

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDRÉ SCHWANTZ DE SIQUEIRA

**Atendente Móvel: solução em mobilidade
para atendimento personalizado e facilitado
a consumidores**

Trabalho de Graduação.

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, junho de 2010.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do CIC: Prof. João César Netto

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço a UFRGS e ao Instituto de Informática pelo ensino gratuito e de qualidade.

Agradeço aos colegas e principalmente aos diretores da Mobiltec, pelos ensinamentos e pela oportunidade de crescer profissionalmente.

Agradeço especialmente a minha família. Meus irmãos Rafael e Henrique, e meus pais, Luisa e José, que me apoiaram e incentivaram ao longo de todo o curso.

Obrigado a todos.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	8
RESUMO.....	9
1 INTRODUÇÃO	10
1.1 Organização do texto	10
2 FUNDAMENTOS.....	11
2.1 Computação e Dispositivos móveis.....	11
2.1.1 Windows Mobile	12
2.2 Sistema de Posicionamento Global	13
2.2.1 Dispositivos móveis com GPS e A-GPS	16
2.3 Código de Barras	16
3 ATENDENTE MÓVEL	19
3.1 Objetivos da Aplicação.....	19
3.2 Benefícios do uso da Aplicação.....	20
3.2.1 Situações de uso da Aplicação	21
4 MODELAGEM DO SISTEMA.....	23
4.1 Arquitetura do Sistema	23
4.2 Banco de Dados	23
4.2.1 Banco de dados do Legado.....	24
4.2.2 Banco de dados do Atendente Móvel	25
4.3 Aplicações	27
4.3.1 Atendente Client	27
4.3.2 Atendente Server	35
5 IMPLEMENTAÇÃO	43
5.1 Escopo do Protótipo	43
5.2 Ambiente de desenvolvimento	43
5.3 Apresentando o Protótipo	43
5.4 Bibliotecas utilizadas	47
5.4.1 Scanner de identificação do código de barras	47
5.4.2 Localização geográfica via GPS.....	48
5.5 Detalhes da Implementação.....	50
5.5.1 Interface do Web Services.....	50
5.5.2 Modelo do XML de resultado da consulta	51
5.5.3 Verificação das coordenadas do cliente.....	51
5.6 Testes.....	53
5.6.1 Testes de reconhecimento do código de barras	53
5.6.2 Testes com a coleta de coordenadas	53

5.6.3	Testes Gerais das aplicações Cliente e Servidor	54
6	CONCLUSÃO.....	55
6.1	Resultados.....	55
6.2	Implantação do sistema	55
6.3	Melhorias e novas funcionalidades	56
	REFERÊNCIAS.....	57

LISTA DE ABREVIATURAS E SIGLAS

EAN	European Article Number
UPC	Universal Product Code
MSDN	Microsoft Developer Network
BLL	Business Logic Layer
DAL	Data Access Layer
SGDB	Sistema Gestor de Base de Dados
NMEA	National Marine Electronics Association
UML	Unified Modeling Language
PDA	Personal Digital assistant
GPS	Global Positioning System
XML	Extensible Markup Language
API	Application Programming Interface

LISTA DE FIGURAS

Figura 2.1: Percentual dos sistemas operacionais em smartphone no ultimo ano (CANALYS, 2009).....	12
Figura 2.2: Constelação de satélites [Monico, 2000]	14
Figura 2.3: Método de triangulação dos satélites.	15
Figura 2.4: Exemplo de UPC.....	17
Figura 2.5: Exemplo de EAN 13	18
Figura 4.1: Arquitetura do sistema	23
Figura 4.2: Modelo de Dados do Legado	24
Figura 4.3: Modelo de dados do Atendente Móvel	26
Figura 4.4: Arquitetura do aplicação	28
Figura 4.5: Diagrama de classes do Cliente	30
Figura 4.6: Casos de uso do cliente	31
Figura 4.7: Diagrama de atividades da consulta de produtos.....	32
Figura 4.8: Diagrama de atividades da consulta de produtos	35
Figura 4.9 Diagrama de Caso de Uso do Atendente Server	36
Figura 4.10: Arquitetura do Atendente Server	37
Figura 4.11: Diagrama de Classes do Atendente Server	38
Figura 4.12: Diagrama de Sequência do Atendente Server.....	40
Figura 5.1: Tela inicial da aplicação.....	44
Figura 5.2: Tela de captura da imagem	44
Figura 5.3: Tela com o Atendimento Personalizado	45
Figura 5.4: Tela que exibe as duas imagens do produto.....	46
Figura 5.5: Tela com a exibição das coordenadas.....	46
Figura 5.6: Console de saída do servidor após a operação.....	47
Figura 5.7: Estrutura dos fontes da solução Zxing.	48
Figura 5.8: Plano de localização de uma área específica.	52
Figura 5.9: Exemplos de imagens não reconhecidas(escura e com reflexo, respectivamente).....	53

LISTA DE TABELAS

Tabela 2.1: Composição do código EAN/UPC	18
Tabela 4.1: Definição dos atributos do banco de dados do Legado	24
Tabela 4.2: Atributos dos atributos do banco de dados do Atendente Móvel	26

RESUMO

O Atendente Móvel é uma aplicação que pode ser utilizada pelos usuários de smartphone para obter informações sobre um determinado produto desejado, sem a necessidade de digitação, apenas direcionando a câmera do aparelho para o código de barras. Ao mesmo tempo, é determinado, através das coordenadas geográficas obtidas pelo GPS do dispositivo, se o cliente deverá receber algum tipo de atendimento diferenciado ou promoção por parte da loja.

Neste trabalho será apresentado o funcionamento detalhado da aplicação e seus respectivos objetivos. Será detalhado aspectos de sua modelagem e definindo como a mesma deverá ser implementada através de um projeto de suas classes e funcionalidades. O trabalho também apresenta conceitos técnicos importantes para o estudo como códigos de barras, localização por GPS e a importância dos dispositivos móveis.

Por fim, realizou-se um protótipo da aplicação, onde foram implementadas as principais funcionalidades do sistema. Desta forma foram validados os conceitos e a modelagem proposta.

1 INTRODUÇÃO

A popularização dos smartphones tem tornado os mesmos um dos mais promissores produtos da atualidade, criando uma grande oportunidade para novas criações em softwares. Este trabalho tem como objetivo apresentar uma solução em mobilidade para facilitar os consumidores a consultar produtos desejados, além de propor aos varejistas um novo canal de comunicação com o mercado.

O Atendente Móvel é uma aplicação que utiliza recursos que estão presentes na maioria dos smartphones lançados atualmente, como o receptor GPS e a câmera, para proporcionar de maneira inovadora uma forma de consumidores visualizarem informações de produtos na tela do dispositivo. Será possível efetuar esta operação sem a necessidade de digitação, apenas com o direcionamento da câmera do dispositivo.

Paralelamente à facilidade proporcionada ao cliente, a ferramenta proporciona ao varejista uma oportunidade de interagir com seus clientes, oferecendo vantagens e benefícios baseados na localização geográfica do consumidor.

1.1 Organização do texto

Nos próximos capítulos, este trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta os fundamentos que serão importantes para o entendimento do restante do trabalho;
- O Capítulo 3 apresenta os principais objetivos da aplicação que está sendo proposta, demonstrando situações de uso da mesma e quais os benefícios que os usuários terão com ela.
- O Capítulo 4 define a modelagem da aplicação, demonstrando como será sua arquitetura e funcionalidades, definindo o modelo de dados utilizado e o projeto das classes do sistema.
- O Capítulo 5 apresenta um protótipo da aplicação demonstrando o funcionamento de todas as telas do sistema e detalhando o modo como são implementadas as funcionalidades mais complexas do sistema.
- O Capítulo 6 apresenta os resultados obtidos ao longo deste trabalho e prevê um cenário de melhorias e ajustes a serem aplicados em futura implantação do mesmo.

2 FUNDAMENTOS

Neste capítulo serão apresentados fundamentos e conceitos importantes para a compreensão da aplicação que será apresentada neste trabalho.

2.1 Computação e Dispositivos móveis

Hoje no mundo existem aproximadamente 1,5 bilhão de televisores, aproximadamente 1 bilhão de pessoas têm acesso a internet e quase 3 bilhões de pessoas tem algum tipo de dispositivo móvel, tornando o mesmo um dos mais promissores produtos no mundo em termos de hardware e software. O avanço da tecnologia tem proporcionado reduções significativas no tamanho físico de componentes computacionais, como processadores, memórias e controladores de redes. Aliado ao avanço nas telecomunicações e redes de alta velocidade tem-se, hoje no mercado, dispositivos móveis com capacidade computacional suficiente para executar operações antes só imaginadas em computadores pessoais.

O termo “dispositivo móvel” abrange os conceitos de celular, PDA e smartphone, sendo o último o avanço e aprimoramento dos demais. Os PDAs (*Personal digital assistants*), também chamados de *handhelds*, foram os primeiros hardwares a apresentar o conceito de computador de “mão”, apresentando funcionalidades e sistema operacional que utilizavam conceitos dos PCs. Seguindo uma linha diferente, o celular nasceu com objetivo de facilitar as telecomunicações, porém foi agregando conceitos e funcionalidades, como memória e programas, que o aproximam dos computadores e PDAs.

Os smartphones são o dispositivo móvel que engloba os conceitos presentes nos celulares e PDAs, muitas vezes sendo confundindo-se com os mesmos. Apesar de os primeiros modelos terem surgido no final dos anos 90, os smartphones são o conceito mais atual de computação móvel, integrando softwares e internet.

Algumas das principais características presentes no smartphones mais atuais são:

- Armazenamento na ordem dos gigabytes;
- Receptores GPS;
- Câmeras digitais;
- Telas com recursos *touchscreen*;
- Acesso a redes wireless, Bluetooth, GPRS, 3G, etc;
- Funções de um telefone celular.

Nesta linha de dispositivos o que se destaca é a grande variedade de plataformas existentes no mercado. Apesar de existirem diferentes fabricantes e tecnologias, o que mais influi na aceitabilidade do mesmo diante do mercado é a plataforma utilizada. A falta de um padrão e de um sistema operacional único faz com que cada software tenha que ser codificado de diferente forma para cada um dos modelos existentes. Conseqüentemente, a quantidade e qualidade dos softwares desenvolvidos para um determinado modelo se tornam um diferencial na competição comercial.

Global Smartphone Sales, Q2 2009

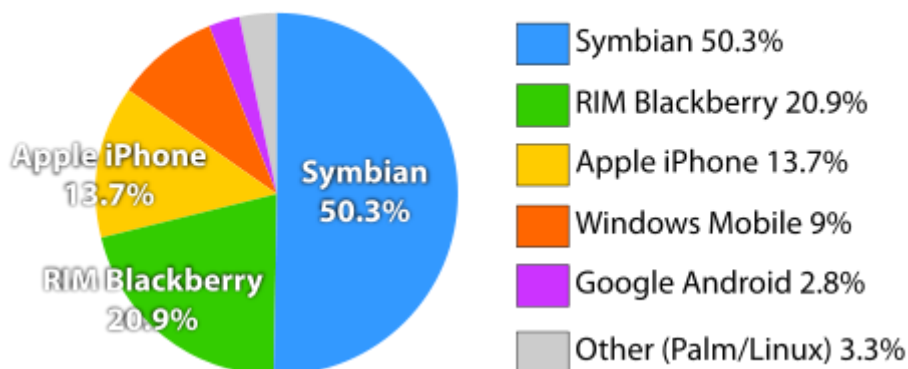


Figura 2.1: Percentual dos sistemas operacionais em smartphone no ultimo ano (CANALYS, 2009)

Um marco importante na história dos smartphones foi o surgimento, em 2007, do Iphone, dispositivo móvel fabricado pela Apple. Este aparelho apresentou novos conceitos ao mercado ocasionando um grande impacto comercial em usuários domésticos. Grande parte das vendas de smartphones, até alguns anos, era feita para empresas e os softwares oferecidos tinham funcionalidades voltadas para o ambiente corporativo. Com o surgimento do Iphone, o foco das vendas em smartphone ficou direcionado para os usuários finais e os fabricantes de software passaram a elaborar soluções voltadas ao entretenimento, utilizando recursos de mobilidade. Aproximadamente 50% dos Iphones comprados foram com o objetivo de substituir celulares convencionais. Neste período também foi observado um grande aumento no acesso a internet feito por dispositivos móveis, aumentando de aproximadamente 24% o valor das contas telefônicas nos Estados Unidos em 2008 em relação ao ano de 2007.

O sucesso do dispositivo da Apple provocou reações dos grandes concorrentes, como Microsoft e Google. Lançando as suas próprias plataformas, Windows Mobile 6 e Android respectivamente, estes fabricantes de software encontraram em parceiros como Samsung e HTC o suporte necessário para desenvolver o hardware.

2.1.1 Windows Mobile

O Windows Mobile é a plataforma para smartphones da Microsoft, que vem se esforçando para manter o sistema relevante frente aos concorrentes. Diferente do que temos nos PCs, onde o Windows possui quase 90% do mercado, o Windows Mobile tem uma participação relativamente pequena nos smartphones (inferior a 10%, conforme figura 2.1).

O histórico da companhia é bem extenso em termos de sistemas operacionais, apesar de sua pouca representatividade no mercado de smartphones. Os primeiros modelos baseados no Windows Mobile, criados no início da última década, eram divididos em

dois grupos: os Pocket PCs, que incluem telas sensíveis ao toque e podem ser operados da mesma forma que um PDA e os smartphones clássicos, que abandonam o uso da tela *touchscreen* em favor do uso de um direcional e botões de atalho. Em ambos os casos, o fabricante pode escolher entre usar apenas a tela *touchscreen*, incluir um teclado QWERT ou incluir um teclado numérico. A partir do Windows Mobile 6, a Microsoft adotou uma nova nomenclatura para as diferentes edições do Windows Mobile. A versão para smartphones sem tela *touchscreen* passou a se chamar Windows Mobile 6 Standard, enquanto a versão para *Pocket PCs*, com suporte a *touchscreen* passou a se chamar Windows Mobile 6 Professional. Completando a família, existe ainda o Windows Mobile 6 Classic, que é destinado a PDAs, sem funções de telefone, que hoje em dia são cada vez mais raros. Com o crescimento no volume de produção dos smartphones, a diferença de preço caiu, de forma que praticamente nenhum fabricante ainda se arrisca a lançar PDAs deste tipo. A mais nova tentativa da Microsoft na busca por aumentar sua participação no mercado de smartphones é o lançamento do Windows Phone 7. Assim será chamada a nova versão do Windows Mobile, que apresenta recursos gráficos melhorados e não mantém a compatibilidade com versões antigas (MICROSOFT, 2010).

O grande diferencial do sistema operacional da Microsoft para smartphones está no desenvolvimento do software. Provendo ferramentas como o Microsoft Visual Studio e frameworks como o .NET Compact Framework a empresa proporciona aos programadores um grande auxílio na criação de softwares. Estes ferramentais diminuem a curva de aprendizado dos desenvolvedores uma vez que utilizam os mesmos conceitos e linguagens do desenvolvimento para PCs e aumentam consideravelmente a produtividade, já que provem uma série de facilidades no acesso aos recursos nativos do hardware.

A facilidade no desenvolvimento de soluções é um forte atrativo para empresas que buscam a criação de soluções móveis próprias, devido à redução de custos em programação. Este é um dos fatores para que o Windows Mobile seja o segundo sistema operacional mais utilizado, com 23% do mercado corporativo (CANALYS, 2009) no período de 2009. O sistema só perde neste segmento para o BlackBerry, plataforma que tem forte tradição entre os executivos devido ao pioneirismo em serviços de e-mail oferecidos nos Estados Unidos.

2.2 Sistema de Posicionamento Global

O Sistema de Posicionamento Global, popularmente conhecido como GPS (*Global Positioning System*), é um sistema de cálculo de posicionamento e tempo baseado em sinais enviados para uma rede de satélites mantidos pelo Departamento de Defesa Norte Americano (*American Department of Defense*). Essa rede pode ser acessado por qualquer um que tenha o receptor correspondente. Os satélites orbitam o planeta em uma órbita geoestacionária, a 20.2 km de altitude, formando uma constelação de satélites, projetada de maneira que pelo menos 4 deles sejam visíveis a partir de qualquer ponto do planeta. Inicialmente a rede contava com 24 satélites, mas desde setembro de 2008 já conta com 27.

Esta tecnologia tem se tornado extremamente útil para uma série de atividades e estudos que necessitam de posicionamento como Cartografia, Meio Ambiente, Controle de frota de veículos, Navegação Marítima e Aérea, etc. Guardas florestais, trabalhos de prospecção e exploração de recursos naturais, geólogos, arqueólogos, bombeiros, são

enormemente beneficiados pela tecnologia do sistema. A popularidade dos receptores tem proporcionado a ciclistas, pescadores e ecoturistas uma nova experiência ao adotar o sistema em suas atividades. A comunidade científica também se utiliza desta tecnologia pelo seu relógio altamente preciso.

