvel, popularmente chamado de MIDP.

O MIDP é voltado para os dispositivos portáteis de baixo processamento e poucos recursos gráficos, possibilitando como dito anteriormente, que recursos de segurança, interface, persistência de dados e conexão de dados via *wireless* sejam desenvolvidos. Os fabricantes dos equipamentos podem ainda customizar o perfil a fim de oferecer componentes opcionais avançados para melhorar o desenvolvimento para tal dispositivo.

Um *Midlet* é o nome de um aplicativo em JME desenvolvido com o uso do perfil MIDP. O ciclo de vida de um *Midlet* passa por apenas três estados: pausado, ativo e destruído, como pode demonstrar a ilustração a seguir.

Figura 3 – Ciclo de vida de um *Midlet*
Fonte: http://imasters.com.br/artigo/3416/java/ciclo_de_vida_do_midlet/

Existem hoje três versões do MIDP, onde a 1.0 foi lançada no ano 2000 atendendo apenas aparelhos monocromáticos, a 2.0 chegou em 2002 trazendo suporte para imagens em 2D e 3D, acesso a *Bluetooth*, entre outros recursos modernos de multimídia e segurança. O MIDP 3.0 traz a melhoria em gráficos 3D, comunicação entre diversas *Midlets*, melhorias na persistência de dados e em todos os outros recursos que a versão 2.0 oferece.

2.1.3 APIs

Conhecidas também por pacotes opcionais, as APIs são componentes opcionais que fornecem novos recursos para uma configuração vinculada com um

perfil. Tais pacotes podem ser usados em dispositivos existentes ou novos que venham a surgir. Utilização de GPS, recursos multimídias, envio de mensagens, comunicação com *Bluetooth*, persistência de dados e imagens 3D são algumas das funcionalidades que podem ser utilizadas em um aplicativo com o uso de pacotes opcionais oferecidos a tecnologia.

A seguir serão descritos três pacotes opcionais que são utilizados dentro do JME: *Mobile Media* API, *Location* API e *Record Management System*.

2.1.3.1 Mobile Media API

Popularmente conhecida como MMAPI, esta API é um pacote opcional que permite o acesso e controle sobre recursos de multimídia dos aparelhos, possibilitando que as aplicações desenvolvidas sejam mais versáteis e contenham mais recursos, como a utilização de câmera de vídeo, controle do áudio do aparelho, arquivos de vídeo, entre outros recursos. É uma API que permite que funcionalidades não sejam implementadas caso os aparelhos não as suportem.

Descrita pela Requisição de Especificação Java (JSR) 135 (com a última versão datada em 2006), a MMAPI compreende três pacotes que são: *javax.microedition.media*, *javax.microedition.media.control*, *javax.microedition.media.protocol*, com cada um oferecendo acesso a diferentes recursos dos dispositivos.

Neste trabalho utilizaremos a câmera de vídeo de dispositivos que a possuem para demonstrar a utilização de um dos muitos recursos da MMAPI.

2.1.3.2 Location API

Definida na JSR 179 a *Location* API é uma especificação que oferece um pacote opcional para o desenvolvimento de soluções em Serviços Baseados em Localização (LBS – do original *Location Based* Services) em dispositivos que apresentam recursos limitados. Como requisito básico para seu funcionamento, deve ser utilizado em equipamentos que apresentem no mínimo a CLDC 1.1, devido

à necessidade de suporte de cálculos com pontos flutuantes e pode ser utilizada com diversos perfis, não dependendo de nenhum em específico.

O desenvolvimento de aplicativos baseados em serviços de localização, como o nome já indica, necessita de algum mecanismo que provenha tal informação. Estes dados podem ser oriundos de um GPS já existente no aparelho ou ainda de provedores de localização que podem ser tanto privados quanto em alguns casos gratuitos. Neste trabalho utilizaremos o GPS de alguns dispositivos para demonstrar a captura da localização do dispositivo utilizando um dos recursos da *Location* API.

2.1.3.3 Record Management System

O uso de RMS que permite que dados sejam armazenados internamente dentro dos dispositivos. É formado por registros de armazenamento que guardam os dados como um *array* de *bytes* e tem um identificador para cada registro a fim de permitir a recuperação das informações, chamado de *Record ID*. O *Midlet* é o responsável pela criação de um registro de armazenamento, desta maneira, quando ocorre a sua exclusão todos os dados também são apagados.

A API RMS oferece operações básicas de inserir, alterar, apagar, recuperar e listar os registros armazenados. Como já informado, em RMS os dados são armazenados como um *array* de *bytes* e por causa disso os dados enviados para gravação necessitam ser convertidos antes de realizar a operação.

Neste trabalho utilizaremos RMS para manter os dados baixados do servidor no aparelho e também para realizar a carga de dados que forem informados localmente pelos usuários.

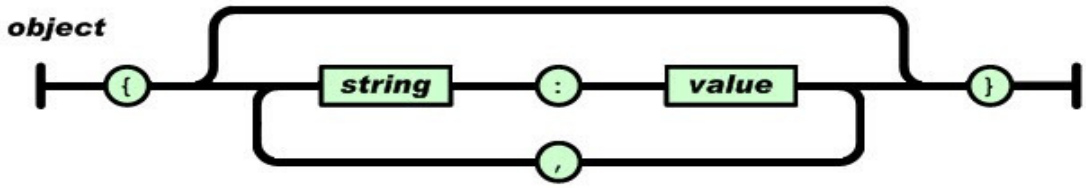
2.2 JSON

JavaScript Object Notation é a definição do acrônimo JSON que tanto tem circulado pela *web* nos últimos tempos. Criado por *Douglas Crockford*, a descrição do estabelecido pode ser encontrado na RFC 4627. Ele nada mais é do que um formato de dados leve que proporciona a transferência e serialização de

informações via rede, sendo uma grande alternativa ao padrão dominante no mercado hoje, o XML.

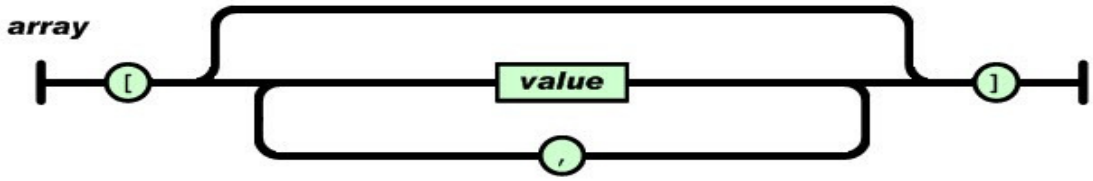
Como o próprio nome já diz, o JSON é baseado em um subconjunto de JavaScript, porém, pode trabalhar independente de sua linguagem "mãe" e possui hoje um grande suporte e muitas bibliotecas em todas as mais utilizadas linguagens de programação como Java, C, C++, C#, Ruby, PHP, Objective C, dentre outras.

A composição do JSON se dá pelos seguintes tipos de dados suportados: objeto, *array*, *string*, número ou valor. As figuras 1, 2 e 3 demonstram alguns exemplos de como podem ser organizados os dados com JSON.




```
{
    nome: "Exemplo",
    sobrenome: "Json",
    sexo: "M"
}
```

Figura 4 – Exemplo de uma estrutura de objetos em JSON
Fonte: http://www.json.org/




```
[1, 2, 3, 4, 5]
```

Figura 5 – Exemplo de uma estrutura de *Array* em JSON
Fonte: http://www.json.org/

Como pode ser visto nas figuras 1 e 2, a maneira como representar uma estrutura em JSON é bastante simples. Na figura 4 tem-se uma chave sucedida por um valor com cada par sendo separado por vírgula. O exemplo da figura 5 mostra um *array* de dados com os valores separados também por uma vírgula.

Em relação ao seu grande concorrente hoje (XML), o JSON apresenta inúmeras vantagens que podem ser apresentadas, dentre elas: maior facilidade de leitura e compreensão aos seres humanos uma vez que é muito mais intuitivo (como

visto nas figuras 1 e 2); melhor velocidade no processamento e transmissão dos dados, como pode ser visto na figura 3.

	JSON	XML
Number Of Objects	1000000	1000000
Total Time (ms)	78257.9	4546694.78
Average Time (ms)	0.08	4.55

Figura 6 – Comparação de tempo de transmissão entre JSON e XML
Fonte: Comparison of JSON and XML Data Interchange Formats: A Case Study

A figura 3 demonstra um estudo de caso realizado pelo Departamento de Ciências da Computação da Universidade de Bozeman, no estado de Montana nos Estados Unidos da América. Foram realizados vários testes comparativos entre os formatos concorrentes, e o ilustrado acima foi o primeiro deles. O objetivo era medir o tempo de transmissão de dados e para isso um *software* cliente fez feito o envio de um milhão de objetos codificados para o servidor tanto em XML quanto em JSON. Ao término do experimento o JSON obteve resultados de desempenho melhores do que o XML.

