

**Universidade Tecnológica Federal do Paraná
Programa de Pós – Graduação em Tecnologia
Especialização em Tecnologia Java**

Anderson Luis Canalle

Empregando tecnologia Java, Android e Geoprocessamento em aplicativos móveis

Monografia de Especialização

Curitiba
2011

Anderson Luis Canalle

Empregando tecnologia Java, Android e Geoprocessamento em aplicativos móveis

Monografia apresentada ao Programa de Pós-Graduação em Tecnologia Java da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Especialista em Tecnologia Java.

Orientador:

Prof. Gustavo Alberto Gimenez Lugo

Curitiba
2011

Sumário

Sumário	3
Lista de Figuras	5
Lista de Abreviaturas	6
Lista de Abreviaturas	6
Resumo	7
1. Introdução	8
1.1 Motivação	8
1.2 Objetivo	8
1.3 Objetivos Específicos	8
1.4 Justificativa	8
1.5 Estrutura de trabalho	9
2. Conceitos	10
2.1 Geoprocessamento	10
2.1.1 Sensoriamento remoto	10
2.1.2 Sistema de posicionamento global (GPS)	11
2.1.3 Cartografia digital	11
2.1.4 Topografia e levantamento de campo	11
2.1.5 Processamento digital de imagens	12
2.1.6 Sistemas de informações geográficas	12
2.2. Android	13
2.2.1 Open Handset Alliance	13
2.2.2 Sistema operacional Linux	14
2.2.3 Máquina Virtual Dalvik	14
2.2.4 API e ferramentas	14
2.2.5 Serviço Web	20
2.2.6. PostgreSQL	21
3. Protótipo	23
3.1 Lista de Requisitos	23
3.2 Use Cases	24
3.2.1 Login	24
3.2.2 Pesquisa	25
3.2.3 Mapa	26
3.3 Arquitetura	27
3.4 Diagramas de classes	29
3.4.1 Persistência	29
3.4.2 Webservice	30
3.4.3 Cliente Android	31
3.5 Diagrama de sequência	32
3.5.1 Login Cliente Android	32
3.5.2. Login WebServices	33
3.5.4 Pesquisar Local	34
3.6 Diagrama de Entidade e Relacionamento	36
4. Experimento	37

4.1 Resultados obtidos	42
Conclusões	44
Trabalhos Futuros	44
Referências Bibliográficas	45

Lista de Figuras

Figura 1 - Emulador Android (Alliance)	15
Figura 2 - Página de geração de API Key.....	16
Figura 3 - Arquitetura Web Services (Marzullo, 2009).....	20
Figura 4 - Validar Login	24
Figura 5 - Pesquisar Locais.....	25
Figura 6 - Exibir Mapa.....	26
Figura 7 - Arquitetura do Protótipo	27
Figura 8 - Arquitetura Cliente Servidor (Machado, 2010)	28
Figura 9 - Diagrama de classe camada persistência.....	30
Figura 10 - Diagrama de classe camada web services.....	30
Figura 11 - Diagrama de classe cliente android.....	32
Figura 12 - Diagrama de seqüência login - Cliente Android.....	33
Figura 13 - Diagrama de seqüência login – WebServices	33
Figura 14 - Diagrama de seqüência Pesquisar local - Cliente Android	34
Figura 15 - Diagrama de seqüência Pesquisar local - Web Services.....	35
Figura 16 - Diagrama de entidade e relacionamento	36
Figura 17 - Tela de login.....	37
Figura 18 - Tela de pesquisa	38
Figura 19 - Lista de resultado	39
Figura 20 - Mapa.....	40
Figura 21- Menu do mapa.....	41
Figura 22 -Visualização de satélite	42

Lista de Abreviaturas

Abreviatura	Nome por extenso
SIG	Sistemas de Informação Geográfica
API	Application Programming Interface (Interface de Programação de Aplicativos)
GPS	Global Positioning System (Sistema de Posicionamento Global)
HTTP	HyperText Transfer Protocol
IP	Internet Protocol (Protocolo de Internet)
JVM	Java Virtual Machine (Maquina Virtual Java)
MIME	Multipurpose Internet Mail Extensions (Extensões Multi função para Mensagens de Internet)
OHA	Open Handset Alliance
RPC	Remote procedure call (Chamada de procedimento remoto)
SAGA	Sistema de Análise Geo-Ambiental
SGBDOR	Sistema gerenciador de banco de dados
SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol (Protocolo de Controle de Transmissão)
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modeling Language (Linguagem Unificada de Modelagem)
URI	Uniform Resource Identifier (Identificador Uniforme de Recursos)
W3C	World Wide Web Consortiun
WSDL	WebService Description Language
XML	Extensible Markup Language

Resumo

Esta monografia tem por objetivo apresentar um estudo sobre a utilização da plataforma Android para o desenvolvimento de aplicativos utilizando algumas características de sistemas de geoprocessamento de informação, para dispositivos móveis.