O princípio básico de navegação pelo GPS consiste na medida de distância entre o receptor do usuário e quatro satélites. Conhecendo as coordenadas dos satélites num sistema de referência apropriado, é possível calcular as coordenadas da antena do usuário no mesmo sistema de referência dos satélites. Do ponto de vista geométrico, apenas três distâncias, desde que não pertencentes ao mesmo plano, seriam suficientes. Neste caso o problema se reduziria a solução de um sistema de três equações, a três incógnitas. A quarta medida é necessária em razão do não sincronismo entre os relógios dos satélites e do usuário, adicionando uma incógnita ao problema.

O GPS consiste de três segmentos: espacial, controle e de usuários.

Segmento Espacial: Formado pelos satélites que estão em órbita, consiste de 24 satélites, em 6 órbitas diferentes com 4 satélites em cada órbita. Os satélites percorrem a órbita em torno da Terra a cada 12 horas, a uma altitude de aproximadamente 11.000 milhas náuticas. Cada satélite têm 28 graus de visualização sobre a Terra e estão inclinados 55 graus em relação à linha do Equador. Por alguns momentos, vários pontos da Terra são visualizados ao mesmo tempo por 6 a 10 satélites que focalizam a mesma área. Isto fornece redundância, pois apenas 4 satélites são necessários para uma determinação tridimensional de posição.

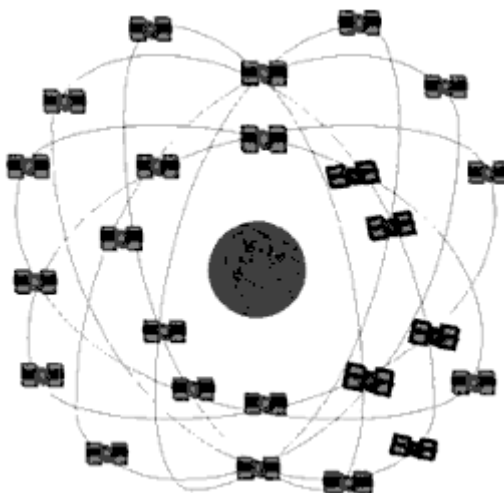


Figura 2.2: Constelação de satélites [Monico, 2000]

Segmento de Controle: Os satélites do segmento espacial são controlados em pelo segmento de controle na terra. Este controle é feito por uma estação de controle *Master* localizado no Colorado, nos Estados Unidos. Ela é responsável por monitorar o rastro dos satélites com o auxílio de cinco estações de monitoramento espalhados pela Terra, que processam todos os dados e então enviam a correção e sinais de controle para os satélites. O segmento de controle monitora o desempenho total do sistema, corrige posições do satélite e reprograma o sistema com o padrão necessário. A localização de cada estação de monitoramento oferece um monitoramento constante de cada satélite. Todas as estações de monitoramento trilham os satélites, determinam seu desempenho operacional, checam os parâmetros e passam estas informações para a estação *Master*. A estação *Master* pode então determinar os parâmetros de órbita de um satélite e transferir

dados de correção para o mesmo satélite. A determinação desse número de satélites circulando o globo, mais os planos de órbita dos satélites, junto com a estrutura de comando e controle, faz com que o GPS assegure que um número mínimo de quatro satélites sempre estará disponível para oferecer, seja de dia ou de noite, em qualquer lugar da superfície da Terra, uma posição precisa de determinado objeto (através de um receptor - um GPS Receiver). O mesmo aspecto de precisão espacial com o mínimo de recurso necessário, oferecendo uma informação segura, é assegurado na localização de cada estação de monitoramento. Também é assegurada a posição exata de cada satélite, supervisionando-o a todo o momento. Estes dois fatores são necessários para assegurar uma precisão tridimensional na determinação da posição - localização - geográfica.

Segmento dos usuários: este é constituído pelos receptores GPS, os quais devem ser apropriados para os propósitos a que se destina, tal como navegação ou outra atividade qualquer. Os receptores GPS convertem os sinais dos satélites em posição, velocidade, e tempo estimado. Quatro satélites, no mínimo, são necessários para calcular as quatro dimensões: x, y, z (posição) e t (tempo). Essa posição (x, y, z) pode ser determinada com os valores da distância de três posições diferentes conhecidas pelo método de triangulação.

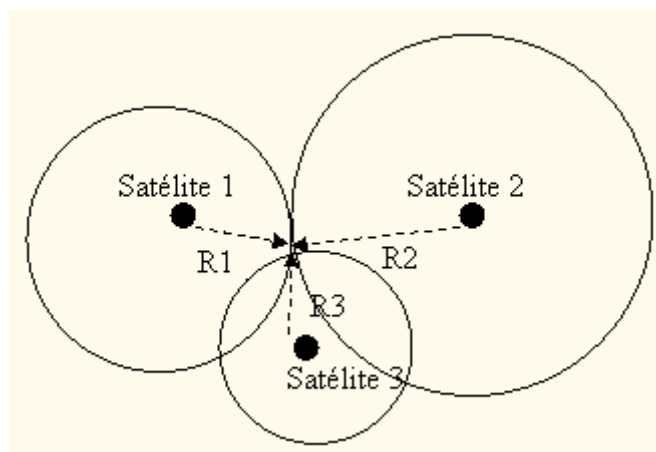


Figura 2.3: Método de triangulação dos satélites.

Conhecendo a distância em relação a três ou mais satélites, o receptor pode calcular a sua posição com base numa série de equações. Em teoria, a distância pode ser calculada multiplicando o tempo que o sinal demora a chegar pela velocidade a que este viaja (a velocidade da luz). No entanto, na prática são necessários cálculos mais sofisticados, uma vez que podem existir inúmeras interferências, como as condições atmosféricas ou as perturbações do Sol. A informação de três satélites é necessária para calcular a longitude e a latitude; no entanto, são necessários quatro para calcular também a altitude.

Durante muitos anos, o Departamento de Defesa Norte Americano reservou-se o direito de incluir um grau de erro aleatório na precisão do GPS. Este erro proposital causava uma imprecisão de aproximadamente 100m nos receptores dos usuários. O objetivo deste erro induzido era não fornecer aos inimigos militares uma importante ferramenta, já que é o GPS também é usado para o direcionamento de diversos tipos de armamentos de precisão, como as bombas JDAM (*Joint Direct Attack Munition*) e os famosos mísseis Tomahawk.

Ainda que atualmente não se aplique tal erro, a precisão do sistema GPS depende do número de satélites visíveis em um momento e posição determinados. Com um elevado

número de satélites sendo captados (7, 8 ou 9 satélites), e se estes têm uma geometria adequada (estão dispersos), pode se obter precisões inferiores a 2,5 metros em 95% do tempo.

Atualmente, o mercado disponibiliza receptores GPS em uma serie de utilitários que são utilizados para os mais diversos fins. Relógios, notebooks, navegadores automotivos e principalmente em dispositivos móveis.

2.2.1 Dispositivos móveis com GPS e A-GPS

Grande parte dos novos dispositivos móveis que são lançados atualmente possui um receptor GPS agregado ao seu hardware. Utilizados para os mais diversos fins, estes dispositivos estão ganhando uma grande variedade de softwares que fazem uso do poder da informação que localiza o usuário.

O fato da maioria dos dispositivos móveis ter acesso a uma rede de dados, fez com que os fabricantes dos mesmos desenvolvessem um aperfeiçoamento do receptor. Os receptores GPS convencionais, por diversos fatores, nem sempre conseguem se comunicar com os satélites em condições ideais. Por exemplo: quando um usuário está cercado de grandes edifícios os sinais sofrem com fenômeno *multipath*, fato que resulta em graves atrasos no processo de leitura. Pensando nisso, surgiram vários métodos com o objetivo de reduzir ao máximo esses atrasos. O A-GPS é o principal deles.

O A-GPS (*Assisted GPS*) é o processo onde o sinal dos satélites é combinado com o uso de triangulação celular e de um servidor remoto. O servidor funciona como um “assistente”, pois o mesmo é responsável por fazer o download completo da lista de coordenadas de todos os satélites presentes no espaço. Quando isto acontece, a operadora de telefonia rastreia a torre na qual o celular com A-GPS está conectado. Como a operadora conhece as coordenadas de cada torre, ela processa a lista (juntamente com outros dados) e identifica as coordenadas dos satélites que provavelmente estão visíveis para o celular requisitante. Após a filtragem, essas informações são enviadas ao aparelho por meio da rede de telefonia, como GPRS, EDGE, HSPA ou qualquer outra. Ou seja, o aparelho é poupado de diversas tarefas, afinal, recebendo uma informação já processada. Com isso, diminui-se o Tempo de Localização Inicial (*TTF* – *Time To First Fix*) de todo o sistema reduzindo o tempo que um aparelho leva para receber a sua localização geográfica.

Os dispositivos móveis fabricados por HTC, Nokia, Apple, entre outros e que possuem o recurso GPS, utilizam a tecnologia A-GPS.

2.3 Código de Barras

O código de barras é um modo gráfico de se representar números e caracteres através de um conjunto de barras paralelas de diferentes larguras que usualmente são lidos por um leitor. O princípio do mesmo é codificar informações, utilizando a tecnologia adequada de impressão, de forma que a mesma possa ser lida automaticamente por um equipamento leitura e comunicada a algum computador que interpreta o sistema binário correspondente.

A história do código de barras está associada com o desenvolvimento dos computadores no período pós-guerra. Entretanto o seu sucesso explica-se em grande

medida pela facilidade de identificação de produtos, documentos e boletos bancários, em diversas situações, como por exemplo: pontos de vendas, gestão de materiais, comunicação social (controle de tiragens), aviação (controle de carga e passageiros), controle governamental (receitas médicas), telecomunicações, entre outras.

Através da representação simbólica constituída por barras verticais escuras e claras, que formam uma estrutura binária, os códigos de barras podem ser classificados em dois tipos:

- Estrutura simples – constituído por módulos com larguras diferentes, sendo os de maior dimensão representando o *bit* “um”, os de menor dimensão ao *bit* “zero”.
- Estrutura complexa ou delta - constituído por módulos com a mesma largura onde os escuros representando o *bit* “um” e os de claros *bit* “zero”.

Considerando estes fundamentos vários tipos de códigos de barras foram propostos por diferentes instituições para diferentes meios. Segue alguns exemplos:

- UPC (*Universal Product Code*): códigos numéricos para identificação de itens comerciais em produtos de mercado americano. Administrado pelo UCC (*Uniform Code Council*). Conforme figura 2.4.
- EAN: tipos de códigos de barras numéricos para identificação de itens comerciais, regidos pelo órgão internacional GS1 (antiga EAN). Pode ser utilizado para produtos vendidos no varejo (tipo 13), utilizado em fardos e caixas de papelão (tipo 14) e utilizado em troca de dados entre parceiros comerciais (tipo 128). Conforme figura 2.5.
- Code 39: Desenvolvido porque algumas indústrias necessitavam codificar o alfabeto, assim como números, em um código de barras, sendo o Código 39 de longe a simbologia mais popular do código de barras nesta opção. É tipicamente o código de barras mais usado para identificação em estoques e de processos em diversos segmentos industriais. Todavia, o Código 39 produz códigos de barras relativamente longos e pode não ser adequado quando a largura da etiqueta for considerada.
- Code 128: Este código de barras provém da necessidade de uma seleção mais ampla de caracteres do que o Código 39 poderia fornecer. Quando a largura da etiqueta é considerada, o Código 128 é uma boa alternativa porque é muito compacto e resulta em um símbolo denso. Esta simbologia é frequentemente utilizada na indústria de transportes onde o tamanho da etiqueta é um problema.
- ISBN: código de barras utilizado para identificação de obras literárias. Utilizada largamente em bibliotecas.



Figura 2.4: Exemplo de UPC



Figura 2.5: Exemplo de EAN 13

Com base nos dois tipos de códigos de barras mais populares e importantes, o EAN e o UPC, surgiu o código universal EAN/UPC. Este código nada mais é do que uma padronização entre os códigos de produtos de consumo utilizados na Europa (definido pela GS1) e os códigos utilizados na América do Norte. Os leitores de EAN-13 podem ler também os códigos e UPC, e desde 2005 a América do Norte passou a utilizar os estes também com o intuito de se adequar aos padrões mundiais.

O sistema universal EAN/UPC (também conhecido como EAN.UCC, devido a sigla das duas instituições que controlam os códigos originais) foi criado em 1995. O mesmo foi desenvolvido para atender a uma das maiores necessidades que as empresas possuíam, que era a existência de um sistema de identificação que pudesse ser utilizado em todos os setores industriais e varejistas, inclusive mundialmente, e que também garantissem a identificação exclusiva e sem ambiguidades. Com isso, o sistema foi criado para que fabricantes, exportadores, importadores, hospitais, atacadistas, varejistas, etc. pudessem usá-lo para comunicar informações relativas às mercadorias e aos serviços que comercializam.

O padrão universal é controlado por instituições regionalizadas que em cada país garantem a liberação de códigos não ambíguos para os novos registros criados. Essas instituições vendem para as empresas novos registros. No Brasil o Sistema Nacional de Codificação de Produtos, também conhecido como Código de Barras EAN foi oficializado pelo governo para todos os bens de consumo fabricados no país, que devem ter seu respectivo Código Nacional de Produto, indispensável no processo de padronização e informatização de estabelecimentos comerciais, e também nas transações entre a indústria e o comércio. O Código Nacional de Produtos segue o padrão EAN (*International Article Numbering Association*), entidade de âmbito internacional, com sede em Bruxelas, na Bélgica, à qual o Brasil está associado.

O EAN/UPC é composto conforme a tabela:

Tabela 2.1: Composição do código EAN/UPC

<i>Dígitos</i>	<i>Função</i>	<i>Responsabilidade</i>
3	Código que identifica o país.	EAN
5 a 4	Código que identifica a empresa.	EAN
4 a 5	Código que identifica o produto.	Empresa
1	Dígito de controle.	Obtido através de algoritmo.

3 ATENDENTE MÓVEL

Neste capítulo é apresentado o principal objetivo deste trabalho: a definição da aplicação Atendente Móvel, ferramenta para atendimento personalizado de clientes com base em sua localização geográfica, facilitada pela identificação via código de barras de produtos.

3.1 Objetivos da Aplicação

Com o constante surgimento de novos modelos de smartphones, celulares sofisticados e PDAs, a popularização dos mesmos pode ser considerada como uma consequência dos esforços de seus fabricantes em campanhas de marketing e no constante investimento em pesquisa de novos recursos. Associados com a ampliação das redes de acesso a dados de alta velocidade, oferecidas pelas operadoras de telefonia móvel, estes dispositivos já estão inseridos no alvo de empresas como um novo modo de oferecer serviços e publicidade a seus clientes.

Este trabalho tem como objetivo propor um sistema que possa ser uma alternativa para grandes lojas varejistas interessadas em utilizar uma ferramenta própria para atingir este novo mercado, de modo que possa aumentar suas vendas e abrir um novo canal de informação e publicidade com os seus clientes.

O aplicativo que será definido ao longo deste capítulo tem como meta fornecer aos usuários de dispositivos móveis (celulares, smartphones e similares) um sistema que facilite a compra de produtos variados fornecendo informações e atendimento personalizado, sem requerer do usuário a digitação do produto desejado nem a utilização de browsers Web. Por outro lado a ferramenta dará ao varejista, que ofertarem a aplicação para os seus clientes, a oportunidade de oferecer um atendimento diferenciado baseado na localização geográfica dos mesmos.

O usuário inicia o uso da aplicação da seguinte forma: ao encontrar o produto desejado, a ferramenta deverá ser inicializada e o cliente poderá fazer identificação do produto com a câmera do dispositivo. Através da imagem capturada será iniciado um processo de reconhecimento do código de barras através de um determinado algoritmo que efetua um scaneamento da imagem, identificando o item através do seu código EAN/UPC. Paralelamente à identificação do produto é feita a localização geográfica do usuário através das coordenadas fornecidas pelo GPS do dispositivo.

Posteriormente uma chamada consulta é feita ao servidor do sistema, fornecido pelo proprietário do mesmo (no caso a empresa), utilizando as informações coletadas (código do produto e localização geográfica) além da identificação do usuário. Será buscada no banco de dados as informações do produto desejado, como preço, dimensões, detalhes técnicos, etc., e as mesmas serão disponibilizadas para o cliente. Também será

verificado na base de dados se para a localização geográfica do cliente existe algum tipo de atendimento personalizado cadastrado. Atendimento personalizado é como será denominado neste texto as mensagens que poderão ser enviadas para o cliente, propondo alguma promoção ou informação publicitária.