E comprovando o crescimento que o formato JSON vem obtendo, basta verificar que empresas e serviços como *Twitter, Foursquare* e *Yahoo!* já adotaram o "novo" padrão. Essas mudanças feitas por grandes empresas se devem em muito ao fato de que o transporte de dados serializados via JSON seja muito mais fácil de ser desenvolvido do que se fosse utilizado XML, reduzindo assim a complexidade das aplicações que o utilizam.

Portanto, o desenvolvimento deste trabalho irá trabalhar com JSON em detrimento ao XML para a realização da comunicação do *software* móvel (cliente) desenvolvido em JME com o servidor *web* construído em PHP.

2.3 UML

UML é a sigla de *Unified Modeling Language*, que segundo o OMG[1], "é uma linguagem visual para especificar, construir e documentar os artefatos de sistemas". Desenvolvida por Grady Booch, James Rumbaugh e Ivar Jacobson, sua utilização visa gerar esquemas com o uso de técnicas estabelecidas para que seja possível entender um determinado projeto de *software* antes do início da codificação. Comparando com outros ramos de atividade, uma modelagem realizada com UML representa o mesmo que plantas elaboradas para a construção cívil.

A padronização estabelecida pela UML visa que as modelagens orientadas a objetos possam ser entendidas de maneira simples independente do idioma das pessoas ou do tipo de projeto realizado, por exemplo, fazendo-se necessário apenas o conhecimento dos conceitos da linguagem.

Fazem-se presentes na UML inúmeros diagramas que norteiam a criação dos artefatos de modelagem de um projeto, indo desde as fases de análise de requisitos até a fase de testes. Dentre os principais diagramas existentes podemos citar: Casos de Uso, Classes, Atividades, Sequência e Estados.

Os Diagramas de Casos de Uso visam demonstrar quais serão os envolvidos em determinado processo, quais as funções cada um desempenhará e como será a associação de cada ator com as atividades. Um Diagramas de Classe é a base de como será realizada a codificação de um *software*, tende em vista que são elaborados quais serão as entidades de um sistema, seus atributos, associações e comportamentos (métodos). Os Diagramas de Atividades mostra a sequência em que as atividades são realizadas dentro de um fluxo de trabalho, apresentando decisões a serem tomadas e vários caminhos até o desfecho de cada atividade.

Desde 1997 quando foi aprovada como um padrão pelo OMG a UML ganhou uma série de ferramentas com o propósito de facilitar o desenvolvimento da modelagem, dentre elas *Rational Rose*, ArgoUML (*open source*), Jude, *Visual Paradigm*, entre muitos outros. O escolhido para a elaboração dos artefatos de modelagem neste trabalho foi o *Visual Paradigm.*

[1] É uma associação internacional sem fins lucrativos que define e mantém padrões na área de Orientação a Objetos.

Após esta breve explicação a respeito de algumas técnicas muito utilizadas em modelagens de sistemas utilizando-se de UML, seguem algumas ilustrações para que seja possível identificar alguns diagramas em sua forma visual.

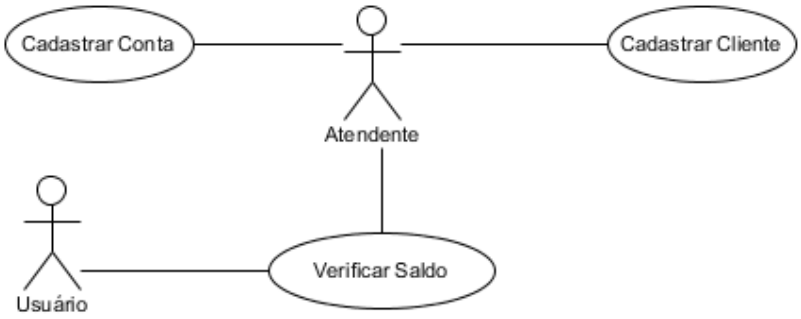


Figura 7 – Exemplo de um Diagrama de Casos de Uso

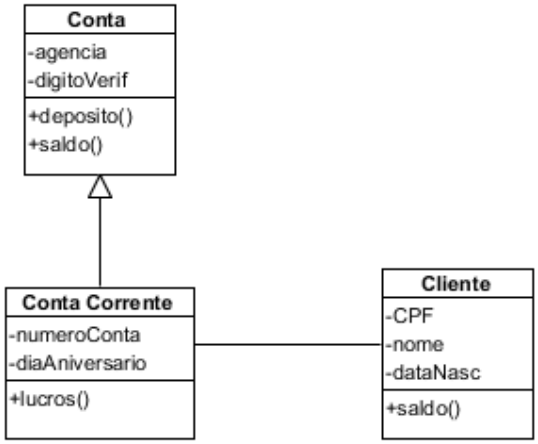


Figura 8 – Exemplo de um Diagrama de Classes

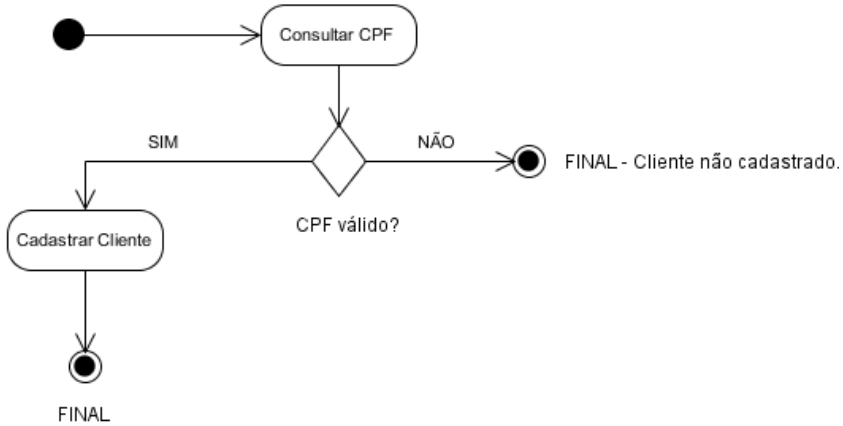


Figura 9 – Exemplo de um Diagrama de Atividades

2.4 NETBEANS

O NetBeans nada mais é do que uma IDE oriunda de um projeto de código aberto que possui a *Sun*, agora comprada pela *Oracle*, como sua principal mantenedora. Basicamente, a função da IDE é auxiliar os desenvolvedores para que possam escrever seus códigos de maneira mais organizada e com uma maior gama de recursos disponíveis no mesmo ambiente.

É possível que recursos como o *Wireless Tool Kit* (WTK) sejam utilizados dentro da IDE, proporcionando a simulação dos aplicativos desenvolvidos, a depuração de erros, escrita de documentação, entre diversas outras funcionalidades. E pelo fato de ser um *software* escrito totalmente em Java, pode ser utilizado em qualquer sistema operacional que suporte a máquina virtual Java (JVM).

2.5 GIT

Visando um melhor controle, organização e segurança do *software* desenvolvido, procura-se utilizar um sistema de versionamento de código, e é disso que o Git trata. Ele é um sistema de controle de versão distribuído de código aberto desenvolvido por Linus Torvalds[2]. O Git foi o escolhido por ser pequeno, rápido, fácil de aprender, além de existirem inúmeros serviços na *web* que oferecem repositórios Git *on-line*, como o famoso GitHub[3] ou o Assembla.

[2] Linux Torvalds é também o desenvolvedor do *Kernel* do Linux.
[3] Possibilitou a criação de uma grande comunidade do Git permitindo desenvolvedores pudessem interagir com diversos projetos, apresentando um grande objetivo social do serviço.

3 METODOLOGIA

3.1 METODOLOGIA

Para que o desenvolvimento do sistema móvel proposto seja possível, ele deve passar por uma análise visando explicitar tecnicamente todo o processo que se deseja informatizar e de que maneira ele deve ser codificado. Para isto, serão apresentados a seguir os seus requisitos, diagramas de casos de uso, classes e atividades e as especificações dos casos de uso, seguindo técnicas de análise da metodologia de processos *Rational Unified Process* (RUP) aliada a UML para a apresentação dos artefatos necessários para a elaboração do *software*.

3.2 MODELO DE PROCESSOS DE ENGENHARIA DE SOFTWARE

A documentação necessária para a elaboração do projeto está disponível na seção 4 deste documento e também nos documentos anexos *descricao_requisitos_mobile.pdf*, *plano_projeto.pdf*.

3.3 PLANO DE ATIVIDADE

Todas as informações necessárias a respeito do projeto podem ser encontradas nos documentos anexos *plano_projeto.pdf*, *TccPos.pod*.

3.4 PLANO DE RISCOS

O levantamento de riscos do projeto está presente no documento anexo *Lista de Riscos.pdf*. Pode ser visualizado qual é o risco, exposição, descrição, impacto e estratégia de mitigação.

3.5 RESPONSABILIDADES

A definição dos papéis e responsabilidades para a execução do projeto encontram-se nos documentos anexo *plano_projeto.pdf* na seção *4.3 Papéis e Responsabilidades* e *Termo de Abertura.pdf*.

3.6 MATERIAIS

As características do projeto podem ser acompanhadas na seção 4 deste documento e também nos documentos anexos *Declaração de Escopo.pdf*, *plano_projeto.pdf*, *plano_gerencia_configuracao.pdf* e *descricao_requisitos_mobile.pdf*.