Com a evolução apresentada pelos sistemas de informação, disponibilizando importantes recursos, os quais foram aplicados na disciplina de geoprocessamento, possibilitando a evolução de mapeamento e gerenciamento de informações georreferenciadas, foi possível dar origem aos SIGS (Sistemas de Informação Geográfica), sistemas que propiciaram um avanço na produção de artefatos na área de estudos geográficos. Atualmente devido ao poder de processamento que os celulares possuem, os mesmos estão recebendo foco para utilização de aplicações SIG moveis, disponibilizando aos usuários varias facilidades com sua utilização.

Visualizando o potencial de mercado que os dispositivos móveis disponibilizaram no atual momento, o Google em conjunto com o Open Handset Alliance disponibilizou a plataforma Android, a qual utiliza tecnologia Java combinada com Sistema Operacional Linux, oferecendo uma série de funcionalidades e permitindo o desenvolvimento de aplicações com segurança e flexibilidade para explorar e combinar os recursos da plataforma com o poderio dos dispositivos móveis atuais.

Neste estudo serão descritos os conceitos das tecnologias utilizadas em combinação com a plataforma Android para construir um protótipo que apresente algumas funcionalidades que um GIS móvel poderia oferecer no mercado. Além disso, serão apresentadas informações sobre a arquitetura, recursos e dificuldades encontradas e aplicadas durante o experimento da utilização desta plataforma com este objetivo.

Palavras-chave: Java, Plataforma Android, GeoProcessamento

1. Introdução

1.1 Motivação

Ao longo do tempo os sistemas de informação evoluíram a tal ponto que disponibilizaram importantes recursos, os quais foram aplicados aos estudos de mapeamento e gerenciamento de posições geográficas, dando origem aos SIGS (Sistemas de Informação Geográfica), sistemas que propiciaram um avanço na produção de artefatos na área de estudos geográficos. Atualmente devido ao poder de processamento que os celulares possuem, os mesmos estão recebendo foco para utilização de aplicações SIG moveis, disponibilizando aos usuários varias facilidades com sua utilização.

O Google utilizando tecnologia Java combinada com Sistema Operacional Linux disponibilizou a plataforma Android, a qual oferece uma série de funcionalidades, permitindo o desenvolvimento com segurança e flexibilidade para explorar e combinar os recursos da plataforma com o poderio dos dispositivos móveis atuais.

1.2 Objetivo

Este estudo tem como objetivo geral explorar as condições associadas à construção de aplicações SIG móveis utilizando a plataforma Android.

1.3 Objetivos Específicos

Os objetivos específicos deste estudo são propor uma arquitetura para implementação de um SIG móvel, utilizando a plataforma Android, e também analisar a arquitetura proposta, além de apresentar alguns dos conceitos utilizados na construção do protótipo utilizado como experimento.

1.4 Justificativa

Devido à grande popularidade dos celulares, os quais deixaram a muito tempo de executarem apenas a função de transmissão de voz, abrindo espaço para serviços de transmissão de dados, propiciam um vasto campo para disponibilização de serviços baseados em SIG aos usuários da telefonia celular. Os serviços baseados em SIG podem oferecer desde funcionalidades de localização de endereços até funcionalidades combinadas a outros serviços disponibilizados na web, tais como redes sociais, e-commerce, games, etc.

Visualizando o potencial deste nicho combinado com as facilidades disponibilizadas pela plataforma Android, este estudo irá propor uma arquitetura para implementação de um serviço baseado nos conceitos de um SIG e posteriormente efetuar um experimento utilizando a arquitetura proposta para construir um protótipo que possua características de um SIG.

1.5 Estrutura de trabalho

Este documento está organizado da seguinte maneira, no **capítulo 2** serão apresentados os conceitos das tecnologias empregadas na implementação de um protótipo de um SIG móvel, o qual terá sua arquitetura e funcionalidades especificadas no **capítulo 3**.

O **capítulo 4** apresentará um passo a passo da utilização do protótipo do SIG móvel implementado, além de descrever suas características, e por fim no **capítulo 5** é realizada uma avaliação quanto à viabilidade de utilização da arquitetura proposta e as tecnologias envolvidas.

2. Conceitos

Este capítulo apresenta os conceitos e tecnologias utilizadas para construção do protótipo que será alvo de estudo neste trabalho.

2.1 Geoprocessamento

Geoprocessamento é um ramo da ciência que estuda e propicia técnicas de processamento de informações georreferenciadas lançando mão de aplicativos desenvolvidos com este objetivo, tais aplicativos são conhecidos como SIG (Sistemas de informações geográficas) (Piroli, 2010), além de equipamentos, tais como aparelho de GPS (Sistema de Posicionamento Global) (Huerta, 2005), computadores e periféricos.