Atendimento personalizado, também poderá ser feito com mensagens enviadas ao operador do sistema, informando que o cliente está na loja e deverá ser atendido particularmente.

O desenvolvimento desta aplicação pode ser considerado tecnologicamente inovador, por envolver questões atuais como mobilidade e localização geográfica, e ao mesmo tempo desafiadora do ponto de vista de implementação uma vez que utiliza recursos usualmente vistos em aplicativos para desktop, como processamento de imagens (leitura do código de barras) e também acesso a Webservice a partir de um dispositivo móvel.

3.2 Benefícios do uso da Aplicação

A aplicação que está sendo proposta engloba uma serie de facilidades que são benéficas para o usuário da mesma e são um importante artefato para varejistas atraírem novos clientes para a sua loja e aumentando a possibilidade de efetuar boas vendas.

O cliente usuário da aplicação será beneficiado pelas seguintes facilidades:

- Obter informações do produto em qualquer lugar: utilizando o seu dispositivo móvel compatível e com um acesso a internet, o usuário poderá obter informações sobre um produto de seu interesse, como preço, especificação, condições de pagamento, etc., em qualquer lugar de forma rápida e personalizada. O usuário também poderá obter amostras digitais dos produtos de seu interesse, como trechos de uma música ou trailer de um filme.
- Fácil identificação do produto: sem a necessidade de digitar a característica do produto ou o seu código, o usuário poderá identificar o mesmo utilizando somente a câmera do dispositivo direcionada ao seu código de barras.
- Atendimento personalizado: efetuando a consulta de informações pela aplicação o usuário terá a oportunidade de receber um atendimento diferenciado baseado em sua localização geográfica.

Para o varejista que utilizar o Atendente Móvel em sua loja terá os seguintes benefícios:

- Oportunidade de atrair novos clientes: a disponibilização de uma ferramenta, como o Atendente Móvel, que oferece novos recursos para o cliente conhecer produtos e facilidades pode ser considerada um grande atrativo para ampliar sua clientela. A crescente popularidade de smartphones e aplicativos para os mesmos também pode ser considerado como um catalisador na busca de novos consumidores.
- Atendimento personalizado: utilizando a configuração adequada para seu ambiente físico, o varejista terá a oportunidade de efetuar um atendimento personalizado a seus clientes baseado no produto consultado, na localização do mesmo no espaço da loja e com informações prévias do perfil do comprador.

- Novo meio de comunicação com o cliente: através da interface da aplicação, o varejista terá mais uma oportunidade de divulgar o seu estabelecimento através de imagens publicitárias (banners). Além disso, também será possível estabelecer um novo grupo consumidor que receberá propagandas e promoções baseado em sua localização.
- Relatórios de presença de clientes: considerando que hoje as empresas, na busca pelo aumento das vendas, não medem esforços em obter informações sobre os seus clientes para enriquecer seus cadastros. Essas informações são utilizadas em sistemas especialistas para fins diversos e campanhas de marketing. Com a utilização do Atendente Móvel o varejista terá em seus registros os logs de consultas feitas no sistema juntamente com a localização do usuário, produtos consultados, etc.

3.2.1 Situações de uso da Aplicação

Para demonstrar os benefícios da aplicação, podemos relacionar algumas situações onde varejistas e clientes utilizariam o sistema e suas respectivas vantagens:

- Atendimento para usuários presentes na loja:
 - Operador cadastra as coordenadas do perímetro da loja, selecionando que para clientes de determinado perfil presentes nesta localidade uma mensagem deverá ser exibida ao operador. Dependendo do espaço físico da loja, poderão ser cadastrados os perímetros exatos de setores da mesma. Por exemplo: “Cliente com bom poder de compra consultando produto no setor de eletrônicos”.
 - Benefícios do usuário: recebe um atendimento pessoal de um vendedor ao efetuar a consulta a um determinado produto.
 - Benefícios do varejista: tem a chance de saber quando um determinado cliente (com bom poder de compra, por exemplo) está presente na loja aumentando a chance de efetuar uma boa venda efetuando um atendimento personalizado.
- Atendimento para usuários que estão presentes em lojas concorrentes:
 - Operador cadastra as coordenadas do perímetro da região com lojas concorrentes (um shopping, por exemplo), selecionando que, para clientes presentes nesta localidade, uma mensagem deverá ser enviada para o smartphone com a seguinte frase: “Cobrimos ofertas dos concorrentes em 10%.”
 - Benefícios do usuário: recebe uma oportunidade de desconto no produto desejado.
 - Benefícios do varejista: tem a oportunidade de oferecer uma vantagem para o cliente que está efetuando a consulta, possivelmente, em uma loja concorrente.
- Utilização do sistema para divulgação de ofertas e propagandas
 - Operador cadastra que para todas as consultas feitas via Atendente Móvel, o sistema deverá repassar alguma mensagem promocional ao

cliente como, por exemplo, “Liquidação de livros neste fim de semana”, “Início da pré-venda do filme X”, etc.

- Benefícios do usuário: oportunidade de receber mensagens de seu interesse.
- Benefícios do varejista: o varejista tem a oportunidade de estabelecer mais um meio de comunicação com o cliente para divulgar informações que possam de algum modo auxiliar nas vendas.

4 MODELAGEM DO SISTEMA

Este capítulo define a modelagem do sistema Atendente Móvel, incluindo o modelo do banco de dados, a definição das classes das aplicações, os diagramas de iterações, considerações sobre o tratamento de falhas e refinamento da arquitetura das aplicações.

4.1 Arquitetura do Sistema

A arquitetura do sistema é uma arquitetura Cliente-Servidor onde a aplicação desenvolvida para o dispositivo requisita informações ao servidor e esse por sua vez atende as requisições dos vários clientes que o acessam. A arquitetura será dividida, para fins de modelagem, em Aplicações e Banco de dados, separando a modelagem dos dados do projeto da implementação. Ao longo do capítulo serão apresentados detalhadamente os componentes apresentados na figura abaixo e de que modo é feita a iteração entre eles.

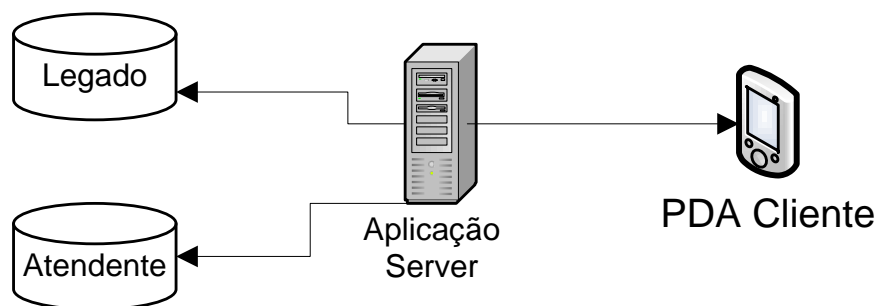


Figura 4.1: Arquitetura do sistema

4.2 Banco de Dados

O projeto do sistema depende fundamentalmente de dados que já sejam de conhecimento do varejista, como o cadastro dos usuários e produtos, e que deverão ser compartilhados com a aplicação que está sendo definida. Diante desta necessidade, a arquitetura do sistema prevê a utilização de dois bancos de dados independentes que serão nomeados como Legado e Atendente e serão apresentados a seguir. O entendimento de suas especificações é fundamental para a compreensão da modelagem das aplicações.

4.2.1 Banco de dados do Legado

O banco de dados denominado como Legado nada mais é do que o banco de dados onde o varejista já possui as informações básicas do seu atual sistema. Este banco de dados eventualmente pode ser o mesmo utilizado por algum outro sistema de vendas, sendo fundamental que nele estejam os dados básicos necessários, como o cadastro de usuários e de produtos com os seus respectivos identificadores.

Para efetuar a modelagem completa da aplicação neste texto foi necessário especificar um modelo de dados que contemple todos os atributos que serão fundamentais na utilização do sistema. Abaixo segue a especificação do modelo que será utilizado para simular o ambiente real de um banco de dados que já deverá estar devidamente preenchido pelo no atual sistema da loja com dados dos cadastros.

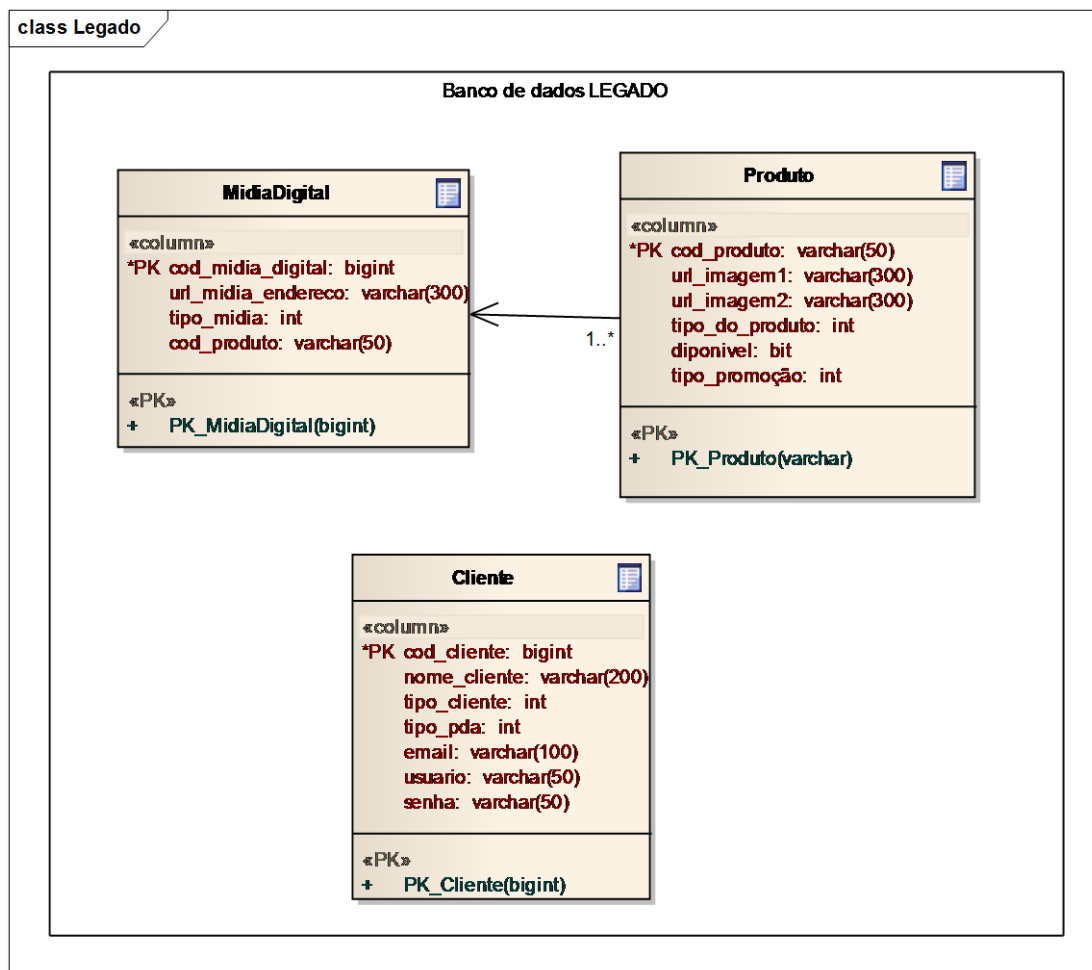


Figura 4.2: Modelo de Dados do Legado

Abaixo segue a relação dos campos e tabelas com as suas respectivas descrições. O tipo dos atributos e tamanho dos campos foi determinado considerando as limitações e restrições do SGBD Microsoft SQL Server 2008.

Tabela 4.1: Definição dos atributos do banco de dados do Legado

Tabela	Atributo	Tipo	Tam	Descrição
Cliente	cod_cliente(PK)	bigint		Código de identificação do usuário.

<i>Cliente</i>	<i>nome_cliente</i>	<i>varchar</i>	200	Nome do cliente.
<i>Cliente</i>	<i>tipo_cliente</i>	<i>int</i>		Classificação do cliente de acordo com o consumo na loja. Definido como de 0(sem compras registradas) até 4(grande numero de aquisições).
<i>Cliente</i>	<i>tipo_pda</i>	<i>int</i>		Tipo de PDA associado ao usuário que define a plataforma que o cliente possui. Para este atributo será definida a seguinte enumeração: 0 - Desconhecido, 1- Windows Mobile, 2 - Iphone, 3- Symbian, 4- Android.
<i>Cliente</i>	<i>email</i>	<i>varchar</i>	100	Email do cliente.
<i>Cliente</i>	<i>usuário</i>	<i>varchar</i>	50	Usuário do cliente utilizado para identifica-lo no sistema.
<i>Cliente</i>	<i>senha</i>	<i>varchar</i>	50	Senha do usuário. Obs.: para fins de prototipação do sistema a senha será armazenada diretamente no banco de dados sem utilizar os padrões básicos de banco de dados.
<i>MidiaDigital</i>	<i>cod_midia_digital(PK)</i>	<i>bigint</i>		Cód. da mídia digital disponível.
<i>MidiaDigital</i>	<i>url_midia_endereco</i>	<i>varchar</i>	300	URL da mídia digital
<i>MidiaDigital</i>	<i>tipo_midia</i>	<i>int</i>		Tipo de mídia armazenada.
<i>MidiaDigital</i>	<i>cod_produto(FK)</i>	<i>varchar</i>	50	Código do produto que a mídia está relacionada.
<i>Produto</i>	<i>cod_produto(PK)</i>	<i>varchar</i>	50	Código do produto.
<i>Produto</i>	<i>url_imagem1</i>	<i>varchar</i>	300	URL da imagem (figura 1) do produto.
<i>Produto</i>	<i>url_imagem2</i>	<i>varchar</i>	300	URL da imagem (Figura 2) do produto.
<i>Titulo</i>	<i>Titulo</i>	<i>varchar</i>	300	Titulo do produto.
<i>Produto</i>	<i>tipo_do_produto</i>	<i>int</i>		Define o tipo do produto cadastrado. (CD, DVD, Livro).
<i>Produto</i>	<i>diponível</i>	<i>bit</i>		Flag que indica se o produto está disponível.
<i>Produto</i>	<i>tipo_promoção</i>	<i>int</i>		Indica o tipo de promoção que o produto faz parte.
<i>Produto</i>	<i>preco</i>	<i>double</i>		Valor monetário do produto.

4.2.2 Banco de dados do Atendente Móvel

O banco de dado do Atendente Móvel é o banco principal da aplicação proposta neste trabalho. Neste serão armazenadas as informações pertinentes às funcionalidade propostas, como os atendimentos personalizados com as suas respectivas localidades do perímetro. Estão contemplados também no modelo as informações de histórico de consultas feitas aos sistemas para fins de relatórios e estatísticas.

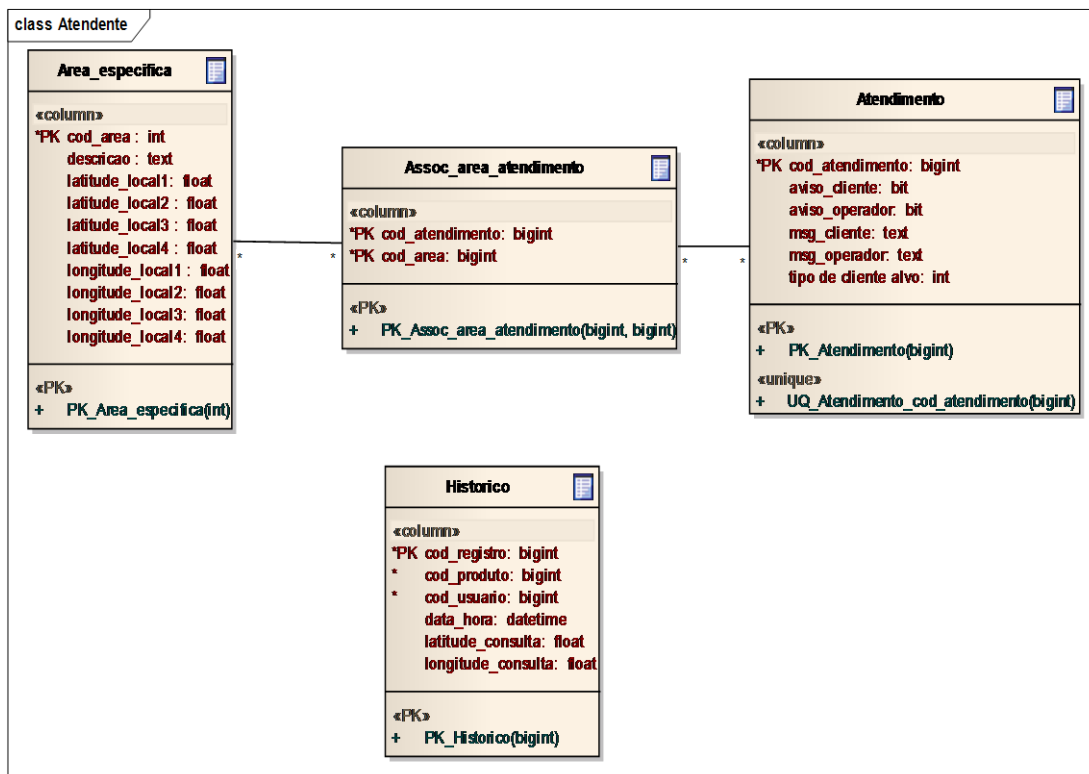


Figura 4.3: Modelo de dados do Atendente Móvel

Assim como o banco de dados do Legado, este banco também teve o seu projeto físico de definido para o SGBD Microsoft SQL Server 2008. Abaixo segue a relação dos campos e tabelas com a suas respectivas descrições.