4 ANÁLISE E MODELAGEM

Para que o desenvolvimento do sistema móvel proposto seja possível, ele deve passar por uma análise visando explicitar tecnicamente todo o processo que se deseja informatizar e de que maneira ele deve ser codificado. Para isto, serão apresentados a seguir os seus requisitos, diagramas de casos de uso, classes e atividades e as especificações dos casos de uso, seguindo técnicas de análise da metodologia de processos *Rational Unified Process* (RUP) aliada a UML para a apresentação dos artefatos necessários para a elaboração do *software*.

4.1 REQUISITOS

Tabela 1 – Requisitos do Sistema de Gestão de Pontos de Venda

Requisitos	Descrição
RF1 – Manter Login	O sistema deve realizar a autenticação dos usuários no sistema e manter os dados filtrados de acordo com quem está logado.
RF2 – Baixar Campanhas	O sistema deve permitir que as campanhas cadastradas pela empresa em seu sistema gerenciador sejam baixadas para o aparelho móvel para que os promotores possam coletar os dados necessários referentes a cada uma.
RF3 – Responder Auditorias	O sistema deve permitir que os promotores efetuem as auditorias referentes às campanhas armazenadas no dispositivo respondendo as perguntas cadastradas para cada uma.
RF4 – Enviar Auditorias	O sistema deve realizar o envio das informações coletadas pelos promotores em cada auditoria para o sistema gerenciador da empresa.
RF5 – Manter Localização	Para aparelhos celulares que possuam o recurso do GPS, o sistema deve armazenar as coordenadas onde a coleta dos dados de cada auditoria foi realizada.
RF6 – Tirar Fotos	Para aparelhos celulares que possuam câmera fotográfica, o sistema deve fornecer a possibilidade de capturar fotos referentes às auditorias realizadas, armazenando-as no dispositivo.
RNF1 – Criação do Layout	O layout do aplicativo móvel seguirá o tema utilizado no aparelho celular, adaptando-se aos dispositivos utilizados.

4.2 DIAGRAMA DE CASOS DE USO

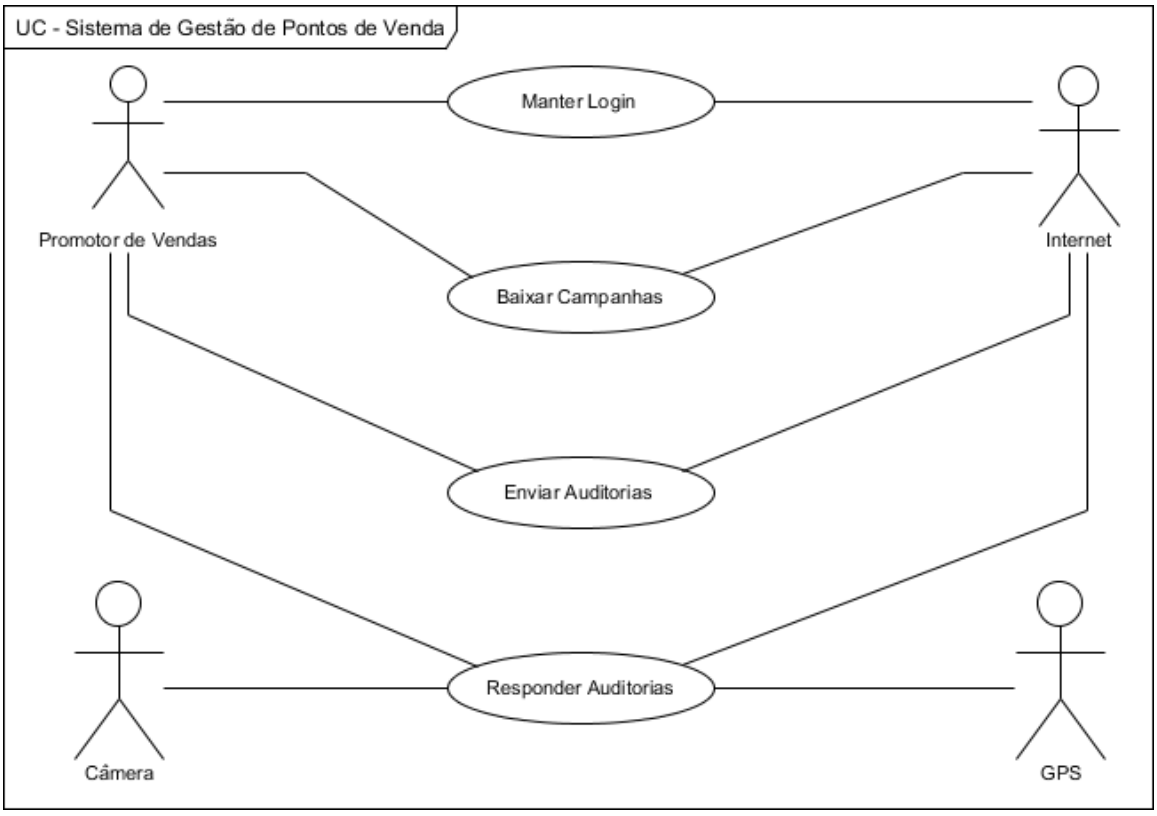


Figura 10 – Diagrama de Caso de Uso

4.3 ESPECIFICAÇÕES DE CASOS DE USO

Estão incluídas nos apêndices do documento as especificações de casos de uso desenvolvidas para a solução móvel proposta, apresentando a descrição dos fluxos do processo e a prototipação de telas que o *software* deverá ter. A tabela 2 relaciona os requisitos apresentados na seção 3.1 com os Casos de Uso visualizados na seção 3.2.

Tabela 2 – Relacionamento entre requisitos e Casos de Uso

Requisito	**Casos de Uso**
RF1 – Manter Login	Manter Login

Requisito	Casos de Uso
RF2 – Baixar Campanhas	Baixar Campanhas
RF3 – Responder Auditorias	Responder Auditorias
RF4 – Enviar Auditorias	Enviar Auditorias
RF5 – Manter Localização	Responder Auditorias
RF6 – Tirar Fotos	Responder Auditorias

4.4 DIAGRAMAS DE ATIVIDADES

4.4.1 Manter Login

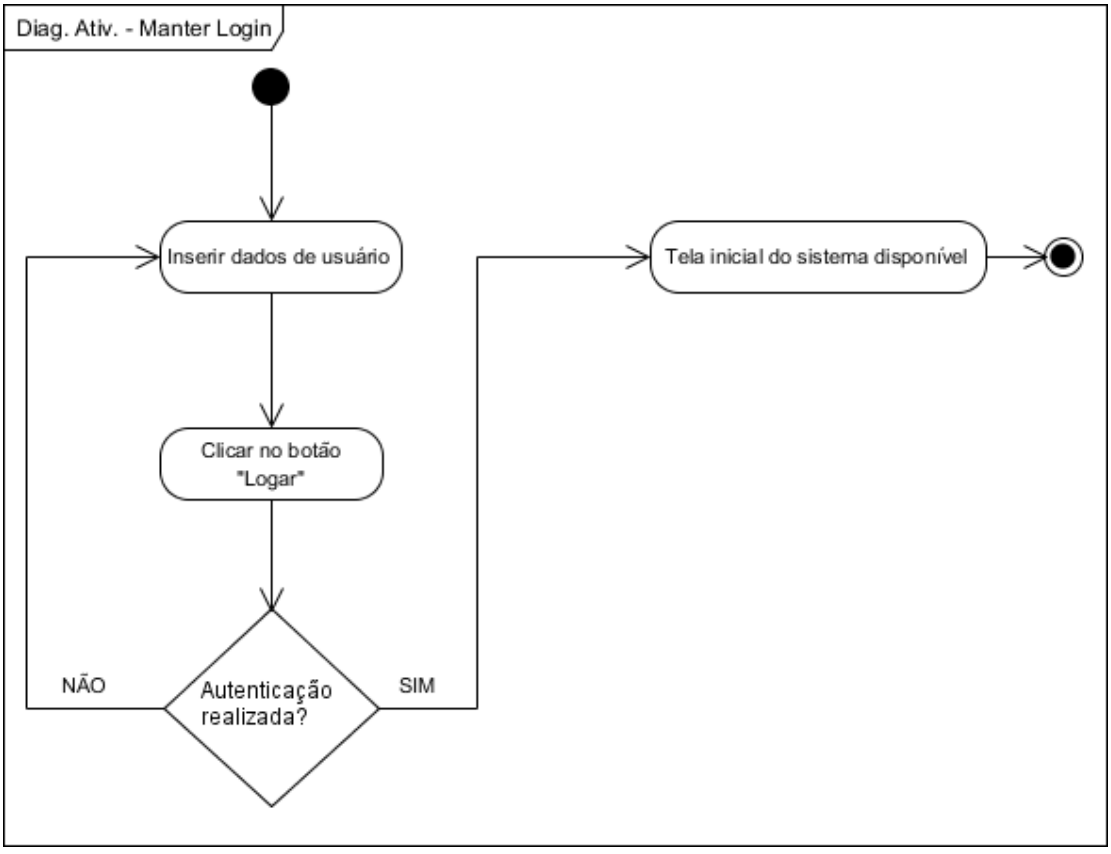


Figura 11 – Diagrama de Atividades (Manter Login)