O geoprocessamento permite manipular as informações georreferenciadas, avaliá-las e por fim gerar produtos de cunho cartográfico ou estatístico referente a localizações de informações sobre superfície terrestre.

O geoprocessamento não existiria sem a utilização de tecnologias como softwares, computadores e dispositivos móveis, os quais permitiram o desenvolvimento das geotecnologias. Estes equipamentos permitem a coleta, transmissão de informações via rede de um ponto a outro, desde que estejam conectados por algum tipo de rede de comunicação. Tal característica permite a realização de análises geográficas de grandes volumes destes dados, tarefa a qual despenderia uma enorme quantidade de tempo para executá-la de forma manual.

Para melhor entender a estrutura desta ciência (Gilberto Câmara, 2006), podemos dividi-la nas seguintes partes:

2.1.1 Sensoriamento remoto

Conforme (Piroli, 2010) sensoriamento remoto é definido como a aquisição de informação de determinada característica de um objeto ou fenômeno, por um dispositivo de registro que não possua contato com o objeto de estudo.

O sensoriamento remoto é dividido em duas formas:

Orbital: Quando a coleta de informações é realizada por sensores localizados em órbitas ao redor do planeta, tendo como alvo de estudo a superfície do planeta. As imagens de satélites são um exemplo deste tipo de sensoriamento.

Sub-Orbital: Tendo como mesmo objetivo que o sensoriamento orbital, porém utiliza equipamentos aerotransportados, não localizados em órbitas. Um exemplo deste tipo de sensoriamento seriam as fotografias aéreas.

2.1.2 Sistema de posicionamento global (GPS)

Sistema de posicionamento global foi criado pelo departamento de defesa dos Estados Unidos da América na década de 1970 (Huerta, 2005), com finalidade de uso militar, sendo liberado para utilização civil a partir do ano de 2000. O sistema é constituído por uma constelação de 28 satélites sendo 4 sobressalentes em 6 planos orbitais que ficam a 20.200 km de altitude, passando sobre o mesmo ponto da terra duas vezes por dia.

O sistema funciona da seguinte forma, os satélites emitem sinais de rádio que são captados pelos aparelhos receptores GPS, os quais utilizam sinais de no mínimo 4 satélites e calculam a distância entre a recepção e cada um dos satélites, pelo intervalo de tempo entre o instante local e o instante em que os sinais foram enviados, esta distância é chamada pseudodistância, desta forma é possível efetuar a triangulação das posições e identificar a coordenada geográfica em que o aparelho receptor se encontra (Huerta, 2005).

A característica de fácil identificação da posição geográfica em que o usuário do aparelho GPS se encontra, permitiu avanços significativos em relação à forma de mapeamento da superfície terrestre, visto que possibilitou a automatização da coleta das informações.

2.1.3 Cartografia digital

Digitalização de mapas, cartas topográficas ou fotos aéreas, que estão disponíveis em formato analógico (em papel).

Estes materiais podem ser digitalizados utilizando-se scanners, buscando-se evitar resoluções baixa o que pode comprometer a qualidade das informações. Desta forma sempre é realizado um estudo sobre a resolução adequado ao material utilizado.

2.1.4 Topografia e levantamento de campo

Segundo (Ferreira, 1985) topografia é a “arte de representar no papel a configuração de uma porção de terreno com todos os acidentes e objetos que se acham a sua superfície” Embora atualmente existam imagens de satélite de alta resolução, o custo e a dificuldade em obtê-las, fazem com que a técnica de topográfica seja a solução viável, pois esta técnica permite o detalhamento com a qualidade requerida, principalmente em pequenas áreas. Um exemplo disso é o levantamento topográfico em áreas urbanas.

2.1.5 Processamento digital de imagens

Técnica em que se efetua adaptações e tratamentos necessários para modificar uma imagem com a finalidade de ajuste necessário a uma determinada finalidade de trabalho. Geralmente este trabalho consiste em composições de bandas de imagens de satélite, correções atmosféricas, aplicações de filtros e de contraste, classificação de imagens, entre outros.

2.1.6 Sistemas de informações geográficas

Os SIGs são sistemas computacionais desenvolvidos para aplicação no processamento de dados georreferenciados e coordenadas espaciais.

Rocha (Piroli, 2010) (Rocha, 2007) define SIG como

“Um sistema com capacidade para aquisição, armazenamento, tratamento, integração, processamento, recuperação, transformação, manipulação, modelagem, atualização, análise e exibição de informações digitais georreferenciadas, topologicamente estruturadas, associadas ou não a um banco de dados alfanumérico.”