Tabela 4.2: Atributos dos atributos do banco de dados do Atendente Móvel

<i>Tabela</i>	<i>Atributo</i>	<i>Tipo</i>	<i>Tam</i>	<i>Descrição</i>
<i>Area_especifica</i>	<i>cod_area</i>	<i>int</i>		Identificador único do cadastro
<i>Area_especifica</i>	<i>descricao</i>	<i>text</i>		Descrição da área que foi cadastrada
<i>Area_especifica</i>	<i>latitude_local1</i>	<i>float</i>		Ponto de coordenada Localidade1
<i>Area_especifica</i>	<i>latitude_local2</i>	<i>float</i>		Ponto de coordenada Localidade2
<i>Area_especifica</i>	<i>latitude_local3</i>	<i>float</i>		Ponto de coordenada Localidade3
<i>Area_especifica</i>	<i>latitude_local4</i>	<i>float</i>		Ponto de coordenada Localidade4
<i>Area_especifica</i>	<i>longitude_local1</i>	<i>float</i>		Ponto de coordenada Localidade1
<i>Area_especifica</i>	<i>longitude_local2</i>	<i>float</i>		Ponto de coordenada Localidade2
<i>Area_especifica</i>	<i>longitude_local3</i>	<i>float</i>		Ponto de coordenada Localidade3
<i>Area_especifica</i>	<i>longitude_local4</i>	<i>float</i>		Ponto de coordenada Localidade4

<i>Assoc_Area_Atendimento</i>	<i>cod_atendimento</i>	<i>bigint</i>		Identifica o atendimento da associação.
<i>Assoc_Area_Atendimento</i>	<i>cod_area</i>	<i>bigint</i>		Identifica a área da associação.
<i>Atendimento</i>	<i>cod_atendimento</i>	<i>bigint</i>		Identificador único do registro.
<i>Atendimento</i>	<i>aviso_cliente</i>	<i>bit</i>		Flag que indica se uma mensagem deverá ser enviada para o cliente.
<i>Atendimento</i>	<i>aviso_operador</i>	<i>bit</i>		Flag que indica se uma mensagem deverá ser enviada para o operador do sistema.
<i>Atendimento</i>	<i>msg_cliente</i>	<i>text</i>	250	Mensagem que deverá ser enviada para o cliente.
<i>Atendimento</i>	<i>msg_operador</i>	<i>text</i>	250	Mensagem que deverá ser enviada para o cliente.
<i>Atendimento</i>	<i>tipo de cliente alvo</i>	<i>int</i>		Classificação do cliente de acordo com o consumo na loja. Definido como de 0 (sem compras registradas) até 4 (grande número de aquisições). Definido de acordo com o Legado.

O cadastro das informações deste banco de dados deverá ser feita por uma aplicação paralela ao sistema e que não está projetada neste trabalho. Esta aplicação poderá ser um novo software ou apenas um novo módulo no software de gestão já utilizado pelo futuro proprietário do sistema. Serão necessárias telas de cadastro, edição, exclusão e consultas para os atendimentos e suas respectivas áreas.

4.3 Aplicações

O sistema do Atendente Móvel engloba a utilização de duas aplicações que interagem entre si através das chamadas *WebService*. A partir deste ponto a aplicação cliente do dispositivo será chamada de *Atendente Client* e a aplicação que rodará no servidor será denominada como *Atendente Server*. Nas próximas seções será abordado detalhadamente o funcionamento de cada uma das aplicações envolvidas.

4.3.1 Atendente Client

Um dos grandes desafios no projeto de uma aplicação para dispositivos móveis como celulares e smartphones é modelar os componentes da mesma de modo que os mesmos sejam genéricos o suficiente para que a implementação possa ser feita em mais de uma plataforma, sistema operacional e tipo de hardware. O projeto do *Atendente Client* foi feito para que possa ser portado para diferentes linguagem e dispositivos, porém em algumas situações foram necessárias algumas definições de baixo nível, como o uso da câmera e do GPS. Para estes casos foi considerado como sistema operacional alvo o *Windows Mobile 6.0* da Microsoft, que também é a plataforma onde o protótipo foi desenvolvido.

4.3.1.1 Arquitetura do Atendente Client

O Atendente Client será um programa que ficará armazenado no dispositivo do usuário e a partir de sua execução o mesmo poderá usufruir das funcionalidades já relatadas. Esta aplicação terá sua arquitetura própria, definida pelos componentes demonstrados na figura 4.4.

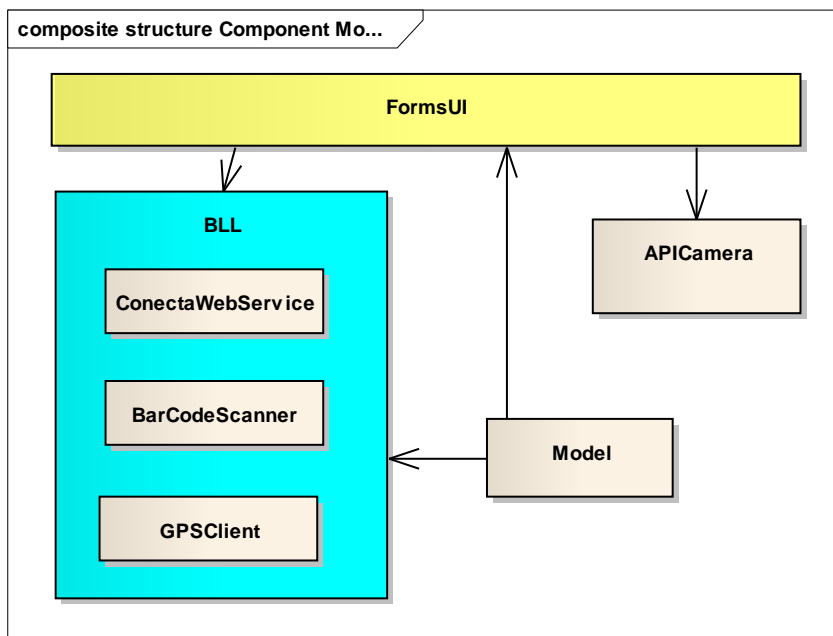


Figura 4.4: Arquitetura do aplicação

As estruturas apresentadas são conceituadas da seguinte forma:

- FormsUI: parte da arquitetura onde serão definidas todas as interfaces da aplicação e define todos os eventos de iteração com o usuário. Acessa a camada BLL e também a APICamera.
- BLL: (Business Logic Layer): Nesta camada ficam as classes responsáveis pela área de negócio da aplicação utilizados para comunicação, gerenciamento e sincronização das informações. Interage com as principais camadas da arquitetura, centralizando as informações que serão enviadas para a interface.
- Model: camada onde estão definidas todas as entidades e os tipos abstratos de dados, que serão utilizados para transportar e representar os dados entre as camadas.
- GPSCient.: componente que interage diretamente com o hardware do dispositivo para obter informações do GPS. Seu funcionamento e implementação serão descritos com mais detalhes no capítulo 4.
- APICamera: este componente fornece a aplicação os métodos para manipulação da câmera do dispositivo. Seu funcionamento e implementação serão descritos com mais detalhes no capítulo 4.
- ConectaWebServer: Ultima camada da aplicação, responsável interagir com o Webservice.
- Scanner: componente responsável por executar o algoritmo para scanear a imagem com o código de barras.

4.3.1.2 Diagrama de Classe

A figura 4.5 ilustra o Diagrama de Classes da aplicação Atendente Client incluindo importantes detalhes que serão seguidos na implementação, como assinatura de métodos, atributos, propriedades e interação entre as classes. Este diagrama omite classes que não fazem parte do fluxo principal da aplicação e que são utilizadas como auxílio aos módulos GPSClient e Scanner.

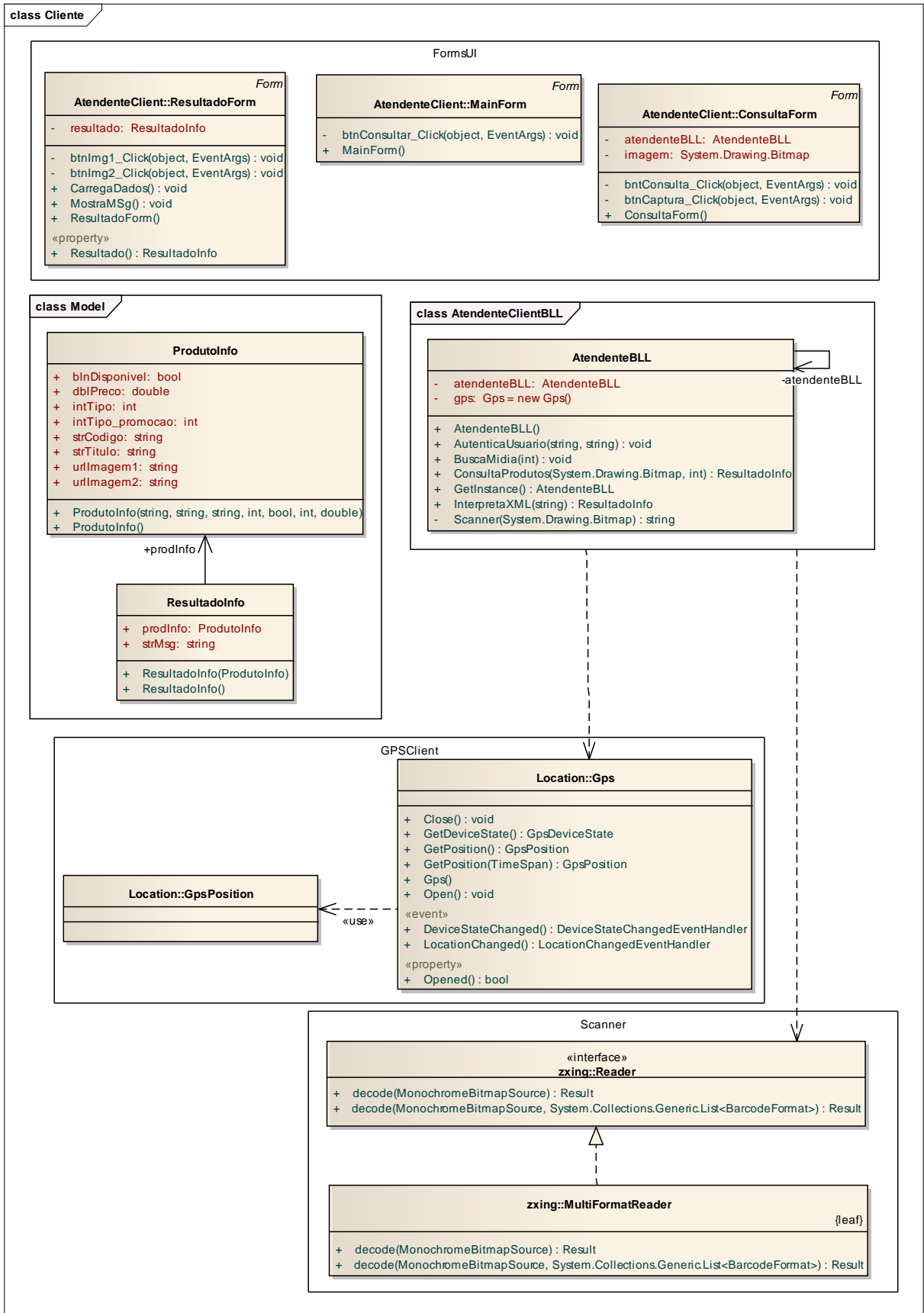


Figura 4.5: Diagrama de classes do Cliente

4.3.1.3 Funcionalidades

O principal objetivo desta aplicação será interagir com as funcionalidades disponibilizadas pela aplicação do servidor, o Atendente Server, que será modelada na sequência do texto.

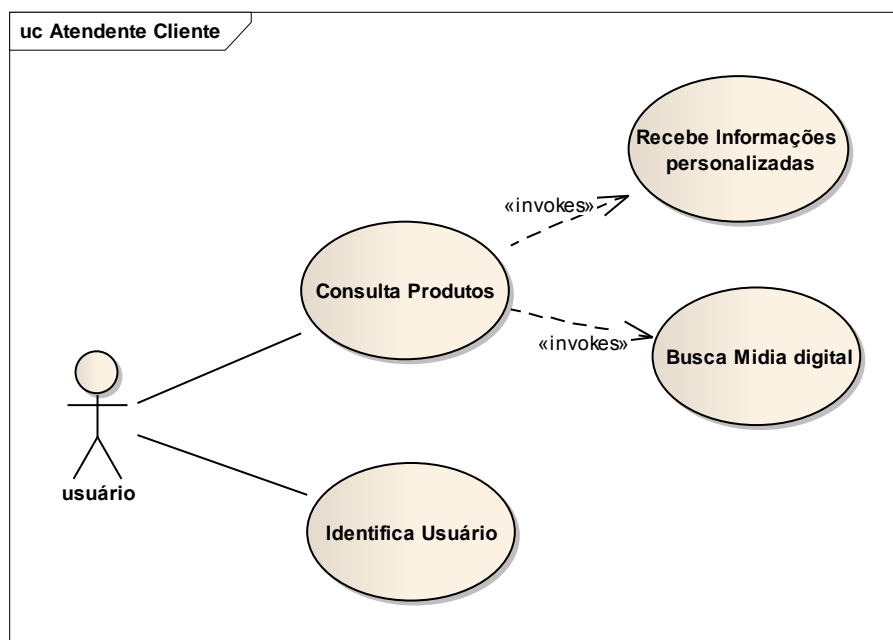


Figura 4.6: Casos de uso do cliente

Conforme demonstrado acima, no diagrama de casos de uso da aplicação, existem duas funcionalidades que serão inicializadas diretamente por ações do usuário através da interface do dispositivo. São elas:

- **Identifica Usuário**

Todas as interações da aplicação com as ações realizadas pelo servidor estão relacionadas com a identificação do usuário. Para atender a este requisito o sistema disponibiliza uma tela onde o usuário informa o usuário e sua senha. Essas informações serão enviadas ao serviço do Atendente Server que autentica o usuário, o web método *AutenticaUsuario* (que será especificado posteriormente). Caso a autenticação do cliente seja feita com sucesso, a aplicação receberá o código do mesmo e armazenará a mesma no arquivo de configuração da aplicação. O armazenamento desta credencial possibilita que o usuário efetua a identificação somente uma única vez, nos demais acessos será utilizado o código armazenado. A modelagem desta funcionalidade não abrange questões críticas e necessárias de segurança, como criptografia e protocolos seguros, pois os mesmos não fazem parte do foco principal do estudo deste texto. Para um futuro uso comercial da aplicação este deverá ser um ponto que será fortemente redefinido para a implantação do sistema.

- **Consulta Produtos:**

A consulta de produtos se destaca como a principal funcionalidade do sistema por envolver os principais benefícios da aplicação e por sua complexidade do ponto de vista tecnológico. Depois de já feita uma primeira identificação do usuário, o cliente poderá efetuar a consulta de informações sobre um determinado produto. Este será reconhecido automaticamente pelo sistema com o uso da câmera do dispositivo voltada para seu

código de barras. Paralelamente a aplicação também estará identificando a localização geográfica do usuário. Estas informações serão utilizadas na consulta que será feita ao servidor. Obtendo sucesso em todas as operações efetuadas o cliente visualizará na tela de seu dispositivo móvel as informações detalhadas do produto de seu interesse e também terá a possibilidade de receber uma mensagem personalizada oferecendo algum serviço ou promoção, denominada Atendimento Personalizado. O fluxo completo das atividades que envolvem esta funcionalidade poderá ser observado na figura 4.7.