4.4.2 Baixar Campanhas

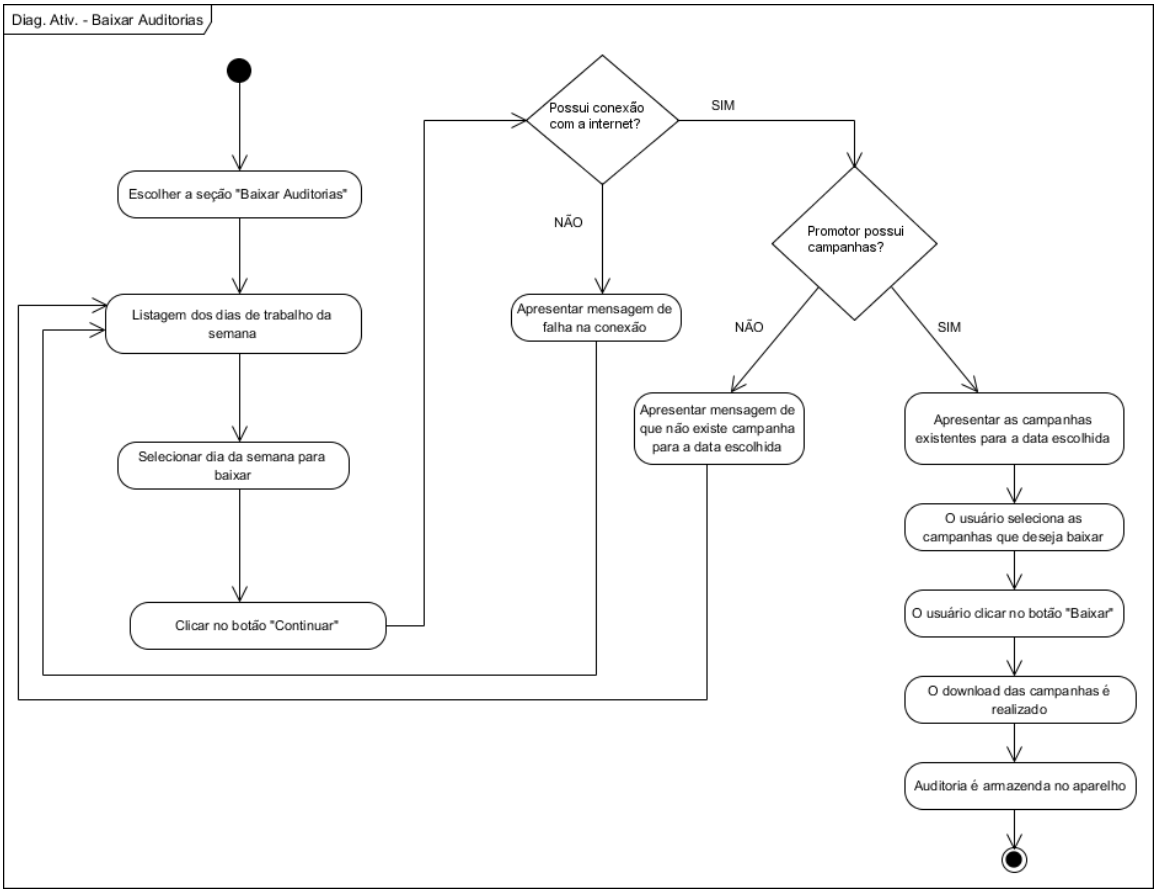


Figura 12 - Diagrama de Atividades (Baixar Campanhas)

4.4.3 Responder Auditorias

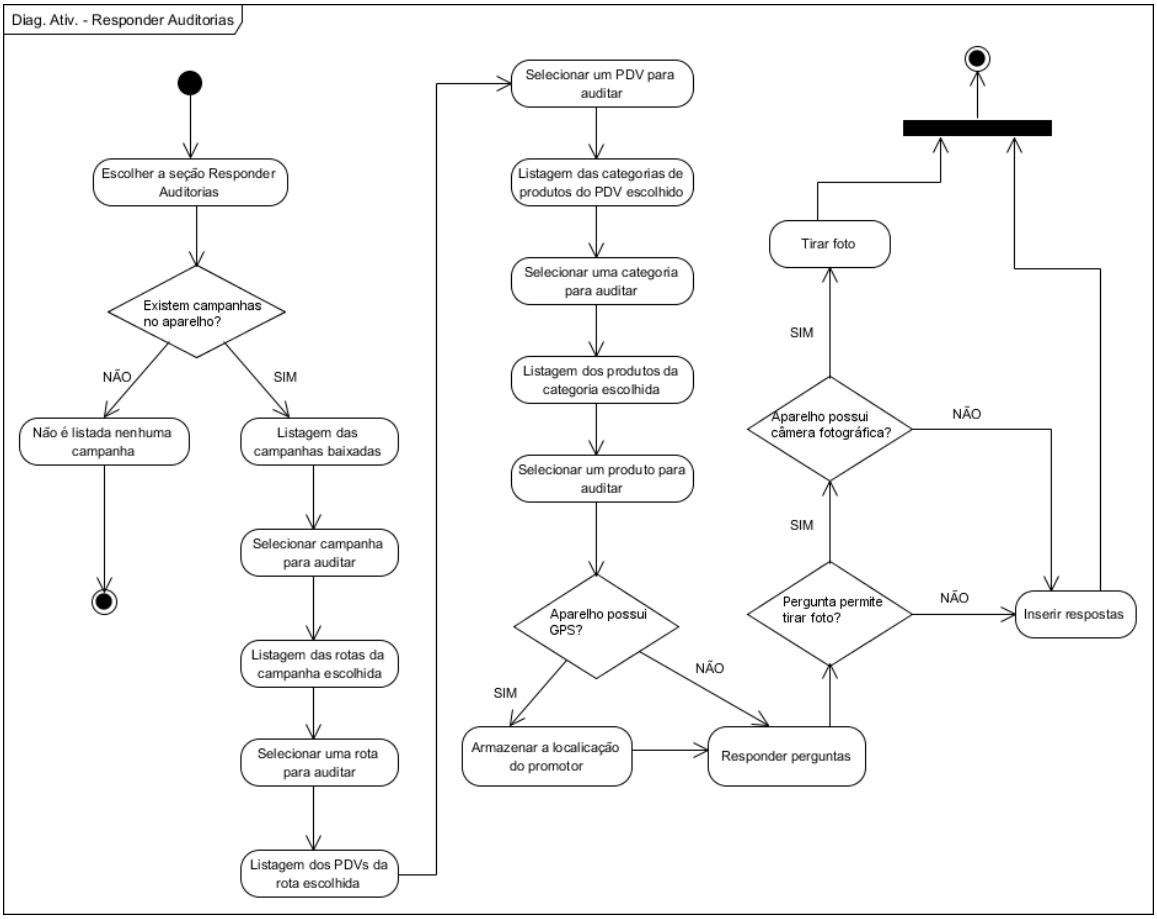


Figura 13 - Diagrama de Atividades (Responder Auditorias)