Os SIGs tiveram sua origem na década de 50, quando foi iniciado o processo de automatização da produção de mapas na Inglaterra, este trabalho tinha como objetivo representar informações sobre a botânica, enquanto nos Estados Unidos, era utilizado no auxílio da organização do tráfego da cidade. A primeira utilização por um centro de pesquisa foi em 1960, na Universidade de Washington, sendo que em 1964 foi desenvolvido o primeiro SIG para o inventário das terras do Canadá.

No Brasil, o precursor do uso dos SIGs foi o SAGA (Sistema de Análise Geo-Ambiental) na década de 1980. Atualmente temos diversos SIGs no mercado, sendo muitos de distribuição gratuita, como o caso do SPRING, além do caso de softwares híbridos como o Google Earth, o qual possibilita a consulta de informações via web.

Como os SIGs possuem a característica de relacionar uma determinada informação com sua localização espacial, estes aplicativos permitem a manipulação das informações georreferenciadas, assim como seus atributos, além de realizar operações de análise geográfica integrando as informações processadas.

Conforme (Piroli, 2010) os SIGs possuem três aplicações fundamentais na área geográfica

- 1 – Podem ser usados como ferramenta para produção de mapas, e ainda para geração e visualização de dados espaciais;*
- 2 – Podem ser usados como suporte para análise espacial de fenômenos e para a combinação de informações espaciais, e;*
- 3 – Podem ser usados como bancos de dados geográficos, que tem funções de armazenamento e recuperação de informações espaciais.”*

2.2. Android

O mercado de celulares atualmente cresce de forma expressiva, sendo que cerca da metade da população mundial possui um aparelho. Estes usuários procuram cada vez mais aparelhos que disponibilizem recursos tais como players de músicas, câmeras, Bluetooth, GPS, acesso a internet, etc. Um grupo dentre estes usuários tem um perfil corporativo, o qual também possui suas necessidades em relação às aplicações móveis, tais como aplicativos que agilizem seus negócios, integrando com seus sistemas de back-end, aplicando dessa forma o conceito de mobilidade em seus negócios (Lecheta, 2010).

Desenvolvedores e empresas buscam uma plataforma ágil e moderna para o desenvolvimento dos aplicativos “desejados“ e necessários para preencher as necessidades atuais do mercado. O Google visualizando esta necessidade apresentou como resposta a plataforma de desenvolvimento para aplicativos móveis Android, a qual é baseada no sistema operacional Linux, sendo um ambiente de desenvolvimento robusto e flexível.

2.2.1 Open Handset Alliance

A OHA - Open Handset Alliance é um grupo formado por grandes empresas que atuam no mercado de telefonia de celulares, sendo liderados pelo Google. Os principais integrantes do grupo até o momento em que foi escrito este estudo são: HTC, LG, Motorola, Samsung, Sony Ericsson, Toshiba, Sprint Nextel, China Mobile, T-Mobile, ASUS, Intel e Garmin.

O objetivo deste grupo é padronizar uma plataforma única e aberta para celulares para aumentar a satisfação dos consumidores em relação ao produto final. O objetivo secundário da aliança é criar e manter uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas (Lecheta, 2010).

O Android foi o resultado do trabalho da OHA, tendo como principais características um sistema operacional baseado em Linux, interface visual rica, GPS, diversas aplicações já instaladas além de um ambiente de desenvolvimento robusto e flexível. Outro ponto importante desta plataforma é a possibilidade da utilização da linguagem Java para o desenvolvimento de aplicações.

A plataforma Android é de código aberto como o Linux, o que contribui para o seu aperfeiçoamento, pois desenvolvedores de todo mundo podem contribuir para o código-fonte, corrigindo falhas ou contribuindo com novas funcionalidades.

2.2.2 Sistema operacional Linux

O sistema operacional do Android foi baseado no kernel 2.6 do Linux, o qual é responsável pelo gerenciamento da memória, threads, processos, segurança dos arquivos, redes e drivers.

Como no Linux cada aplicação executada no Android gera um novo processo no sistema operacional, podendo ser exibidos em tela ou executados em segundo plano. O sistema operacional permite a execução de diversos processos simultaneamente, além de decidir encerrar determinado processo para liberar memória e recursos.

Cada aplicação é executada em um processo diferente, o qual possui sua própria thread. Esta característica foi herdada da segurança do Linux, assim como a regra de instalação dos aplicativos, que dita que para cada aplicação instalada é criado um usuário no sistema operacional para ter acesso a sua estrutura de pastas, evitando dessa forma que outro usuário acesse as pastas de outro usuário.

2.2.3 Máquina Virtual Dalvik

A plataforma Android possibilita a utilização da linguagem Java para construir suas aplicações. O sistema operacional possui uma máquina virtual Java (JVM) otimizada para execução em dispositivos móveis, conhecida como Dalvik (Rick Rogers, 2009).