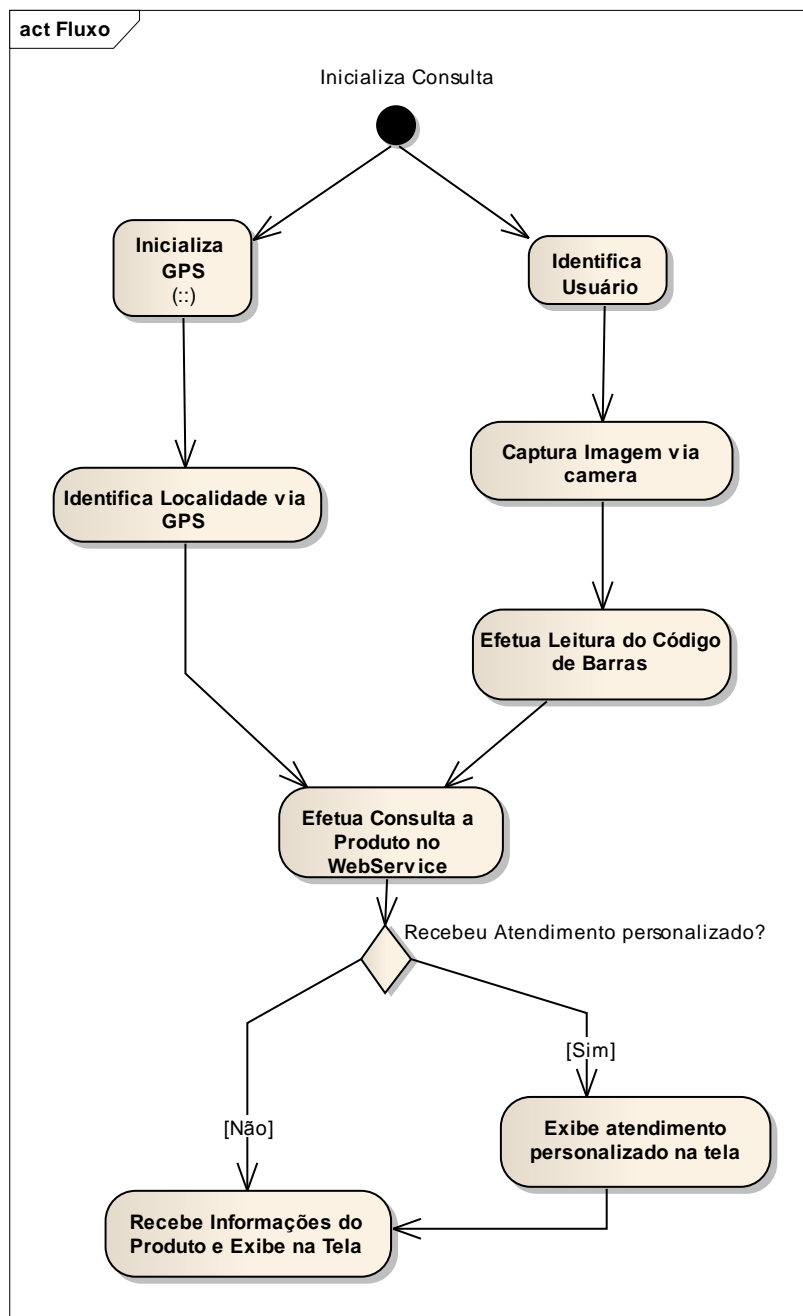


Figura 4.7: Diagrama de atividades da consulta de produtos.

Este fluxo é iniciado pelo usuário na tela de consulta de produtos e utiliza todos os elementos da arquitetura da aplicação. Os cenários do fluxo básico deste caso de uso serão descritos abaixo com os devidos casos de exceção.

1. Captura da imagem: no formulário de consulta um painel com a imagem em movimento captada pela câmera do dispositivo estará sendo exibido. O usuário deverá direcioná-lo para o código de barras do produto e apertar o botão *Capturar*. A imagem capturada será salva em arquivo com o nome *captura.jpg*. Automaticamente o botão Consultar estará habilitado e o mesmo deverá ser acionado.
Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo.
2. Execução do Scanner: com o arquivo da imagem salvo na memória do dispositivo, o módulo de identificação, o *Scanner*, é executado para o reconhecimento do código de barras. Este módulo é apenas uma interface para a biblioteca responsável pelo algoritmo de reconhecimento, que no caso desta modelagem será o *Zxing*[ZXING,2009] representado pela classe *Reader* no diagrama (detalhes do uso desta biblioteca serão feitos no capítulo 4). Em caso de sucesso será conhecido o código do produto escolhido pelo cliente e o fluxo da funcionalidade continuará.
Caso de Exceção: caso o algoritmo não reconheça nenhum código na imagem a mensagem “Não foi possível identificar o código de barras” será exibida para o usuário e fluxo será reiniciado.
3. Obtendo localização geográfica: a identificação das coordenadas do cliente é uma informação obrigatória para a continuidade da consulta. Para isso é acionado o módulo *GPSClient* que fornecerá a latitude e a longitude de onde a consulta está sendo realizada.
Pré-requisito: a conexão com o GPS deverá ser iniciada antes deste cenário. A mesma deverá ser feita no início da execução da aplicação.
Caso de Exceção: caso a conexão com o GPS não esteja ativa ou não atinja o nível mínimo de qualidade desejada, a aplicação deverá exibir a seguinte mensagem ao cliente: “Problemas com a conexão ao GPS”.
4. Consulta informações no Servidor: com todos os parâmetros já carregados: código do produto, código do usuário (obtido das configurações da aplicação), latitude e longitude, a aplicação pode efetuar a consulta com o *Web Services* para obter os dados desejados. O caminho do servidor deverá estar armazenado no arquivo de configuração padrão da aplicação e o web método a ser chamado é o *ConsultaProduto*. Obtido sucesso na transação uma *string* com as informações será recebida e deverá permanecer armazenada em memória.
Pré-requisito: o dispositivo deverá ter algum tipo de conexão com a internet e a aplicação deverá ter autorização para trafegar dados na rede utilizada.
Caso de Exceção: caso não seja possível estabelecer conexão com o servidor por indisponibilidade da rede a seguinte mensagem deverá ser exibida ao usuário: “Rede de dados indisponível, verifique configurações”. Também podem ocorrer erros no processamento do servidor, que fará uma série de validações sobre os dados enviados e o cliente deverá estar preparado para receber uma exceção de erro. Neste caso a mesma deverá ter sua mensagem diretamente exibida ao usuário, pois já estará tratada com conteúdo amigável ao cliente.
5. Interpreta XML recebido: no cenário anterior o cliente recebe uma *string* como resultado da consulta onde seu conteúdo é a sequência de caracteres que compõem um arquivo XML, sendo assim a mesma deverá ser interpretada como

tal. Todas as informações contidas neste arquivo são suficientes para que aplicação possa criar um objeto da classe *ResultadoInfo* (vide diagrama). Este objeto será a base para o próximo formulário de exibição.

Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo.

6. Fluxo Alternativo – Exibir Atendimento Personalizado: caso o objeto criado no cenário anterior tenha armazenado a informação de *strMsg*, deverá ser invocada a funcionalidade de Exibir Atendimento Personalizado. A mesma será responsável por criar um formulário temporário com a mensagem recebida do servidor. Após acionar o botão *Ok* o usuário retorna o fluxo da consulta de produtos.
Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo
7. Exibir informação do produto: a funcionalidade é concluída quando o formulário de exibir as informações do produto é instanciado com base no objeto *ResultadoInfo* criado.
8. Fluxo Alternativo – Busca de Mídia Digital: com as informações sobre o produto desejado exibidas na tela do dispositivo, o usuário terá a opção de visualizar conteúdos multimídia sobre o mesmo. Na visualização de um CD ou DVD, por exemplo, o cliente poderá requisitar a apreciação de um trecho de uma determinada música ou até mesmo um trailer relacionado ao vídeo escolhido. Para obter essas informações a aplicação cliente deverá consultar o web método *BuscaMidiaDigital* para receber o arquivo associado ao produto passado como parâmetro. Concluído o recebimento dos dados de retorno, será feita a chamada ao aplicativo compatível com a mídia recebida para que a mesma possa ser reproduzida.

Sobre este ultimo cenário é importante ressaltar que, apesar de envolver um dos principais benefícios oferecidos ao cliente, a exibição de conteúdo multimídia não faz parte do enfoque principal deste texto. Detalhes sobre a sua implementação e projeto serão relevados neste capítulo, ficando registrado apenas o modelo previsto para mesma. Em uma futura implantação do sistema esta funcionalidade deverá ser especificada com mais rigor.

Na especificação dos cenários podemos verificar que a funcionalidade de consultar produtos efetua uma transação completa sobre todas as camadas da arquitetura, sendo iniciada na interface (FormsUI) e realizando uma chamada ao Web Services. Apesar disso a maioria da lógica e chamada de classes de apoio (GPSClient e Scanner) fica definida na camada de negócios (BLL), como pode ser visto no diagrama de sequência a seguir:

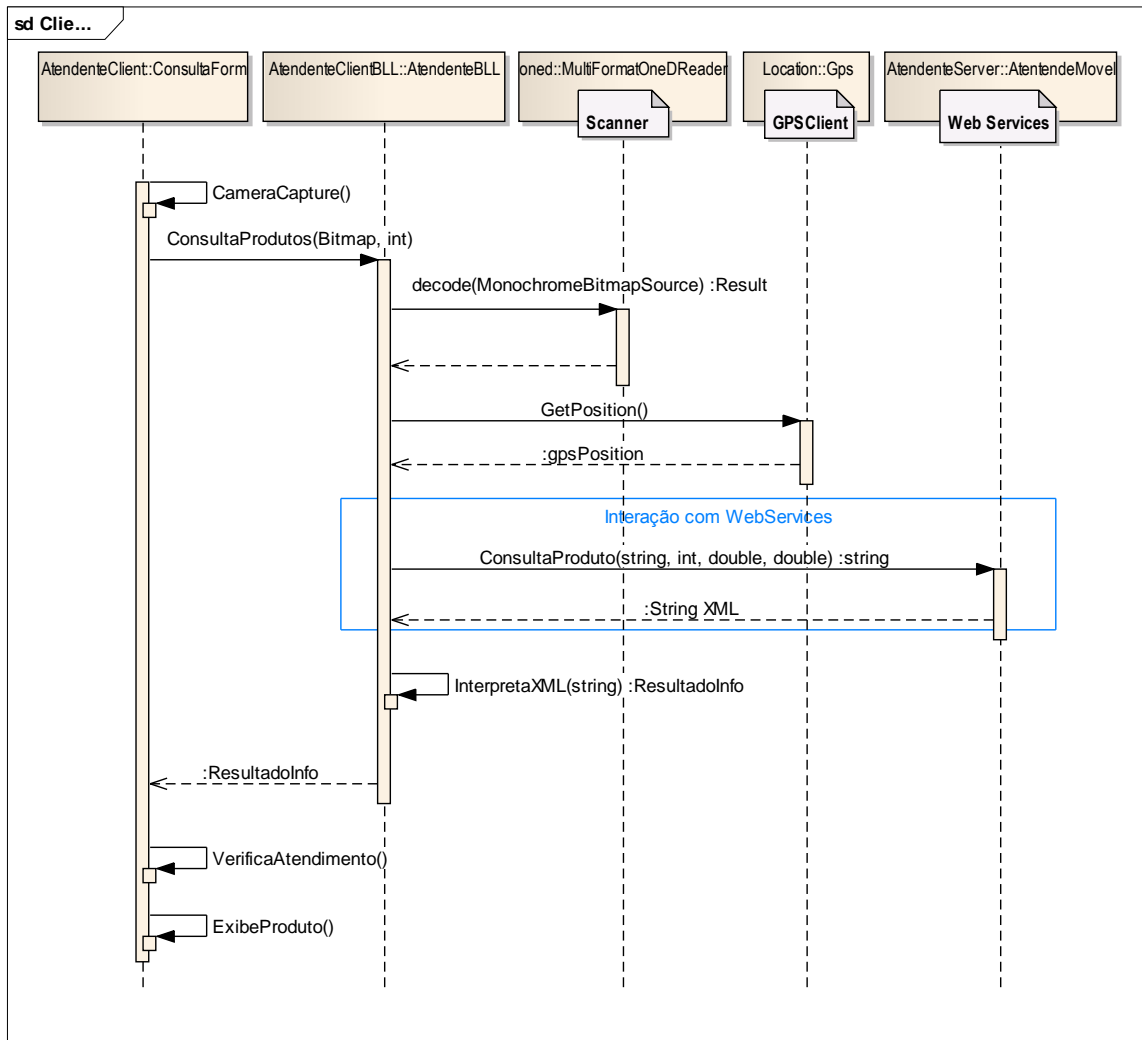


Figura 4.8: Diagrama de atividades da consulta de produtos

4.3.2 Atendente Server

A aplicação que será executada no servidor será denominada por Atendente Server. Este módulo do sistema terá como principais funções a integração com o banco de dados e a disponibilização das funcionalidades via Web Services. Esta interface web deverá fornecer para os dispositivos os três serviços referenciados pelos casos de uso demonstrado na figura 4.9.

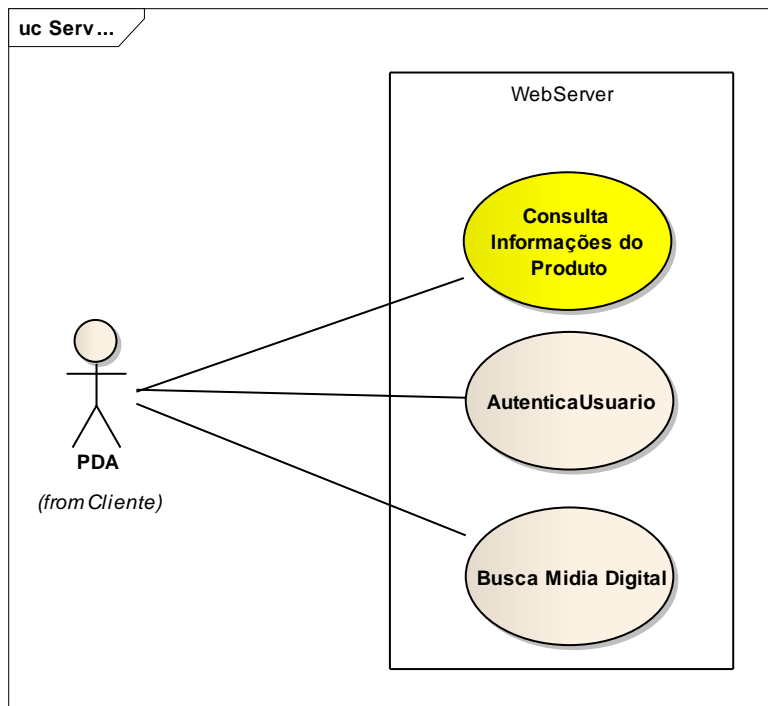


Figura 4.9 Diagrama de Caso de Uso do Atendente Server

Para a modelagem da aplicação foi escolhida a arquitetura padrão para desenvolvimento em camadas, sendo ela dividida em Camada de Apresentação, Camada de persistência e Camada de negócio. A nomenclatura destes componentes segue, respectivamente, o seguinte padrão sugerido pela MSDN (TECHNET, 2010):

- Model: camada onde estão definidas todas as entidades e os tipos abstratos de dados, que serão utilizados para transportar e representar os dados entre as camadas;
- DAL(*Data Access Layer*): Camada de acesso ao banco de dados onde está a lógica de conexão com o banco de dados e a definição dos scripts SQL para inserções e consultas que deverão ser efetuadas no SGBD;
- BLL(*Business Logic Layer*): São definidas nesta camada as classes responsáveis pela área de negócio da aplicação que são responsáveis pela comunicação, gerenciamento e sincronização das informações. Interage com funções das classes do DAL para persistir os dados no banco local;
- Apresentação: usualmente definida como uma interface definida por Formulários ou páginas Web, nesta modelagem esta camada de apresentação será composta pelo Web Services que definem os serviços que serão utilizados pelos smartphones.