4.4.4 Enviar Auditorias

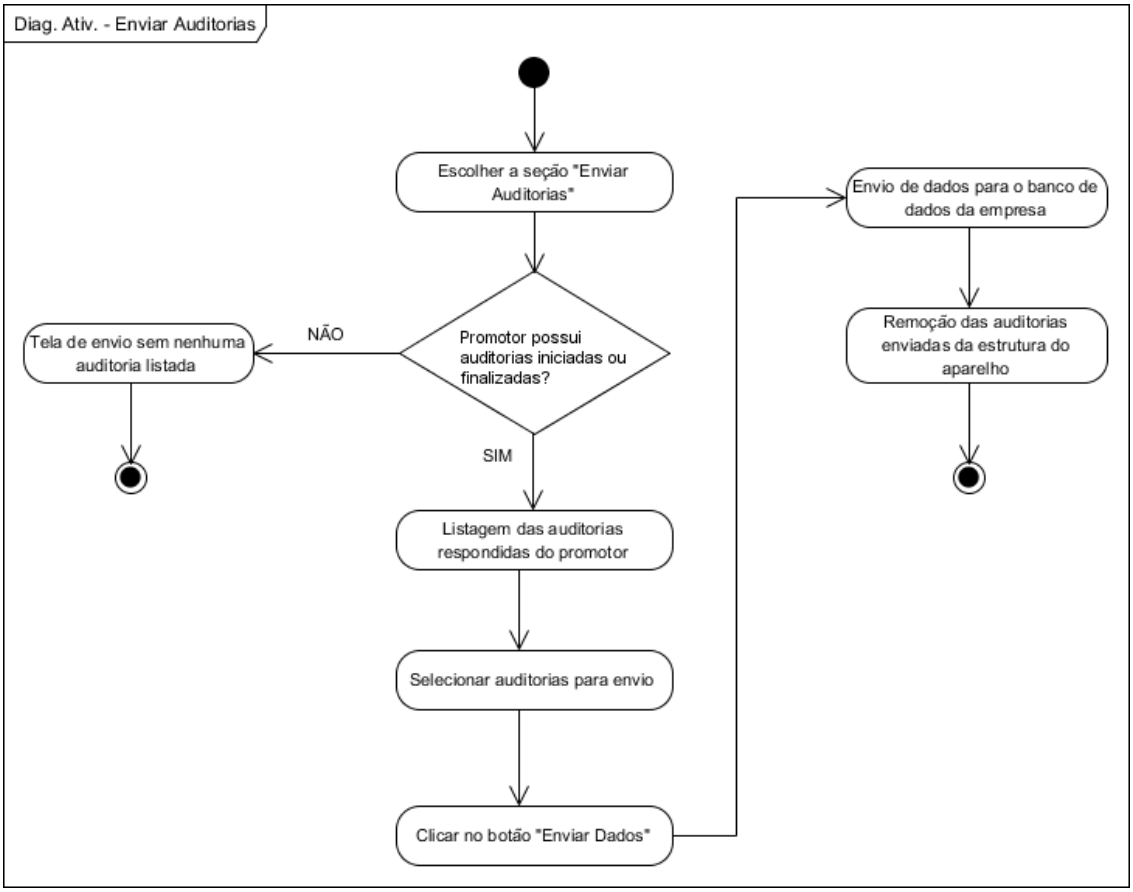


Figura 14 - Diagrama de Atividades (Enviar Auditorias)

4.5 DIAGRAMA DE CLASSES

O fluxo de funcionamento do aplicativo começa com a classe Application. Ela é a Midlet do aplicativo. Ela possui os métodos obrigatórios para que o aplicativo Java tenha o funcionamento adequado. Ao ser executado o método *startMidlet* um novo objeto da classe *MainController* é instanciado. Através deste objeto é feito as chamadas para os outros objetos apresentados através do diagrama de classe.

As classes Campanha, Pdv, Produto e Rota tem como principal objetivo armazenar os seguintes dados:

- Campanha que o usuário está vinculado;
- Pontos de Venda que um usuário deve auditar;
- Produtos que o usuário deve auditar;
- Quando, aonde e o que o usuário deve auditar.

As classes Questão e Alternativa tem como finalidade armazenar o formulário de forma dinâmica que deverá ser respondido de acordo com a campanha.

A classe resposta é responsável por armazenar as respostas provenientes da auditoria.

Abaixo segue a representação do diagrama de classes do projeto em dois níveis de apresentação diferentes. A diferença entre os dois modelos é a existência de atributos e métodos nas classes do diagrama da seção 5.5.2.

4.5.1 Nível 0

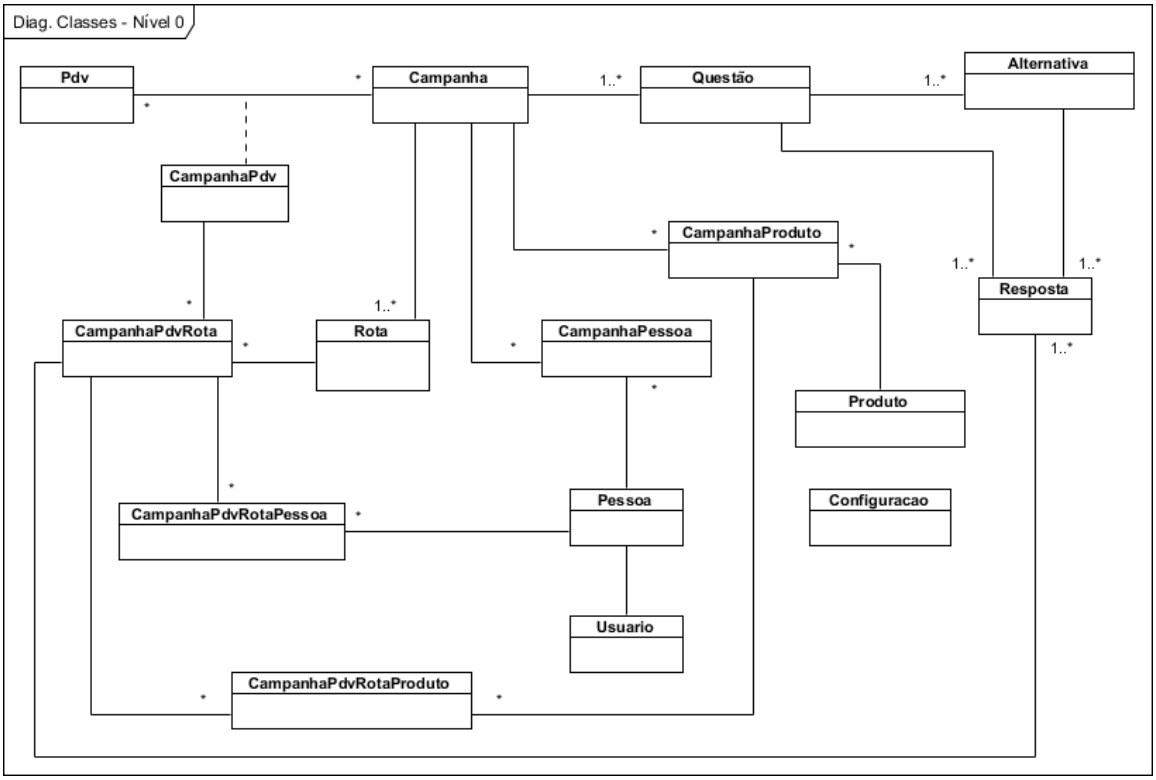


Figura 15 - Diagrama de Classes (Nível 0)

4.5.2 Nível 1

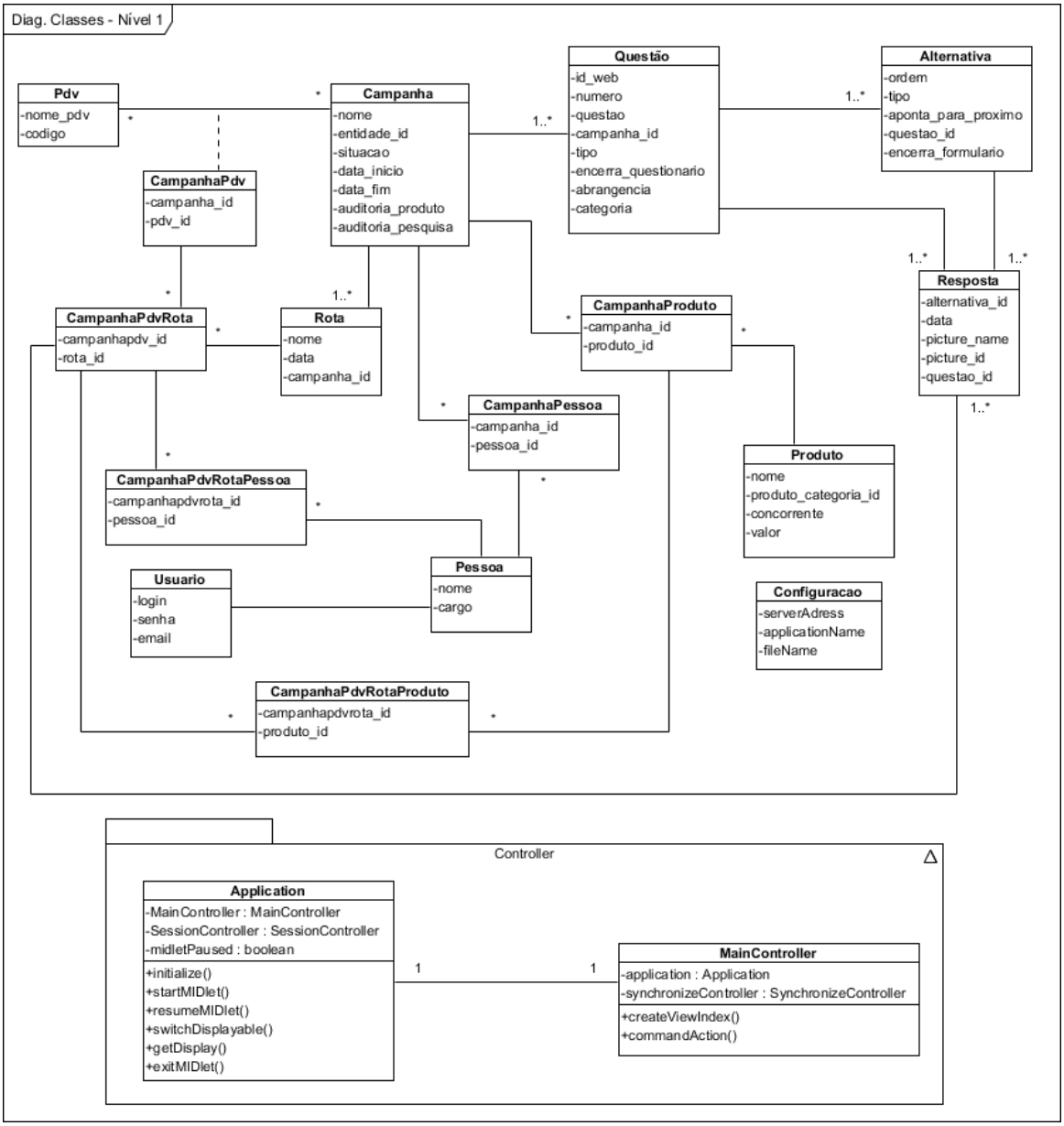


Figura 16 - Diagrama de Classes (Nível 1)