Após o desenvolvedor criar uma aplicação utilizando linguagem Java, o código fonte “.java” é compilado em “.class”(bytecode), o qual é convertido para o formato “.dex” (Dalvik Executable). Este formato representa a aplicação Android compilada. Para ser distribuído e instalado, o sistema compacta em um único arquivo com a extensão “.apk” (Android Package File), os arquivos “.dex” e outros recursos como imagens (Lecheta, 2010).

2.2.4 API e ferramentas

Uma das funcionalidades mais interessantes que a plataforma Android disponibilizou no mercado foi a API (Interface de programação de aplicações) para manipulação de Mapas e GPS. Esta API disponibiliza recursos para construção de mapas, tais como controle de zoom, tipo de visualização por rua e satélite, exibição de imagens sobre o mapa. Funcionalidades que evitam todo o trabalho de execução das tarefas de levantamento de materiais necessários para aplicações que utilizam o georreferenciamento, tais como sensoriamento remoto, cartografia digital, topografia e processamentos de imagens, permitindo ao usuário focar na solução de negócio.

Para utilização das APIs do Android é necessária a instalação da SDK (Standard Developer Kit) que é o software utilizado para desenvolver aplicações para o Android,

contendo a API completa para a linguagem Java, emulador para simular o celular e ferramentas utilitárias

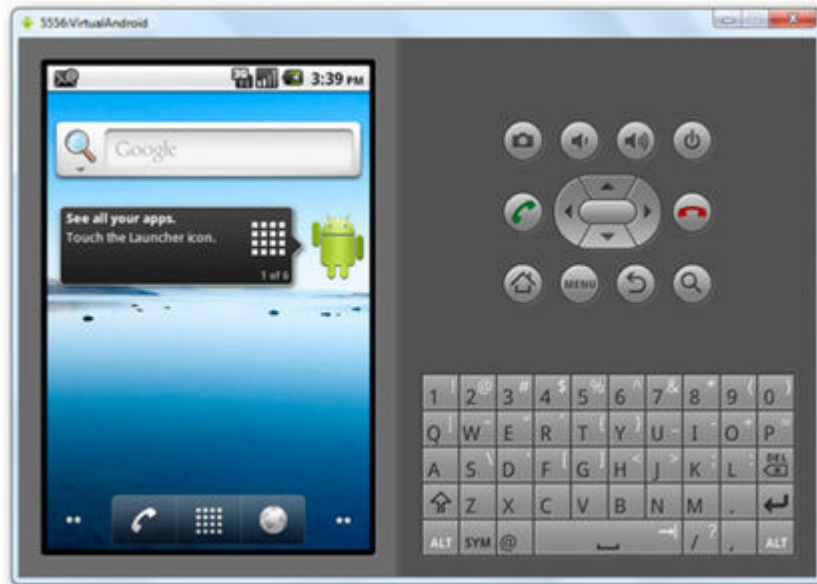


Figura 1 - Emulador Android (Alliance)

Abaixo segue lista das APIs android disponíveis até o momento da escrita deste trabalho.

Versão	Nome	Nível
Android 2.3	Gingerbrad	9
Android 2.2	FroYo	8
Android 2.1	Eclair	7
Android 1.6	Donut	4
Android 1.5	Cupcake	3
Android 1.1	Petit Four	2
Android 1.0	Não batizado	1

Tabela 1 - Versões Android (Alliance)

Além do SDK é necessário utilizar o ADT(Android Development Tools), um plugin para a IDE Eclipse disponibilizado pela Google, o qual facilita o desenvolvimento de aplicações Android, pois estende ao Eclipse as funcionalidades da SDK Android.

API Key

O Google controla o acesso a funcionalidade Google Maps, serviço que é acessado pelos aplicativos que utilizam mapas. Desta forma é necessário obter uma chave para utilizar esse serviço (Alliance, ; Lecheta, 2010).

Como requisito para a solicitação da chave é necessário fornecer o código do certificado digital utilizado para assinar a aplicação desenvolvida. O desenvolvedor que utilizar o SDK e o plugin do Eclipse para desenvolvimento de aplicações Android terá

automaticamente seu aplicativo assinado por um certificado digital no momento da compilação. A chave da assinatura será encontrado no certificado digital debug.keystore

No site <http://code.google.com/android/maps-api-signup.html> é disponibilizado o gerador da chave de certificação da Google, bastando apenas informar a chave de assinatura do seu certificado e pressionar o botão Generate API Key e você terá sua chave para utilização de mapas.