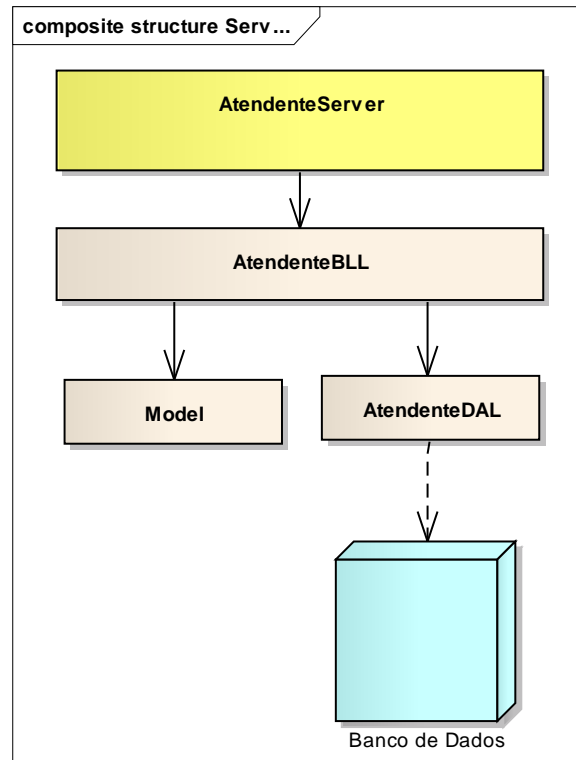


Figura 4.10: Arquitetura do Atendente Server

Os aspectos referentes à implementação serão abordados com maiores detalhes no capítulo seguinte, porém é importante ressaltar que toda a modelagem desta aplicação foi baseada na tecnologia Microsoft.NET, com as suas respectivas bibliotecas, e no sistema operacional Windows Server, com o servidor de aplicação IIS.

4.3.2.1 Diagrama de Classes

A figura 4.11 ilustra o Diagrama de Classes da aplicação Atendente Server incluindo importantes detalhes que serão seguidos na implementação, como assinatura de métodos, atributos, propriedades e interação entre as classes. A nomenclatura das classes permite identificar a camada em que cada uma pertence através do sufixo utilizado: DAL, BLL ou Info.

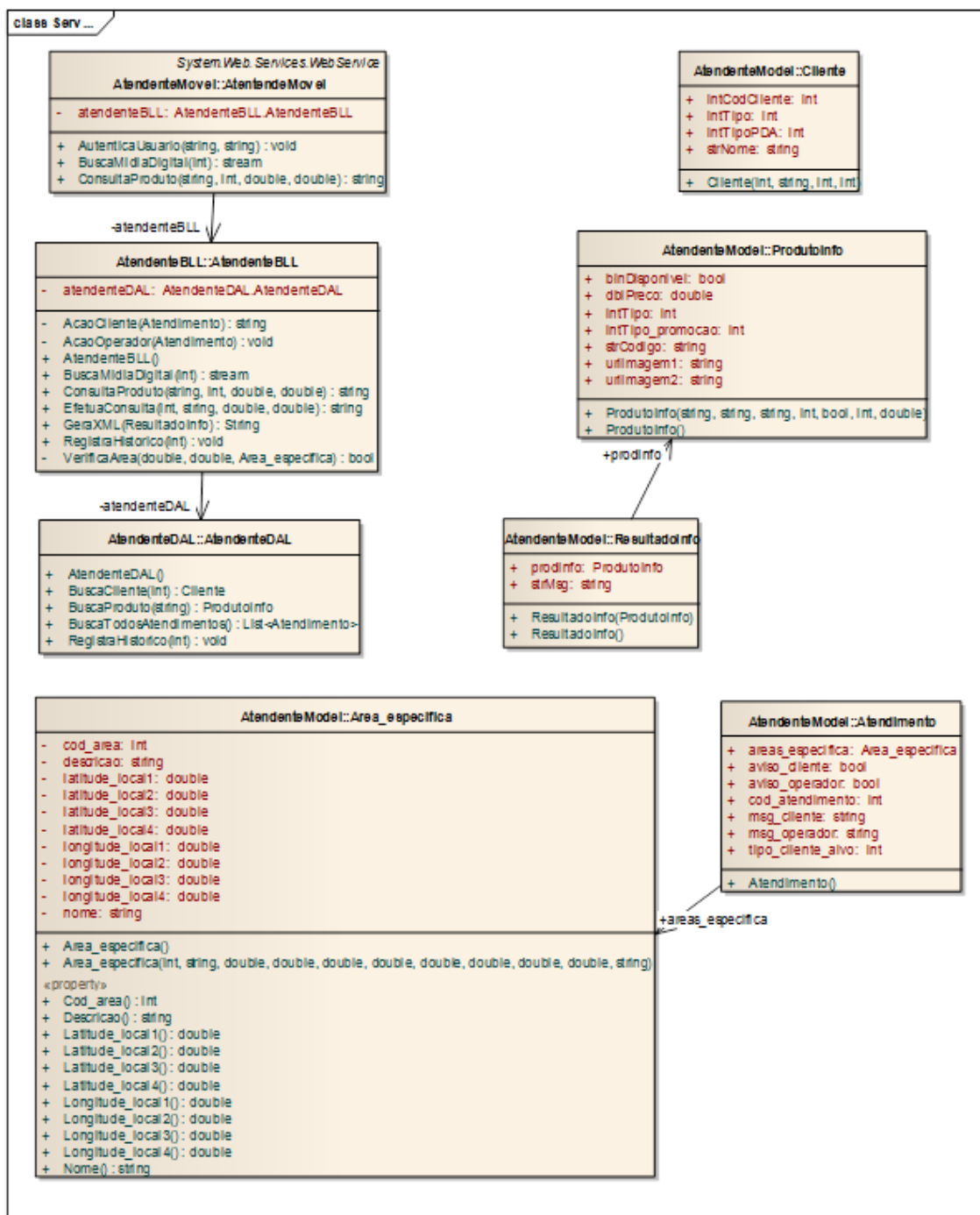


Figura 4.11: Diagrama de Classes do Atendente Server

Nesta ilustração é possível identificar as classes que serão utilizadas para representar as entidades do sistema e fazem parte da camada *Model*. São elas: *Area_especificaInfo*, *AtendimentoInfo*, *ResultadoInfo*, *ProdutoInfo* e *Cliente*. A definição destes componentes segue com rigor os atributos definidos no banco de dados e os mesmos serão compartilhados pelas demais classes.

4.3.2.2 Funcionalidades

Como visto no diagrama de classes, a classe principal que define o serviço é a *AtendenteMovel*. É nesta classe que estão definidos os web métodos do *WebService* que provêm as seguintes funcionalidades aos dispositivos:

- **Autenticação do Usuário**

Para ser identificado corretamente pela aplicação o usuário deverá efetuar a autenticação no sistema. Essa autenticação deverá ser feita ao menos uma única vez, ficando armazenado no dispositivo as informações do cliente. O web método que deverá ser responsável por esta autenticação é *AutenticaUsuario* que recebe os parâmetros senha(*string*) e usuário(*string*) e que validará essas informações no banco de dados do Legado. Em caso de sucesso na autenticação, o código do cliente será retornado ao usuário para que o mesmo seja utilizado em consultas e chamadas a outras funcionalidades. É importante ressaltar que, por não se tratar de ser o propósito principal deste trabalho, aspectos de segurança não foram considerados na modelagem. Não está prevista nenhuma criptografia para a manipulação da senha e nome do usuário, assim como não é feita nenhuma autenticação adicional nas demais funcionalidades do Web Service. Para um futuro uso comercial da aplicação este deverá ser um ponto que ser fortemente redefinido para a implantação do sistema.

- **Busca de Mídia Digital**

Como forma de motivar os clientes para o uso da aplicação, o sistema prevê a disponibilização de arquivos de mídias digitais, como vídeos, músicas e textos. Estes arquivos seriam utilizados pela aplicação clientes para exibir trailers de filmes, trechos de músicas e livros, que estejam relacionados com o produto consultado. O web método responsável por esta funcionalidade é o *BuscaMidiaDigital* que recebe como parâmetro o código da mídia e retorna um *Stream* de arquivo. O tipo de mídia é uma informação que está associada ao produto e a mesma já é de conhecimento do smartphone do cliente que deverá efetuar o procedimento mais adequado para reproduzir o arquivo recebido. Assim como a funcionalidade da Autenticação, a busca e reprodução da mídia não é foco principal deste trabalho e por este motivo sua modelagem não inclui definições detalhadas de como a mesma será implementação.

- **Consulta de Produtos**

Grande parte do diferencial da aplicação está definida nesta funcionalidade, sendo que uma parte significativa da lógica da mesma é de responsabilidade do servidor que será modelado neste capítulo. A consulta de produtos inicia seu fluxo a partir de uma chamada feita pelo dispositivo cliente ao web método *ConsultaProduto* juntamente com os parâmetros código do produto, código do cliente, latitude e longitude de onde a consulta está sendo feita. Com base nestas informações o servidor inicia o fluxo que está demonstrado no Diagrama de Sequencia abaixo.

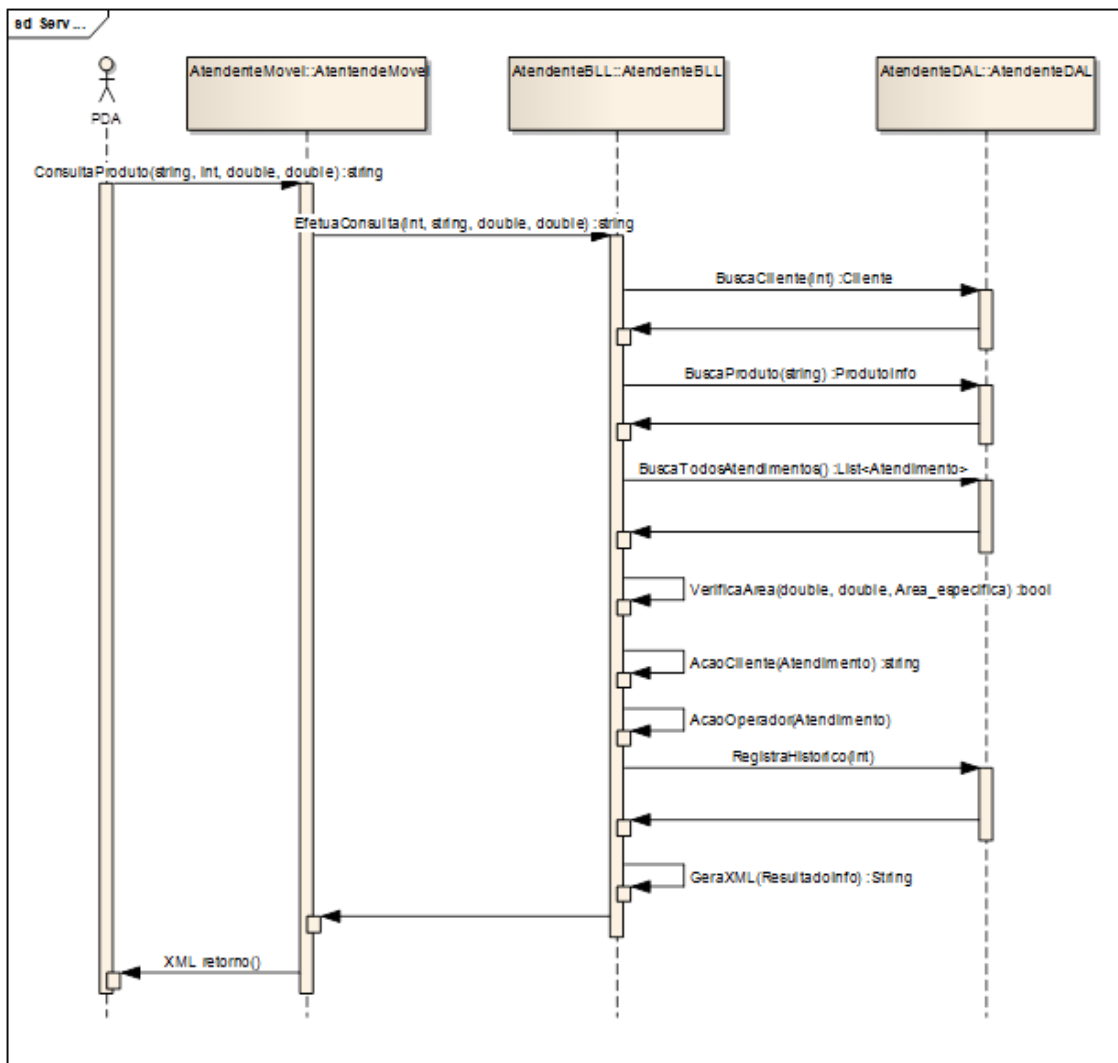


Figura 4.12: Diagrama de Sequência do Atendente Server

Após o recebimento das informações, o Web Service aciona a camada que efetua a lógica de negócio e as validações dos parâmetros chamando o método *EfetuaConsulta* da classe *AtendenteBLL*. Neste método a seguinte sequência de cenários é executada:

1. Validação do Cliente: com base no código do cliente recebido é feita uma chamada à classe *AtendenteDAL* para que seja verificado no banco de dados do Legado a existência de um cliente correspondente. Caso o mesmo exista, um objeto da classe *ClienteInfo* será recebido com as informações obtidas do banco.

Caso de Exceção: caso o cliente não exista o fluxo será encerrado com uma exceção e o erro de “Cliente não encontrado” será retornado para a camada superior.

2. Validação do Produto: com base no código de produto informado, uma nova chamada ao banco de dados do Legado será efetuada utilizando o método *AtendenteDAL.BuscaCliente*. Um objeto de *ProdutoInfo* será recebido em caso de sucesso na busca.

Caso de Exceção: caso o produto não exista o fluxo será encerrado com uma exceção e o erro de “Produto não catalogado” será retornado para a camada superior.

3. Busca de Atendimentos: efetuada as devidas validações, é feita uma consulta ao banco de dados Atendimento para obter a lista de atendimentos cadastrados. Em caso de sucesso na busca, uma lista de objetos *AtendimentoInfo* serão recebidos da classe DAL. É importante ressaltar que o método *AtendenteDAL.BuscaAtendimentos* tem também a incumbência de buscar as áreas específicas, definidas como outra lista de objetos *Area_EspecificInfo* dentro do objeto Atendimento. Este relacionamento pode ser visualizado no diagrama de classes.

Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo.

4. Verificar se a consulta feita está em uma área de atendimento cadastrado: Neste momento do fluxo é iniciada uma iteração sobre os objetos da lista de atendimento recebida no cenário anterior. Para cada um deles será feita uma chamada ao método privado *VerificaArea* para verificar se a consulta feita está no perímetro delimitado para o determinado atendimento. Caso o retorno da verificação seja possível, é necessário verificar as flags que indicam se o atendimento cadastrado indica uma ação personalizada ao cliente e/ou ao operador do sistema. Para estas situações positivas são iniciados os cenários 5 e 6.

Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo.

5. Inicia ação para o cliente: para o caso onde campo *aviso_cliente* do atendimento identificado estiver com o valor positivo o método *AcaoCliente* será acionado. Este método preenche o objeto da classe *ResultadoInfo* com o atendimento que deverá ser retornado ao cliente.

Caso de Exceção: caso exista mais de um atendimento para a consulta feita a exceção adequada deverá ser retornada (para fins de logs), porém a mesma não interrompe o fluxo do método.

6. Inicia ação para o operador: para o caso onde campo *aviso_operador* do atendimento identificado estiver com o valor positivo o método *AcaoOperador* será acionado. Este será responsável por emitir a mensagem determinada no atendimento para a interface do operador, que poderá ser definida como uma caixa de mensagem na tela do servidor ou até mesmo o envio de um e-mail para o responsável. Nesta modelagem simplificada o aviso ao operador do sistema será feito através de uma escrita em um arquivo texto.

Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo.

7. Registra Histórico de consultas feitas: depois de efetuada a lógica para determinar o atendimento que deverá ser encaminhado ao usuário, é necessário registrar a consulta feita no banco de dados. Para isto é chamado o método *RegistraHistorico* do DAL passando o objeto de *ResultadoInfo* criado, que cria um novo registro na tabela *Historico* do banco de dados Atendimento.

Caso de Exceção: para este cenário não existe uma regra de negócio que interrompa o fluxo.

8. Gerar XML de retorno: durante o fluxo desta funcionalidade, um objeto da classe *ResultadoInfo* foi abastecido com as informações obtidas (produto e atendimento com mensagem ao cliente). A partir deste objeto é que o método *GeraXML* cria um arquivo XML com as informações que serão retornadas ao cliente. Por se tratar de um arquivo pequeno, o mesmo será retornado como uma *string*, facilitando a manipulação do mesmo.

Caso de Exceção: caso algumas das informações não estejam corretamente armazenadas no objeto, uma exceção deverá ser lançada interrompendo o fluxo do método com a mensagem “Erro no processamento das informações”.

A Consulta de produtos encerra com o web método retornando para o dispositivo, que efetuou a consulta, a *string* resultante da execução do método *EfetuaConsulta* da cada BLL. Para casos de erro fica a cargo do serviço a identificação do problema para que a mesma seja tratada e logada de maneira correta evitando que erros técnicos sejam destinados ao cliente.