4.5 DIAGRAMAS DE SEQUÊNCIA

4.5.1 Manter Login

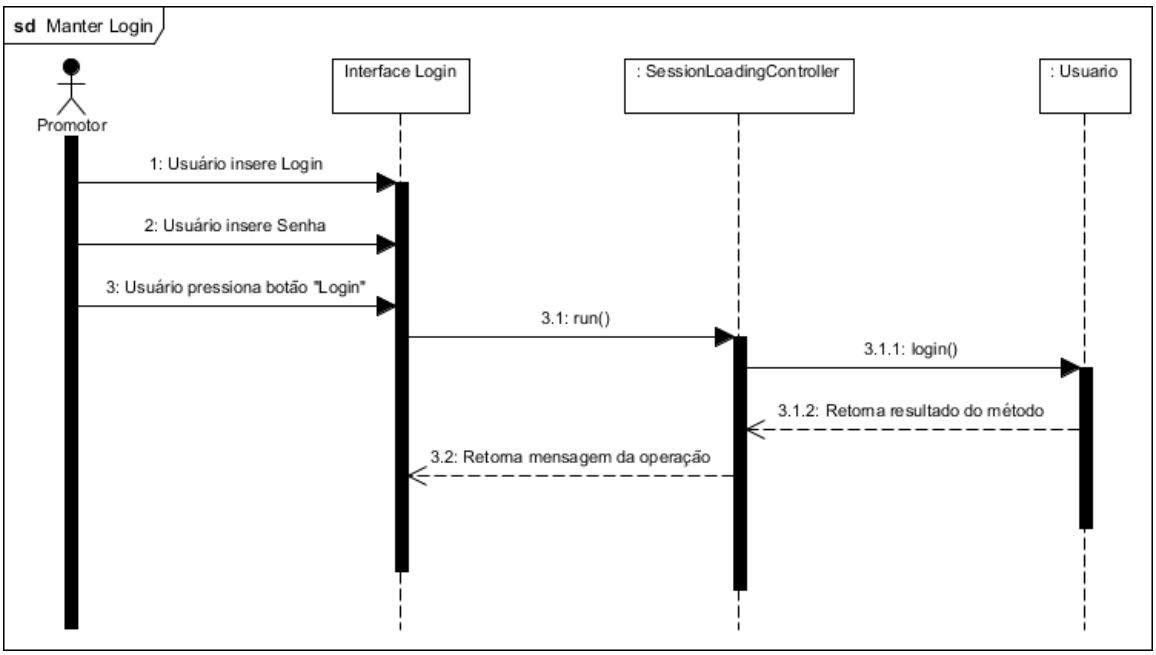


Figura 17- Manter Login

4.5.2 Baixar Campanhas

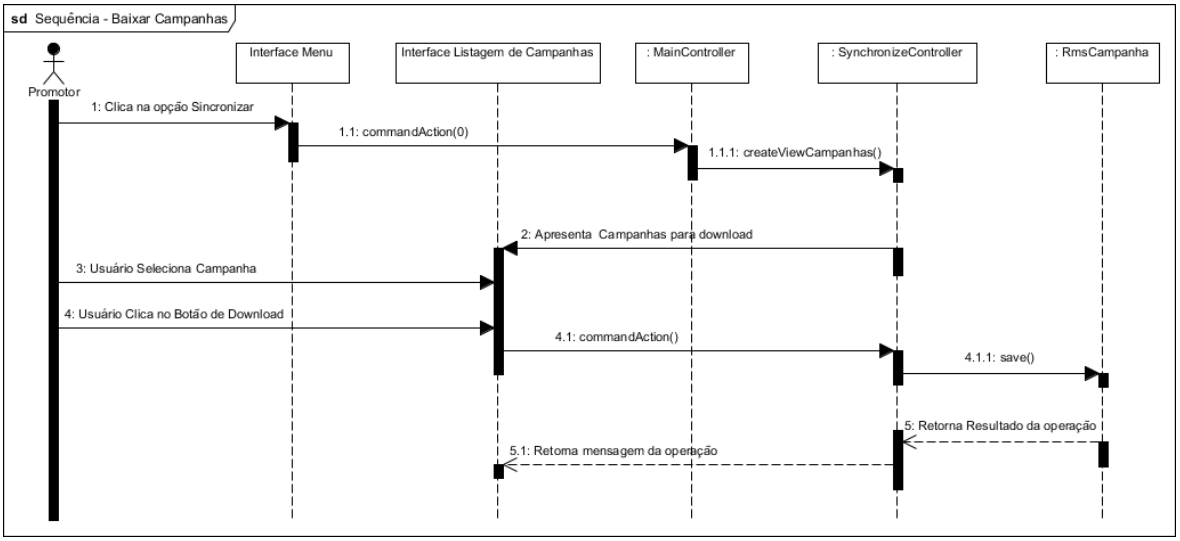


Figura 18 – Baixar Campanhas

5 CONSIDERAÇÕES FINAIS

O estudo desenvolvido neste trabalho abordou a utilização da tecnologia J2ME para o desenvolvimento de uma solução móvel voltada para dispositivos com configuração CLDC e perfil MIDP, capaz de se integrar com outras plataformas e recursos. O aplicativo desenvolvido visa automatizar o processo de coleta e envio de dados em pontos de venda, substituindo planilhas de papel utilizadas pelos promotores para realização dos trabalhos, garantindo ainda a mobilidade para a execução dos trabalhos.

Durante o desenvolvimento da solução móvel encontraram-se obstáculos para se trabalhar devido às limitações que a grande maioria dos aparelhos utilizados apresentam, como uma baixa capacidade de processamento/armazenamento, falta de banco de dados ou poucas opções gráficas, por exemplo. Apesar de tais restrições, o J2ME ofereceu todos os recursos para que o aplicativo cumprisse com as suas especificações definidas.

A integração do aplicativo móvel com um sistema *web* foi outro desafio encontrado e que pôde ser solucionado com a utilização do formato de dados JSON para transmissão e recepção das informações necessárias na hoje famosa “nuvem” de dados.

A lição que pode se tirar após este trabalho é que mesmo com a oferta de poucos recursos pelos aparelhos celulares trabalhados, o J2ME conseguiu atingir os objetivos propostos de forma eficiente, influenciando positivamente os trabalhos de todos os envolvidos, com o fornecimento de um aplicativo funcional em diversos dispositivos.

REFERÊNCIAS

CLARK, James. **XML vs the Web**. Disponível em: <http://blog.jclark.com/2010/11/xml-vs-web_24.html>. Acessado em: 11/04/2011.

CROCKFORD, Douglas. **Introducing JSON**. Disponível em: <http://www.json.org/>. Acessado em: 11/04/2011.

FERRARI, Pollyana (org). **Hipertexto hipermídia**: as novas ferramentas da comunicação digital. São Paulo: Contexto, 2007.

GIGUERE, Eric. **Databases and MIDP, Part 1**: Understanding the Record Management System. Disponível em: <http://developers.sun.com/mobility/midp/articles/databaserms/>. Acessado em: 13/04/2011.

GOMES, Lucas de Souza Reis. **Desenvolvimento de aplicações de dispositivos móveis com J2ME e integração com Web Services**. 2005. 138 f. Trabalho de conclusão de curso (Bacharelado em Sistemas de Informação), Universidade Federal de Santa Catarina, Florianópolis, 2005. Disponível em: <http://projetos.inf.ufsc.br/arquivos_projetos/projeto_207/lucas_gomes_monografia_artigo.pdf>. Acessado em: 13/04/2011.

GUEDES, Gilleanes T. A. **UML – Uma abordagem prática**. 3. ed. São Paulo: Novatec, 2008.

KNUDSEN, Jonathan. **Wireless Java**: Developing with J2ME. 2. ed. New York: Apress, 2003.

MAHMOUD, Qusay H. **J2ME and Location-Based Services**. Disponível em: <http://developers.sun.com/mobility/apis/articles/location/>. Acessado em: 14/04/2011.

MAHMOUD, Qusay H. **The J2ME Mobile Media API**. Disponível em: <http://developers.sun.com/mobility/midp/articles/mmapioverview/>. Acessado em: 14/04/2011.

MUCHOW, John W. **Core J2ME:** Tecnologia e MIDP. São Paulo: Makron Books, 2004.

OBJECT MANAGEMENT GROUP. **OMG Unified Modeling Language (OMG UML), Infrastructure**. Disponível em: <http://www.omg.org/spec/UML/2.4/Superstructure/Beta2/PDF>. Acessado em: 15/04/2011.

ORTIZ, C. Enrique. **Using JavaScript Object Notation (JSON) in Java ME for Data Interchange**. Disponível em: <http://java.sun.com/developer/technicalArticles/javame/json-me/>. Acessado em: 11/04/2011.