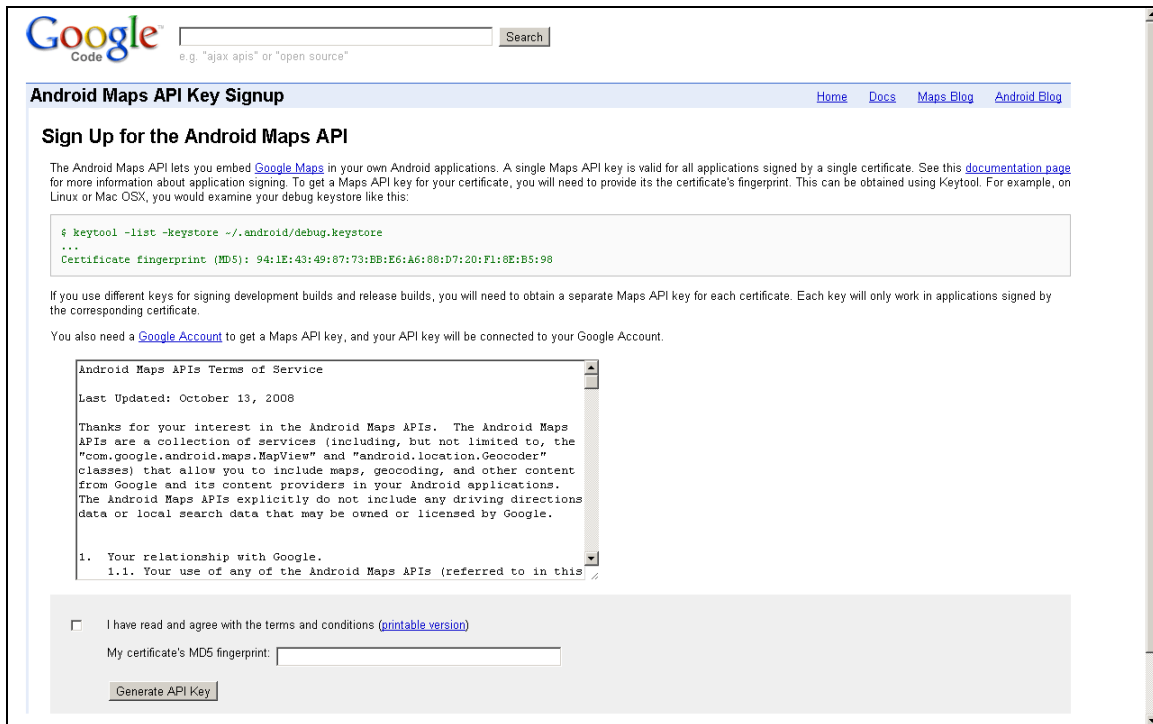


Figura 2 - Página de geração de API Key

A chave gerada deverá ser informada como parâmetro no construtor da Classe MapView, a qual gerencia a apresentação do mapa, a qual será descrita nos próximos parágrafos.

```
MapView mapView = new MapView(this,"03Rjqj69xif3JQBtubSzXfOrAke2FBpXGIg4LFw");
```

Segurança do Android

A plataforma Android possui um controle de segurança que impede que a aplicação instalada no dispositivo móvel acesse recursos sem que sejam autorizados previamente. Este acordo é realizado através do arquivo AndroidManifest.xml o qual declara quais recursos e telas serão acessadas e exibidas pelo aplicativo instalado, evitando dessa forma a solicitação de recursos desnecessário, além de acesso a funcionalidades não pertinentes a aplicação.

Abaixo segue um exemplo da estrutura de um arquivo AndroidManifest.xml, onde é declarado que aplicação utilizará recursos de acesso a internet, mapas e GPS.


```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.com.android.gis"
    android:versionCode="1"
    android:versionName="1.0">

    <application android:icon="@drawable/icon" android:label="@string/app_name">

        <!-- Pacote de mapas -->
        <uses-library android:name="com.google.android.maps" />

        <!--Local para inserção das activity -- >

    </application>
    <!--Permissões de Internet e GPS -->
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

</manifest>

```

Classe MapView

A classe MapView do pacote com.google.android.maps.MapActivity, encapsula todo o código necessário para realizar a comunicação com o GoogleMaps, por este motivo que esta classe recebe como parâmetro no seu construtor a chave de licença para utilização do serviço do GoogleMaps.

Seguem abaixo os principais métodos utilizados na implementação do protótipo, disponibilizados pela Classe MapView

```
MapView mapView = new MapView(this,"03Rjqj69xif3JQBtubSzXfOrAke2FBpXGIg4LFw");
```

Método	Descrição
setStreetView(boolean n)	Controla se o mapa disponibilizará a visualização das ruas no mapa
setSatellite(boolean arg)	Habilita a visualização de satélite no mapa, ou seja a área visualizada no mapa é representada por fotos de satélite. Caso esteja desabilitado o mapa representa a área visualizada através de desenho, utilizando padrão Google.
getController()	Retorna um objeto MapController, o qual permite executar controle de zoom e navegação no mapa

Classe GeoPoint

Classe que representa uma localização, a qual é composta por um par de coordenadas (latitude e longitude), sendo representadas por microdegrees.

A notação microdegrees se refere a um grau multiplicado por 1E6 e apenas numérico, sem decimais. O número 1E6 é a mesma coisa que 1.000.000, onde o 6 significa que existem 6 zeros a esquerda.