Alguns pontos desta modelagem foram definidos de forma abstrata, sem a especificação de algoritmos que deverão ser utilizados nem demonstrados os modelos de dados da aplicação (modelo do XML, por exemplo). Estes aspectos técnicos serão demonstrados no capítulo 5 juntamente com os aspectos de implementação feitos no protótipo.

5 IMPLEMENTAÇÃO

A partir da modelagem feita no capítulo 4, foi implementado um protótipo com as principais funcionalidades a fim de validar o modelo proposto. Para isto será apresentado neste capítulo a implementação das aplicações do cliente e do servidor, com a descrição do funcionamento das mesmas, detalhando os pontos mais complexos, assim como tecnologias que foram utilizadas e testes realizados.

5.1 Escopo do Protótipo

Este protótipo é a demonstração de uma parte do possível sistema, portanto será implementada a funcionalidade principal do sistema que é a Consulta de Produtos. Foram codificadas as aplicações cliente e servidor, contemplando os aspectos referentes à localização geográfica via GPS, o scanear de imagens para identificação do código de barras, interação como Webservice, interpretação e geração dos arquivos XML e a interface do sistema com o usuário.

5.2 Ambiente de desenvolvimento

Para a codificação do protótipo foi utilizada a linguagem de programação C# com a IDE Visual Studio 2008 juntamente com todo o conjunto de ferramentas e bibliotecas do ambiente Microsoft para desenvolvimento, assim como os padrões fornecidos pela MSDN.

A aplicação do servidor foi compilada e desenvolvida na plataforma do .NET Framework 3.5, sendo este um pré-requisito para sua execução. O servidor de aplicação web utilizado é o IIS 6.0 rodando em Windows Server 2003.

Na aplicação cliente foi utilizado o .NET Compact Framework 3.5 agregado com o Windows Mobile 6 Professional SDK, sendo o respectivo sistema operacional um requisito do dispositivo que rodará a mesma.

5.3 Apresentando o Protótipo

Nesta seção será apresentado o protótipo funcional do Atendente Móvel que foi desenvolvido neste trabalho. Todos os aspectos técnicos descritos neste capítulo assim como o projeto da funcionalidade Consulta de Produtos descrita no capítulo anterior foram integralmente codificados e testados.

Na tela inicial da aplicação, apresentada na figura 5.1, temos as informações de *login*, para o usuário inserir suas credenciais. No ponto de vista funcional essas informações não tem utilidade, pois o protótipo considera que o usuário já fez uma

autenticação no sistema, assumindo um valor fixo para o código de cliente que deverá estar armazenado no arquivo de configuração.



Figura 5.1: Tela inicial da aplicação

O próximo passo do usuário é acionar o botão Consultar Produtos para iniciar a referida funcionalidade. Neste momento a aplicação efetiva a inicialização da câmera do dispositivo para capturar a imagem do código de barras do produto desejado. Para facilitar os testes e a apresentação das telas do sistema neste texto, foi criada uma pequena variação do protótipo que não inicializa a câmera do dispositivo, selecionando uma imagem definida e armazenada em memória. Esta simplificação se deve ao fato do emulador do Windows Mobile não simular o driver da câmera de modo que imagens possam ser obtidas.



Figura 5.2: Tela de captura da imagem

Na tela é exibida a um painel com a possível imagem (via câmera ou arquivo) e o usuário tem a opção de confirmar a escolha da mesma clicando no botão Capturar, conforme a figura 5.2. Efetivada a captura da imagem o botão Consulta fica habilitado, e quando acionado pelo usuário inicia o processo definido para esta funcionalidade. Dependendo da capacidade de processamento do dispositivo e da capacidade de trafegar dados da rede utilizada, este processo pode demorar alguns segundos fazendo com que a ampulheta default do sistema operacional seja exibida. Caso o scaneamento da imagem seja concluído com sucesso e o código identificado esteja na base do sistema, o programa segue para a próxima tela.

A tela de resultados carrega todos os dados recebidos do servidor e caso a consulta feita pelo usuário obtenha tenha um Atendimento personalizado cadastrado, o mesmo será exibido. Esta mensagem será exibida sobrepondo os demais dados do produto, conforme figura 5.3.

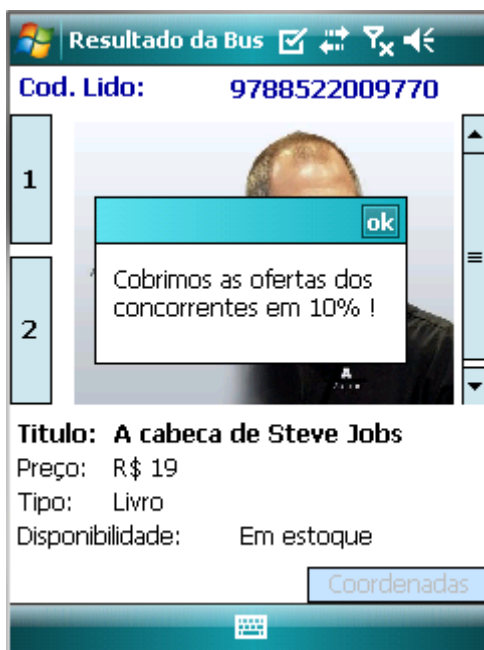


Figura 5.3: Tela com o Atendimento Personalizado

Após a leitura da mensagem o usuário poderá fechar a mesma pelo botão OK no canto da janela e com isso poderá visualizar as demais informações do produto, conforme figura 5.3. No topo da tela é exibido o código lido do produto, e logo abaixo temos a imagem 1 do produto. Ao lado da imagem temos os botões 1 e 2, que permitem que o usuário alterne entre as fotos do produto consultado. Logo abaixo temos as informações de título, preço, tipo do produto e disponibilidade.

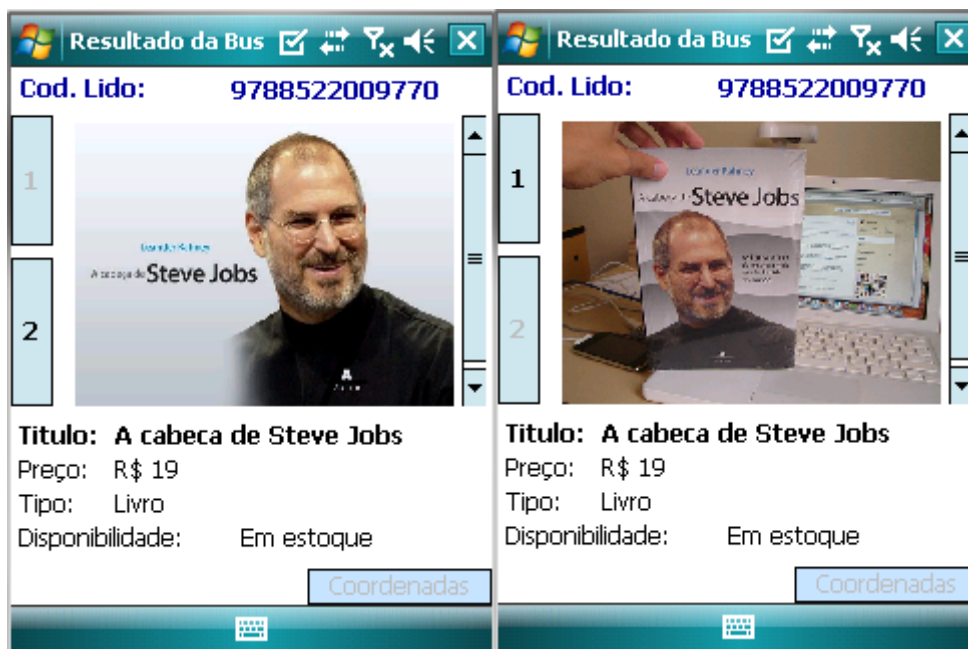


Figura 5.4: Tela que exibe as duas imagens do produto.

Ainda na tela que demonstra os dados do produto, temos o botão **Coordenadas**, localizado no lado direito inferior da interface. Acionando este botão é possível visualizar as coordenadas que foram repassadas ao servidor (conforme demonstrado na figura 5.5). Este botão não é de grande utilidade para usuário, mas do ponto de vista de testes da aplicação o mesmo se torna extremamente útil.

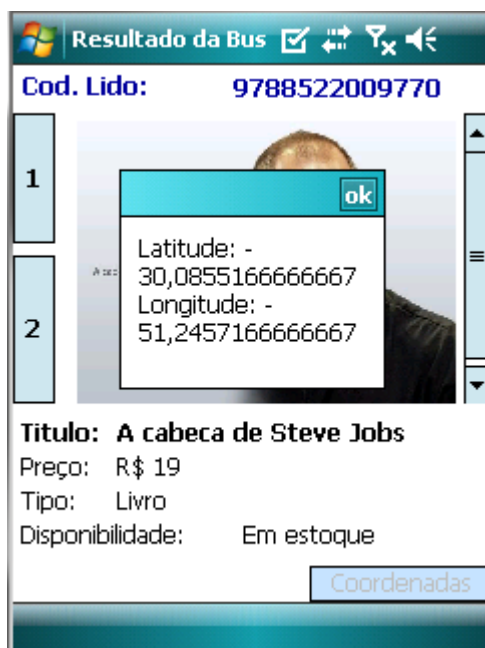


Figura 5.5: Tela com a exibição das coordenadas.

Para a consulta demonstrada nesta consulta tivemos também um processamento efetuado pelo servidor. As operações feitas pelo servidor são registradas em um log, que demonstra as consultas feitas, erros de consultas ao banco de dados e usuários que efetuaram acessos. A informação mais importante registrada neste log é a quando a

consulta feita ao sistema identifica um atendimento personalizado onde uma mensagem deverá ser enviada ao operador do sistema, conforme figura 5.6.

```
usuario: 0001
produto: 9788522009770
latitude: -30,0855166666667

longitude: -51,2457166666667
Atendimento :operador

=====
ALERTA:

Cliente com potencial de compras em livros presente no setor da loja.
Favor encaminhar atendimento pessoal
=====
```

Figura 5.6: Console de saída do servidor após a operação.

5.4 Bibliotecas utilizadas

Um dos objetivos da implementação de um protótipo é validar a viabilidade técnica de componentes que foram projetados, assim como identificar qual o melhor algoritmo ou biblioteca deverá ser utilizada para resolver determinado problema. Na modelagem do Atendente Móvel foi necessário pesquisar e escolher componentes para o reconhecimento do código de barras através da imagem e uma biblioteca de manipulação do receptor GPS do dispositivo. Estes componentes apenas foram adaptados à solução, sendo o mesmo implementado e registrado por terceiros.

5.4.1 Scanner de identificação do código de barras

Na pesquisa realizada para procurar soluções para o problema de reconhecimento de código de barras a partir de imagens foram encontradas diversas ferramentas e aplicações que solucionam o mesmo de maneira muito eficiente. Porém grande parte das soluções são proprietárias e de código não aberto, o que inviabilizaria a integração com o sistema que está sendo proposto. Entre as bibliotecas de código aberto que foram encontradas, a única que obteve um funcionamento adequado nos testes realizados foi a Zxing.

O Zxing (pronuncia-se “zebra crossing”) é um projeto open-source para processamento de imagens e identificação de múltiplos formatos de códigos 1D e 2D inicialmente desenvolvido em Java. O foco principal do projeto é decodificar o código de barras de fotos feitas pela câmera do dispositivo sem interação com servidores. Atualmente a biblioteca tem suporte para os seguintes formatos:

- UPC-A e UPC-E
- EAN-8 e EAN-13
- Code 39
- Code 93
- Code 128
- QR Code
- ITF

- Codabar
- RSS-14
- Data Matrix
- PDF 417

Este é um projeto mantido pelo Google, e tem como alvo o dispositivo Android, J2SE e Iphone, e por se tratar de projeto comunitário vários desenvolvedores portaram o código base da aplicação para outras plataformas, entre elas o C#. Esta iniciativa no ambiente Microsoft, não foi completamente validada e homologada, tendo vários problemas com a identificação de QR Code. Como este tipo de código de barras não é utilizado no Atendente Móvel este problema não inviabilizaria seu uso.

Na adequação da biblioteca à solução do protótipo (figura 5.7) foram removidos os fontes de leitura de códigos 2D, mas mantida a estrutura de leitura dos demais tipos mesmo sendo necessária a leitura do UPC/EAN.

Conforme demonstrado na modelagem do sistema (capítulo 4), classe que efetua a decodificação é a *MultiFormatReader* através do método *decode*, que recebe como parâmetro a imagem e uma lista de formatos possíveis. No modulo Scanner do protótipo foi efetuar duas tentativas de identificar o código de barras, fazendo com o que o método citado fosse acionado duas vezes. Isso foi necessário para que a leitura fosse feita prevendo o código de barras na posição horizontal e vertical, sendo que na segunda tentativa (que só ocorre caso nenhum código tenha sido identificado) é feita uma translação da imagem através de um método da própria biblioteca, a *BufferedImageMonochromeBitmapSource*, que é utilizada para adequar a imagem da classe *Bipmat* do framework .NET para o ambiente da mesma.

A adequação dos fontes da biblioteca para um projeto Windows Mobile foi feita de maneira simples, sendo necessário efetuar apenas ajustes nas referências de suas classes.

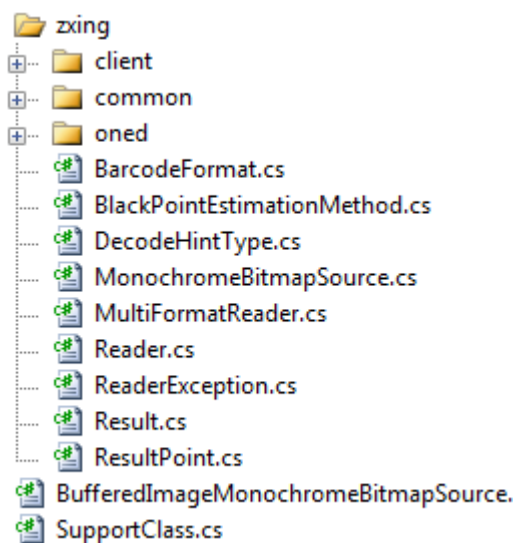


Figura 5.7: Estrutura dos fontes da solução Zxing.

5.4.2 Localização geográfica via GPS

O ambiente de desenvolvimento do .NET Framework oferece uma ampla variedade de para auxílio dos desenvolvedores, principalmente em documentação e bibliotecas.

Entre as facilidades oferecidas está o conjunto de “Samples”(exemplos de códigos) que são instalados junto Windows Mobile SDK (com o *namespace* de *Microsoft.WindowsMobile.Samples*). Juntamente com estes exemplos são disponibilizadas bibliotecas completas que utilizam recursos específicos de hardware ou rede dos dispositivos. Entre essas bibliotecas está o *GPS Intermediate Driver*, modulo que fornece recursos avançados de interação com o sistema de GPS.

O driver intermediário do GPS fornecido no sample é uma biblioteca que faz a interface entre a aplicação e o hardware de GPS do dispositivo. Esta interface foi codificada de maneira abstrata para que uma vez feito o desenvolvimento com a mesma, o software funcionará em diferentes dispositivos. A API proposta nesta biblioteca expõe as funcionalidades da biblioteca de código nativo do sistema operacional. As classes desta API ficam disponíveis no namespace *Location*.

O *Location* foi a classe utilizada no Atendente Móvel para efetuar a localização de geográfica do dispositivo do cliente. Os fontes deste projeto foram agregados integralmente a solução do sistema, com o intuito de não afetar o seu correto funcionamento, mesmo que nem todas as ferramentas que são oferecidas sejam utilizadas.

As coordenadas geográficas do usuário só serão necessárias no momento de efetuar a consulta, porém esta informação precisa ser obtida de forma rápida e segura para não diminuir a usabilidade da aplicação ou comprometer o seu funcionamento. Para isto é importante que o hardware de GPS seja inicializado de forma antecipada. No atendente Móvel isto é feito no momento em que a aplicação é iniciada com as seguintes instruções:

```
Gps gps = new Gps ();
if (!gps.Opened)
    gps.Open ();
```

Com o dispositivo instanciado e inicializado já possível fazer consultas a ele. Para isto é necessário fazer a chamada ao método *gps.GetPosition* que retorna um objeto da classe *GpsPosition*. Neste objeto os atributos que serão considerados são os seguintes:

- ***GpsPosition.Latitude***: atributo com a latitude atual no momento da consulta. Retorna um decimal(*double*), onde o norte é determinado por valores positivos.
- ***GpsPosition.LatitudeValid***: atributo que indica se o estado atual do GPS possui uma latitude valida. Deve ser sempre consultado antes da coordenada.
- ***GpsPosition.Longitude***: atributo com a longitude atual no momento da consulta. Retorna um decimal(*double*), onde o leste é determinado por valores positivos.
- ***GpsPosition.LongitudeValid***: atributo que indica se o estado atual do GPS possui uma longitude valida. Deve ser sempre consultado antes da coordenada.
- ***GpsPosition.SatelliteCount***: Indica o numero de satélites que estão sendo considerados na leitura dos valores. Caso o valor seja menor do que 4 a aplicação não dará continuidade na aplicação, informando ao usuário erro na procura de satélites GPS.