RODRIGUES, Felipe Paz. **Uma arquitetura para mobile marketing usando redes sociais**. 2010. 41 f. Trabalho de conclusão de curso (Bacharelado em Ciências da Computação), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010. Disponível em: <http://www.lume.ufrgs.br/handle/10183/26343>. Acessado em: 12/04/2011.

SUN MICROSYSTEMS. **J2ME Building Blocks for Mobile Devices**. Disponível em: <http://java.sun.com/products/cldc/wp/KVMwp.pdf>. Acessado em: 14/04/2011.

TELECO. **Estatísticas de Celulares no Brasil**. Disponível em: <http://www.teleco.com.br/ncel.asp>. Acessado em: 20/04/2011.

TIOBE. Disponível em: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. Acessado em: 03/05/2011.

APÊNDICES

APÊNDICE A – Especificações de Casos de Uso

UC – Manter Login

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Paulo Zeferino	13/05/2011	Composição
1.1	Paulo Zeferino	04/06/2011	Ajustes de revisão
1.2	Paulo Zeferino	05/06/2011	Protótipos de telas

Descrição
Este caso de uso tem como principal finalidade descrever o processo de *login* no sistema móvel de gestão de Pontos de Venda.

Pré-condições
Este caso de uso pode iniciar somente se:

1. O sistema estiver instalado no dispositivo móvel;
2. O aparelho em questão possuir acesso a *internet*.

Pós-condições
Após o fim normal deste caso de:
1. O ator Promotor de Vendas será redirecionado para a interface inicial do sistema, onde terá acesso a todas as outras funcionalidades do aplicativo.

Atores Primários
Promotor de Vendas
Conexão com a *Internet*

Fluxo de Eventos Principal
1. O sistema apresenta a tela **(DV1)**;
2. O ator Promotor de Vendas insere o seu *login* de acesso;
3. O ator Promotor de Vendas insere sua senha de acesso;
4. O ator Promotor de Vendas clica no botão “Logar”;
5. O sistema apresenta uma mensagem de êxito na realização do login **(R1) (DV2) (E1)**;
6. O sistema apresenta a interface inicial do sistema **(DV3)**;
7. O caso de uso é finalizado.

Fluxos de Exceção

E1. Dados Inválidos:
1. O sistema envia os dados da autenticação para um *WebService*, que consiste se as informações são válidas **(R1)**;

2. O sistema apresenta uma mensagem de que não foi possível realizar o *login* **(DV4)**;
3. O sistema permanece apresentando a tela de autenticação ao usuário;
4. O caso de uso é reiniciado.

Regras de Negócio

R1. O ator Promotor de Conexão com a *Internet* proporciona que um *WebService* receba as informações enviadas através do formulário de autenticação e verifica se *login* e senha existem na base de dados do sistema da empresa (que fica disponível *on-line*). Então, o mesmo *WebService* responde não ao aplicativo móvel se a autenticação foi possível ou através de um JSON.

Data Views

DV1 – Interface de *login* do sistema móvel

DV2 – Interface de *login* realizado com sucesso

DV3 – Interface inicial do sistema móvel

DV4 – Interface de *login* não realizado sem sucesso

UC – Baixar Auditorias

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Paulo Zeferino	13/05/2011	Composição
1.1	Paulo Zeferino	04/06/2011	Ajustes de revisão
1.2	Paulo Zeferino	05/06/2011	Protótipos de telas

Descrição
Este caso de uso tem como principal finalidade descrever o processo de *download* das auditorias que um promotor deve realizar nos Pontos de Venda.

Pré-condições
Este caso de uso pode iniciar somente se:

1. O usuário tiver realizado *login* com sucesso no aplicativo móvel;
2. O sistema tiver executado o UC – Manter Login;
3. O aparelho em questão possuir acesso a *internet*.

Pós-condições
Após o fim normal deste caso de uso:

1. O sistema estará carregado com todas as auditorias (cadastradas no sistema *web* da empresa) para um determinado usuário executar.

Atores Primários
Promotor de Vendas
Conexão com a *Internet*

Fluxo de Eventos Principal

1. O ator Promotor de Vendas acessa a opção de Baixar Campanhas na tela inicial **(DV3 – Manter Login)**;
2. O sistema apresenta a tela **(DV1)**;
3. O sistema apresenta os dias de trabalho da semana (sete dias a partir do atual);
4. O ator Promotor de Vendas seleciona qual dia da semana deseja baixar;
5. O ator Promotor de Vendas clica no botão “Continuar”;
6. Com o uso do ator Conexão com a *Internet*, o sistema verifica se existem campanhas para o usuário autenticado **(E1) (R1)**;
7. O sistema apresenta a tela **(DV3)**;
8. O usuário seleciona quais campanhas deseja baixar;
9. O usuário clica no botão “Baixar”;
10. É realizado o *download* dos itens selecionados e os mesmos são armazenados no aparelho celular **(DV4)**;
11. O sistema apresenta a interface **(DV1 – Responder Auditorias)**;
12. O caso de uso é finalizado.

Fluxos de Exceção

E1. Sem campanhas vinculadas ao usuário autenticado:

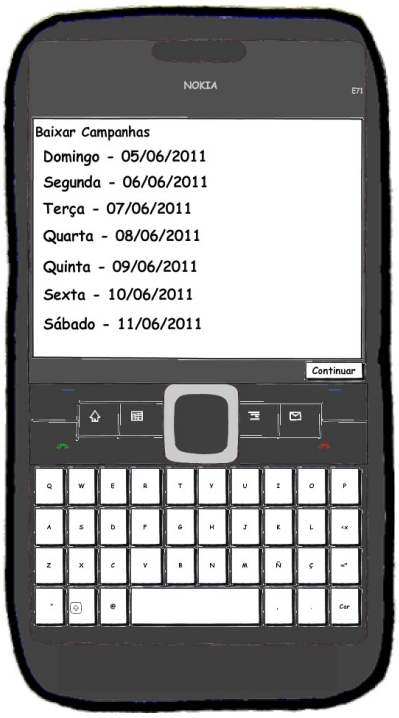
1. O sistema envia os dados do usuário autenticado para um *WebService*, que verifica se existem campanhas para o mesmo **(R1)**;
2. O sistema apresenta uma mensagem de que não existem campanhas para o usuário logado na data especificada **(DV2)**;
3. O sistema retorna a tela **(DV1)**;
4. O caso de uso é reiniciado.

Regras de Negócio

R1. Um *WebService* recebe uma solicitação para o *download* de campanhas e verifica quem é o usuário requerente. Então é feita uma consulta na base de dados da empresa (que fica disponível *on-line*) para saber se determinado usuário tem campanhas vinculadas a si para executar no dia escolhido, retornando o resultado da busca (positivo ou negativo) ao aplicativo móvel através de um JSON.

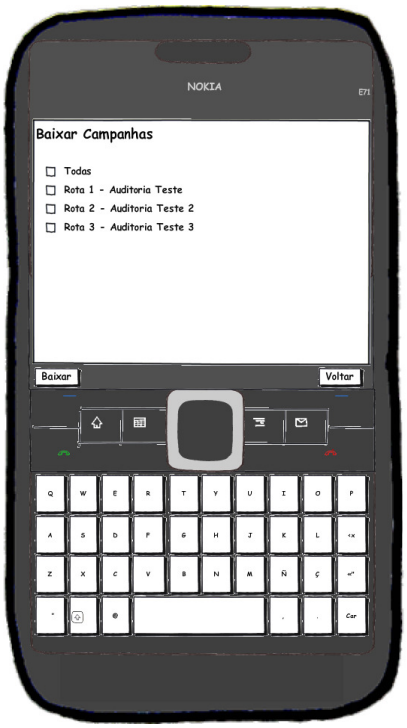
Data Views

DV1 – Interface de listagem dos dias de trabalho da semana



DV2 – Interface de informação sobre a não existência de campanhas para o usuário

DV3 – Interface para realização do *download* das campanhas



DV4 – Interface de informação sobre a realização do *download* com sucesso

UC – Responder Auditorias

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Paulo Zeferino	13/05/2011	Composição
1.1	Paulo Zeferino	04/05/2011	Ajustes de revisão
1.2	Paulo Zeferino	05/06/2011	Protótipos de telas

Descrição
Este caso de uso tem como principal finalidade descrever o processo de execução das auditorias (preenchimento das respostas) existentes no dispositivo móvel.

Pré-condições
Este caso de uso pode iniciar somente se:

1. O sistema possuir alguma auditoria baixada e armazenada no dispositivo móvel;
2. O sistema tiver executado o UC – Baixar Campanhas.

Pós-condições
Após o fim normal deste caso de:
1. O usuário terá respondido o questionário de alguma(s) auditoria(s) e as informações coletadas estarão armazenadas no dispositivo móvel.