Exemplo de conversão:

```
int latitude = (int) (latitude*1E6);
```

Classe MapController

Esta classe pertence ao pacote com.google.android.maps e disponibiliza recurso para controlar a navegação pelo mapa, além do controle de detalhamento. Seguem abaixo os principais métodos utilizados na implementação do protótipo, disponibilizados pela Classe MapView

Método	Descrição
zoomIn()	Efetua ação de aumentar a visualização de detalhe da área centralizada do mapa
zoomOut()	Efetua ação de diminuir a visualização de detalhe da área centralizada do mapa
setZoom(int ZooLevel)	Seta o nível de detalhe de visualização do mapa
setCenter(GeoPoint point)	Centraliza a visualização do mapa na coordenada informada

Classe Overlay

Esta classe possibilita a inclusão de desenho no mapa, sendo freqüentemente utilizado para exibir pontos no mapa, tais como os utilizados para exibir o ponto A e ponto B quando é traçado uma rota no mapa.

Segue abaixo principal método utilizado desta classe:

Método	Descrição
Draw(android.graphics.Canvas canvas, MapView mapView, boolean shadow)	Desenha no ponto informado do mapa

Recuperando a posição do usuário

A plataforma Android dispondo da API para utilização de GPS permite identificar a posição do usuário e conseqüentemente permite traçar rotas para o destino desejado. Isso funciona de forma fantástica, mas tem como custo o alto consumo da bateria. Desta

forma a API da Google oferece facilidades para localização do usuário através de Network Location Provider (Provedor de localização na rede), ou seja, permite localizar o usuário utilizando a triangulação de antenas de celular ou sinal de Wi-Fi. Esta técnica possui resposta rápida, diminui o custo de utilização da bateria, além de funcionar em locais abertos e fechados.

Para informar qual modo o seu aplicativo deve utilizar para identificar a posição atual do usuário basta apenas utilizar as constantes `LocationManager.GPS_PROVIDER` ou `LocationManager.NETWORK_PROVIDER`, como parâmetro no método `requestLocationUpdates`, o qual é descrito nos próximos parágrafos.

Para utilizar funcionalidade de geo-localização, basta criar uma classe que implementa a interface `android.location.LocationListener`, que provê vários métodos interessantes, mas o principal é o método `onLocationChanged(location)`, o qual é chamado automaticamente pelo Android sempre que a localização GPS for alterada.

Ex:

```
LocationListener locationListener = new LocationListener() {  
  
    public void onLocationChanged(Location location) {  
  
        Log.i("GPS", "latitude: "+location.getLatitude()+" , longitude:  
        "+location.getLongitude());  
        loc = location;  
    }  
  
    public void onStatusChanged(String provider, int status, Bundle extras){}  
  
    public void onProviderEnabled(String provider){}  
  
    public void onProviderDisabled(String provider){}  
  
};
```

Segue abaixo exemplo da utilização da classe `LocationManager` e `LocationListener`, além descrição dos parâmetros necessários para a utilização do método de atualização do posicionamento do usuário:

```
LocationManager locationManager = (LocationManager)  
getSystemService(Context.LOCATION_SERVICE);  
  
locationManager.requestLocationUpdates(LocationManager.GPS_Provider,  
tempoMinimo, distanciaMinima, locationListener)
```

Parâmetro	Descrição
String Provider	Identificador do provedor de localização, sendo uma string por padrão. Ex: "GPS"
Long tempoMinimo	Tempo em milisegundos, que representa o intervalo mínimo de atualização da localização
Float distanciaMinima	Distancia em metros, que representa a

	distancia mínima a ser percorrida para efetuar atualização da posição atual.
LocationListener locationListener	Implementação da interface LocationListener, a qual receberá as atualizações das coordenadas pelo método onLocationUpdate(location)

2.2.5 Serviço Web

Serviço Web (WebService) é a disponibilização de um serviço na internet, permitindo que um cliente possa acessar este serviço de qualquer local do planeta. Conforme (Marzullo, 2009) *“Este tipo de serviço representa a lógica de negócio, o qual permite um ou mais clientes enviarem requisições de um tipo bem definido de informação e recebam respostas síncronas ou assíncronas”*.

Um Webservice possui uma estrutura interna fracamente acoplada, suporta chamadas remotas de procedimentos (RPC), permitindo o usuário utilizar operações de objetos remotos utilizando um protocolo baseado em XML, além de cada operação ter a opção de expor parâmetros de entrada e saída e definir o tipo de retorno, de forma a orientar ao cliente sobre como utilizar o serviço.