5.4.2.1 FakeGPS

As principais dificuldades de desenvolver aplicações que utilizam coordenadas do GPS são a execução dos testes e a simulação de dados. Utilizando um dispositivo

normalmente não temos como conseguir coordenadas diferentes daquela em que realmente estamos, e utilizando o emulador nem sempre é possível obter tais informações.

Para minimizar estes problemas no desenvolvimento do protótipo, foi utilizando o software *FakeGPS*, fornecido pela Microsoft juntamente com o Windows Mobile SDK. Este aplicativo simula coordenadas, respondendo para as aplicações como se fosse o próprio hardware de GPS. Quando o mesmo é habilitado, as coordenadas são lidas de um arquivo NMEA e enviadas ao driver GPS sem que seja necessário alterar a codificação da aplicação.

Arquivos NMEA são compostos de mensagens que seguem a especificação NMEA 0183 que define um padrão de dados para a comunicação entre dispositivos eletrônicos de muitos instrumentos (giroscópio, piloto automático, GPS receptores, etc.).

5.5 Detalhes da Implementação

Durante o projeto da aplicação, vários comportamentos foram assumidos como tecnicamente viáveis do ponto de vista de programação sem o devido detalhamento de como os mesmos seria implementados. Para a correta validação do modelo proposto foi necessário que estes componentes fossem desenvolvidos no protótipo. Nesta seção estes comportamentos serão tecnicamente detalhados.

5.5.1 Interface do Web Services

No capítulo 4 foram projetados para a camada de apresentação da aplicação do Atendente Server três serviços que seriam disponibilizados aos dispositivos. No protótipo somente um dos serviços foi implementado: a funcionalidade principal do sistema que é a Consulta de Produtos. O desenvolvimento deste web método gerou a seguinte especificação WSDL:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConsultaProduto xmlns="http://tempuri.org/">
      <_codProduto>string</_codProduto>
      <_codUsuario>int</_codUsuario>
      <_latitude>double</_latitude>
      <_longitude>double</_longitude>
    </ConsultaProduto>
  </soap12:Body>
</soap12:Envelope>
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConsultaProdutoResponse xmlns="http://tempuri.org/">
      <ConsultaProdutoResult>string</ConsultaProdutoResult>
    </ConsultaProdutoResponse>
  </soap12:Body>
```

```
</soap12:Envelope>
```

5.5.2 Modelo do XML de resultado da consulta

O web método *ConsultaProdutos* retorna para o dispositivo uma *string* (conforme visto na especificação WSDL), porém este retorno deverá ser tratado como um arquivo XML estruturado. A consulta deste serviço gera na camada de negócio da aplicação um objeto da classe *ResultadoInfo* com todas as informações que deverão ser retornadas ao cliente. Este objeto é convertido em arquivo XML momento em que sua serialização é feita, resultando no seguinte arquivo:

```
<?xml version="1.0" encoding="Windows-1252"?>
<ResultadoInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <prodInfo>
    <strCodigo>9788522009770</strCodigo>
    <urlImagem1>repositorio\imagem1</urlImagem1>
    <urlImagem2>repositorio\imagem1</urlImagem2>
    <intTipo>2</intTipo>
    <blnDisponivel>true</blnDisponivel>
    <intTipo_promocao>1</intTipo_promocao>
    <dblPreco>50.10</dblPreco>
    <strTitulo>A cabeça de Steve Jobs</strTitulo>
  </prodInfo>
  <strMsg>Cobrimos todos os preços de livros !</strMsg>
</ResultadoInfo>
```

Posteriormente o conteúdo deste arquivo é convertido em uma *string* que será retornada pelo serviço do Atendente Server. A interpretação desta fica a cargo do cliente, que deverá efetuar o processo contrário, gerando um objeto de *ResultadoInfo* a partir do XML recebido.

5.5.3 Verificação das coordenadas do cliente

Um dos pontos críticos para o funcionamento do sistema é a verificação feita no servidor, sobre as coordenadas enviadas pelo dispositivo. De acordo com a latitude e longitude enviada, o Atendente Móvel compara a localização geográfica do usuário com as áreas específicas registradas banco de dados com o objetivo de verificar se tal localização faz parte de um dos perímetros de interesse cadastrados no sistema.

Conforme definido no modelo de dados da aplicação, para cada área específica temos quatro pontos que formam um quadrilátero. Devemos assumir que estes quatro pontos estão ordenados e representados da seguinte forma:

- Ponto 1 (latitude_local1 e longitude_local1) – considerado como o ponto mais ao norte e mais ao oeste.
- Ponto 2 (latitude_local2 e longitude_local2) - considerado o segundo ponto mais ao oeste e mais ao sul do ponto 1.
- Ponto 3 (latitude_local3 e longitude _local3) – considerado o ponto mais ao leste.
- Ponto 4 (latitude_local4 e longitude _local4) - considerado o ponto mais o leste e mais ao sul do ponto 3.

Estando os quatro pontos cadastrados corretamente no banco de dados, teremos um perímetro em formato de um quadrilátero, composto por quatro linhas que ligam os pontos determinados conforme a figura 5.8.

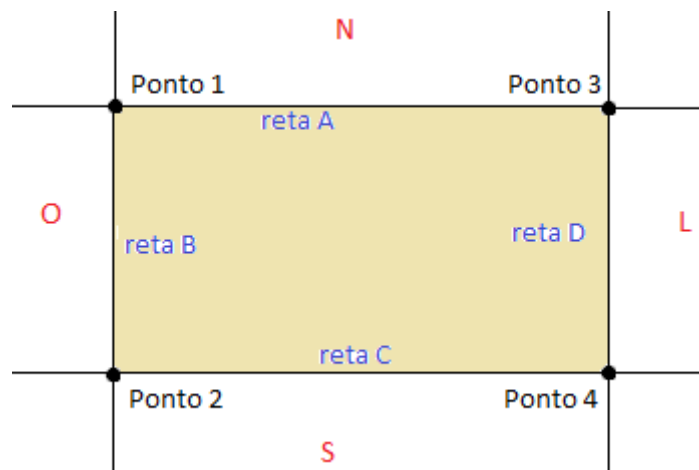


Figura 5.8: Plano de localização de uma área específica.

Mapeados os pontos de uma área específica cadastrada no sistema, devemos fazer os testes para identificar se um ponto está dentro do plano delimitado pelas retas A, B, C e D. As seguintes regras deverão ser validadas para cada uma das áreas cadastradas no sistema:

- O ponto cliente está abaixo da reta A ?
- O ponto cliente está à direita da reta B?
- O ponto cliente está acima da reta C?
- O ponto cliente está à esquerda da reta D?

Caso a resposta para estes quatro testes sejam positivos podemos afirmar que o cliente está na área específica cadastrada pelo operador para um determinado atendimento.

A comparação se um ponto está sendo feita considerando que as retas podem estar em níveis diferentes, ou seja, é necessário fazer a comparação do ponto com a função que defini a reta. Por exemplo, para uma situação com o Ponto 1 (latitude 1, e longitude 1) e Ponto 3 (latitude 5 e longitude 5). A função que define a reta A é calculada da seguinte forma:

- Calcula-se o coeficiente linear:
 - $\Delta \text{ longitude} / \Delta \text{ latitude} = (5-1)/(5-1) = 1$
- Utilizando um dos pontos obtemos a equação da reta no modo ponto-coeficiente usando um dos pontos (5,5):
 - $y - 5 = 1(x - 5)$, reduzida a forma:
 - $x - y = 0$
 - $x - y < 0$, define o plano abaixo da reta.
- Para comparar o ponto devemos substituir o y pela latitude e x pela longitude e verificar se a função é válida.

5.6 Testes

Os testes efetuados no protótipo tinham como objetivo validar apenas os principais pontos do sistema como a identificação do código de barras a partir de uma imagem, a conexão do dispositivo com o Web Services, a obtenção das coordenadas GPS do dispositivo, a manipulação do arquivo XML e navegação na interface do sistema.

Todos os testes foram executados em um emulador, o PTB Windows Mobile 6 Professional. Este emulador faz parte do conjunto de ferramentas de desenvolvimento fornecido no SDK utilizado.

5.6.1 Testes de reconhecimento do código de barras

Como os testes foram feitos em um emulador, não foi possível efetuar testes utilizando uma foto retirada pelo dispositivo que estava executando a aplicação. Para uma maior legitimidade dos testes as imagens utilizadas foram feitas em um smartphone com uma câmera de 3.2 megapixels, sem recursos avançados de controle de luminosidade e ajuste de foco, seguindo o padrão dos principais dispositivos no mercado.

Em geral a aplicação teve um comportamento satisfatório na identificação dos códigos, tendo dificuldade em produtos em que a superfície refletia a luminosidade e em imagens que o código de barras não estava em posição horizontal ou vertical conforme a figura 5.9. A aplicação também não obteve sucesso em fotos prejudicadas pela iluminação.



Figura 5.9: Exemplos de imagens não reconhecidas(escura e com reflexo, respectivamente)

5.6.2 Testes com a coleta de coordenadas

A parte da aplicação cliente foi testada utilizando o *FakeGPS*, com coordenadas inseridas manualmente no arquivo NMEA. Em todas as situações testadas o driver GPS respondeu corretamente para o sistema.

As coordenadas utilizadas nos testes, tanto para o arquivo NMEA quanto as que foram inseridas no banco de dados no cadastro das áreas específicas, foram obtidas utilizando a ferramenta Google Earth. Esta ferramenta possibilitou testes mais reais, uma vez que foram utilizadas coordenadas exatas de zonas conhecidas, sem que fosse necessária a presença de um receptor GPS no local.

5.6.3 Testes Gerais das aplicações Cliente e Servidor

Concluído o desenvolvimento do protótipo foram efetuados testes utilizando o sistema com as aplicações cliente e servidor para verificar o funcionamento geral dos mesmos. Os seguintes cenários de teste foram executados:

- Consulta a um produto cadastrado no sistema utilizando coordenadas que tenham um atendimento específico direcionado ao cliente. Comportamento esperado: aplicação cliente recebe mensagem na respectiva tela;
- Consulta a um produto cadastrado no sistema utilizando coordenadas que tenham um atendimento específico direcionado ao operador do sistema. Comportamento esperado: log do servidor exibe a mensagem direcionada ao especificada no atendimento, e o cliente exibe somente as informações do produto;
- Consulta a um produto não cadastrado no banco de dados. Comportamento esperado: aplicação cliente recebe mensagem do servidor informando que o mesmo não existe na base e exibe a mesma de forma amigável ao cliente;
- Tentativa de consulta utilizando um dispositivo sem acesso a internet. Comportamento esperado: Mensagem de erro (“Sem Rede de dados disponível”) é exibida;
- Tentativa de consulta utilizando um dispositivo sem hardware de GPS. Comportamento esperado: na tela de login a mensagem de erro é exibida (“GPS indisponível”).

Durante a execução dos cenários foi validada a navegabilidade das interfaces, a correta exibição dos dados e o desempenho das operações. Todos os cenários descritos foram executados com sucesso.

6 CONCLUSÃO

Neste capítulo, são apresentadas as conclusões sobre este trabalho com base nos resultados obtidos. Também são apresentados quais deverão ser os próximos passos na implantação do sistema em termos de ajustes e melhorias.

6.1 Resultados

O objetivo deste trabalho foi alcançado com resultado satisfatório, pois foi possível concluir que o sistema, que foi proposto, é viável, e pode se tornar um software de uso comercial. Implementadas todas as funcionalidades que foram projetadas para o sistema, o mesmo pode se tornar uma ferramenta extremamente útil do ponto de vista de vendas para empresa que adquiri-la.

Também podemos concluir, através do protótipo, que o sistema se mostrou bastante atrativo do ponto de vista do usuário. Utilizando recursos tecnológicos que estão se tornando cada vez mais populares, como GPS e câmera, além do próprio smartphone, é possível efetuar consulta a produtos variados e visualizar informações de grande serventia para os consumidores.

Do ponto de vista de técnico, o desenvolvimento deste sistema se mostrou bastante desafiador. Foi necessário um grande esforço em pesquisa para escolher as ferramentas adequadas, a melhor plataforma de desenvolvimento e quais deveriam ser as bibliotecas que seriam utilizadas. A definição da arquitetura e do projeto das classes também foi feito de maneira que o sistema possa ser portado para diferentes modelos e sistemas operacionais de dispositivos uma vez que foram descritos todos os componentes de modo genérico.

Do ponto de vista criativo, a proposta da aplicação também se mostrou bastante inovadora, já que, até a presente data, não foram encontradas aplicações que ofereçam as mesmas funcionalidades.

6.2 Implantação do sistema

A futura utilização do sistema por uma empresa que adquirir a ferramenta requererá um processo de ajustes. Durante o projeto do sistema, várias definições foram feitas considerando que, o futuro cliente do software, já tem uma sólida base de dados. Os módulos de acesso a banco deverão ser modificados para atender o novo modelo, assim como as interfaces deverão ganhar um layout personalizado com os devidos logotipos.

6.3 Melhorias e novas funcionalidades

Implementada e implantada uma primeira versão do sistema, existe uma série de melhorias que poderão ser feitas sobre as funcionalidades atuais e também em termos de novas funcionalidades.

Entre as funcionalidades que estão previstas neste trabalho, a identificação do código de barras é a que pode mais ganhar ajustes que agregariam mais valor à mesma. Hoje existem ferramentas proprietárias que efetuam a leitura do código de maneira muito eficiente, com ajuste automático de foco e luz, sem a necessidade de capturar a imagem, apenas com o direcionamento da câmera. Essas melhorias poderiam ser agregadas ao Atendente Móvel.

Existem também funcionalidades novas que poderiam proporcionar ganhos à empresa que implantar o sistema. A principal delas seria a possibilidade do cliente efetuar a compra do produto pelo próprio dispositivo. Para isto seria necessário ampliar as questões de segurança do aplicativo e desenvolver novos módulos na aplicação cliente para facilitar as formas de pagamento (leitura do cartão de crédito, confirmação via SMS, etc.). Também poderá ser disponibilizada no futuro, ao usuário, a possibilidade de comprar mídias digitais, como músicas em formato digital e livros em formatos e-book para leitura no próprio smartphone.

Por fim, a portabilidade da aplicação para diferentes plataformas seria fundamental para o sucesso do produto. Atualmente não há um modelo específico que domine o mercado de smartphones, e cada dia novos modelos e novos sistemas surgem. Sendo assim a implementação de versões para outros sistemas, além do Windows Mobile, facilitaria a aceitação do serviço entre os consumidores.

REFERÊNCIAS

WIGLEY, A. ; MOTH D. ; FOOT P. ; **Microsoft Mobile Development Handbook**, MICROSOFT PRESS, 2007

CANALYS. **Global Smartfones Sales in Q2 2009**. Disponível em <<http://www.canalys.com>>. Acesso em janeiro de 2010.

MONICO, JOÃO FRANCISCO GALERA; **NAVSTAR-GPS: descrição, fundamentos e aplicações**, 2000.

MOURA, BENJAMIM DO CARMO; **Logística: conceitos e tendências**. Vila Nova de Famalicão: Edições Centro Atlântico, 2006.

GS1 Brasil. **Código de Barras. São Paulo**. Disponível em: <<http://www.gs1brasil.org.br>>. Acesso em junho de 2010.

MORIMOTO, CARLOS E.; **Guia prático do Smartphone**; GDH Press e Sul Editores 2009

US COAST GUARD; **NAVSTAR GPS User Equipment Introduction**, 2006.

Webservices, SOAP, WSDL; **Web Services, SOAP e Aplicações Web**, Disponível em <<http://devedge-temp.mozilla.org>>. Acesso em janeiro de 2010.

MICROSOFT ; **Mobile Developer**, Disponível em <<http://msdn.microsoft.com/pt-br/windowsmobile/>>. Acesso em junho de 2010

ZXING : **zxing, Multi-format 1D/2D barcode image processing library with clients for Android, Java** < <http://code.google.com/p/zxing/>. Acesso em junho de 2009.

TECHNET: **Arquitetura do ASP.NET**< <http://technet.microsoft.com/pt-br/>>. Acesso em 2010.