Ator Primário
Promotor de Vendas

Fluxo de Eventos Principal
1. O sistema apresenta a tela **(DV1)**;
2. O ator Promotor de Vendas seleciona qual campanha deseja realizar;
3. O ator Promotor de Vendas clica no botão “Iniciar”;
4. O sistema apresenta uma relação das rotas existentes para a campanha escolhida **(DV2)**;
5. O ator Promotor de Vendas seleciona uma das rotas e clica no botão “Continuar”;
6. O sistema apresenta a relação de PDVs a serem auditados **(DV3)**;
7. O ator Promotor de Vendas seleciona um PDV e clica no botão “Continuar”;
8. O sistema apresenta a relação de categorias a serem auditadas **(DV4)**;
9. O ator Promotor de Vendas seleciona uma categoria e clica no botão “Continuar”;
10. O sistema apresenta os produtos da categoria escolhida que devem ser auditados **(DV5)**;
11. O ator Promotor de Vendas seleciona um produto e clica no botão “Continuar” **(A1)**;
12. O sistema apresenta as perguntas cadastradas para a auditoria **(DV7) (A2)**;
13. O ator Promotor de Vendas insere a resposta e clica no botão “Responder” **(E1)**;
14. Após o término de todo o questionário o sistema apresenta a interface **(DV10)**;
15. O sistema retorna a listagem de produtos **(DV11)**;

16. O caso de uso é finalizado.

Fluxos Alternativos

A1. Integração com GPS:
1. O aparelho celular possui o recurso de GPS **(DV6)**;
2. O sistema armazena o local onde está sendo feita a coleta de dados;
3. O caso de uso segue seu fluxo para o passo 12.

A2. Integração com câmera fotográfica:
1. O aparelho celular possui recurso de câmera fotográfica;
2. O sistema apresenta uma questão que possibilita a captura de uma imagem **(DV8)**;
3. O ator Promotor de Vendas tira a foto desejada;
4. O caso de uso segue seu fluxo para o passo 14.

Fluxos de Exceção

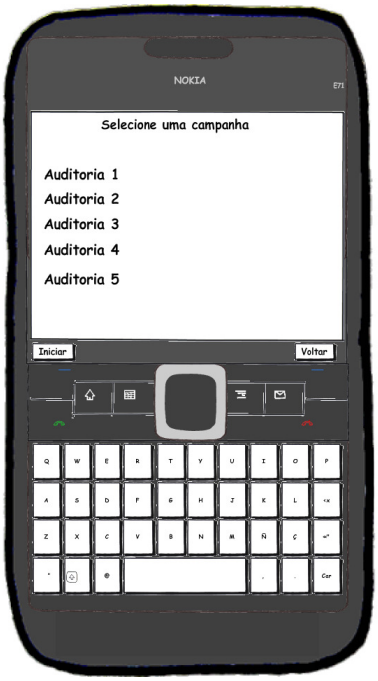
E1. Dados Inválidos:
1. O sistema verifica o campo de uma resposta **(R1)**;
2. O sistema apresenta uma mensagem de erro para o campo **(DV9)**;
3. O sistema permanece apresentando a tela da pergunta ao usuário;
4. O caso de uso é reiniciado.

Regras de Negócio

R1. O ator Promotor de Vendas deve definir uma resposta válida para as perguntas apresentadas nos questionários das auditorias. Não podem ser inseridos dados em branco.

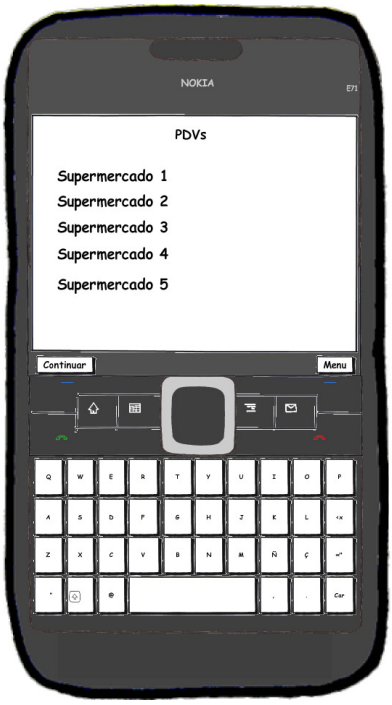
Data Views

DV1 – Interface de listagem das campanhas do usuário autenticado



DV2 – Interface de listagem das rotas existentes para a campanha escolhida

DV3 – Interface de listagem dos PDVs existentes para a rota escolhida

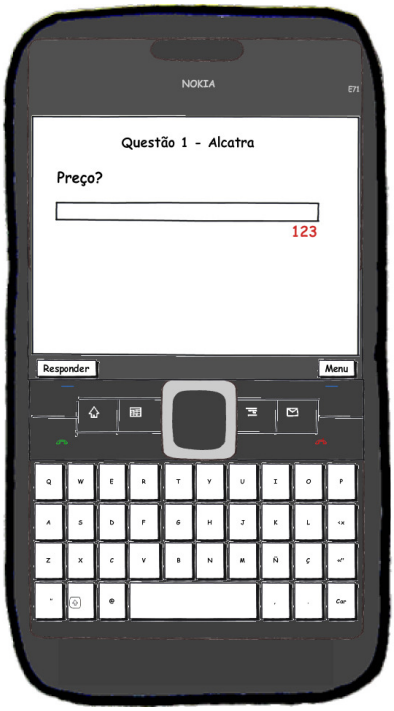


DV4 – Interface de listagem das categorias de produtos a serem auditados na rota escolhida

DV5 – Interface de listagem dos produtos a serem auditados na categoria escolhida

DV6 – Interface de obtenção da localização através do GPS

DV7 – Interface de questionário para o produto escolhido



DV8 – Interface de resposta com utilização da câmera fotográfica

DV9 – Mensagem de validação de uma resposta inserida incorretamente

DV10 – Mensagem de auditoria finalizada com sucesso

DV11 – Interface de listagem dos produtos com a apresentação da situação da auditoria

UC – Enviar Auditorias

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Paulo Zeferino	13/05/2011	Composição
1.1	Paulo Zeferino	04/06/2011	Ajustes de revisão
1.2	Paulo Zeferino	05/06/2011	Protótipos de telas

Descrição
Este caso de uso tem como principal finalidade descrever o processo de envio das informações coletadas nas auditorias ao sistema gerenciador da empresa.

Pré-condições
Este caso de uso pode iniciar somente se:

1. O sistema móvel possuir alguma auditoria já respondida e finalizada;
2. O sistema tiver executado o UC – Responder Auditorias;
3. O aparelho em questão possuir acesso a *internet*.

Pós-condições
Após o fim normal deste caso de uso:
1. O sistema removerá da estrutura de armazenamento do dispositivo móvel todas as informações que já tenham sido enviadas ao sistema *web* da empresa.

Ator Primário
Promotor de Vendas
Conexão com a *Internet*

Fluxo de Eventos Principal
1. O ator Promotor de Vendas acessa a opção de Enviar Auditorias na tela inicial **(DV3 – Manter Login)**;
2. O sistema apresenta a tela com a listagem de todos os pontos de venda já auditados pelo usuário logado **(DV1) (E1)**;
3. O ator Promotor de Vendas pode selecionar quais auditorias deseja enviar;
4. O ator Promotor de Vendas clica no botão “Enviar” **(DV3) (R2)**;
5. O sistema apresenta uma mensagem de êxito no envio das informações **(DV4) (R3)**;
6. O sistema apresenta a interface **(DV1)**;
7. O caso de uso é finalizado.

Fluxos de Exceção

E1. Sem auditorias realizadas:
1. O sistema verifica em sua estrutura que nenhuma auditoria possui status de iniciada ou finalizada **(R1)**;
2. O sistema apresenta a interface **(DV2)**;
3. O caso de uso é encerrado.

Regras de Negócio

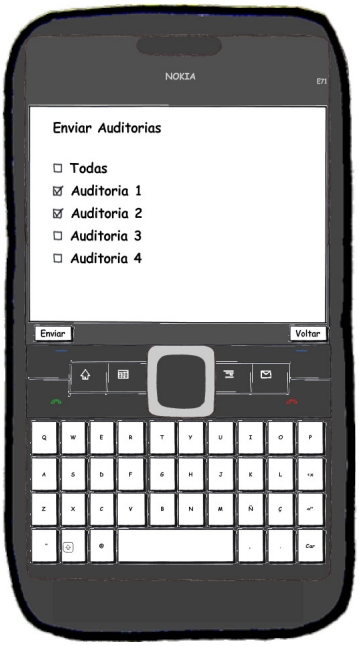
R1. O sistema verifica nas auditorias existentes em sua estrutura se alguma possui as situações iniciada ou realizada. Caso nenhum dos casos seja verdadeiro nenhuma auditoria é retornada para envio a base de dados da empresa;

R2. O ator Conexão com a *Internet* permite que um *WebService* receba as auditorias enviadas através do aplicativo móvel (no formato JSON) e armazena as informações na base de dados do sistema da empresa (que fica disponível *on-line*);

R3. O aplicativo móvel apaga de sua estrutura de armazenamento interna as auditorias que acabaram de ser enviadas ao sistema da empresa.

Data Views

DV1 – Interface de listagem das auditorias respondidas no dispositivo móvel



DV2 – Interface sem nenhuma auditoria para envio

DV3 – Interface de processamento do envio das auditorias

DV4 – Interface de envio das auditorias realizado com sucesso