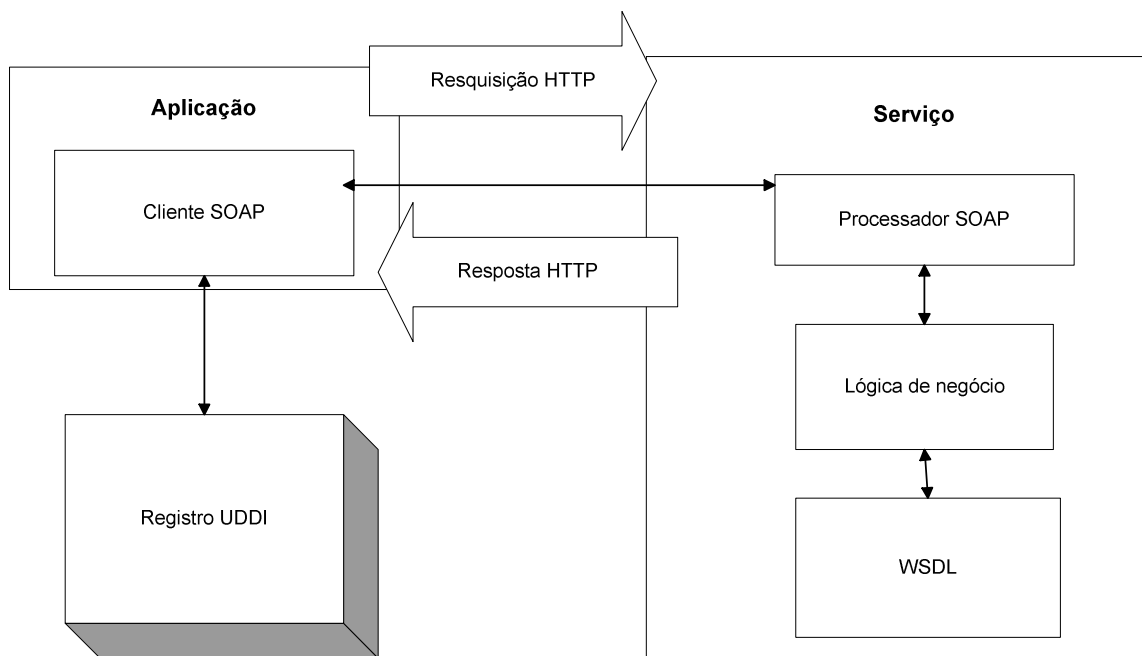


Figura 3 - Arquitetura Web Services (Marzullo, 2009)

O Webservice utiliza em sua arquitetura algumas tecnologias essenciais:

Protocolo HTTP (HyperText Transfer Protocol): Protocolo padrão de transporte de informações da Internet. Este protocolo trabalha em conjunto com os protocolos de rede TCP e IP (Internet Protocol) para que seja possível realizar a conexão entre clientes e servidores, utilizando sockets TCP/IP.

O HTTP utiliza o modelo cliente-servidor, baseado no conceito de requisição e resposta, funcionando da seguinte forma. Uma aplicação cliente abre uma conexão com uma aplicação servidora e envia uma requisição, contendo a versão do protocolo, a URI, uma mensagem MIME (Multipurpose Internet Mail Extensions).

O servidor após receber a requisição do cliente, envia resposta contendo versão de protocolo e um código de operação bem sucedida ou um código de erro, encerrando a conexão após este envio.

Extensible Markup Language (XML): Linguagem de marcação utilizada em larga escala como padrão de armazenamento e transporte de dados. Esta linguagem de marcação é capaz de veicular qualquer tipo de informação necessária em um contexto de negócio. Possui especificação bem definida pela W3C, entidade responsável pela definição das regras de criação de documentos XML.

Baseado na SGML, o XML herdou o poder de marcação, mas recebeu maior facilidade de implementação, após trabalho árduo da equipe de engenheiros da W3C. Desta forma é possível difundir um conhecimento específico de forma inteligente, estruturada e eficaz.

Documentos XML são compostos por elementos denominados nós (tags) que representam parte da informação armazenada, estes nós podem conter outros nós, tal estrutura pode ser visualizada como uma hierarquia em formato de árvore e todo documento deve ser iniciado por um elemento raiz.

Simple Object Access Protocol (SOAP): É uma especificação do W3C para troca de informação estruturada em ambientes distribuídos. Utiliza o XML como estrutura de criação das mensagens trocadas e o HTTP como protocolo de transmissão.

WebService Description Language (WSDL): Tecnologia XML que descreve a interface de um Webservice. Determinando a representação dos parâmetros de entrada e saída de uma chamada externa, além do protocolo de conexão com o serviço.

Universal Description, Discovery, and Integration (UDDI): Descreve um registro mundial de serviços e serve como integração, propaganda e descoberta de serviços.

2.2.6. PostgreSQL

PostgreSQL é um SGBDOR (sistema gerenciador de banco de dados), baseado no POSTGRES desenvolvido pelo Departamento de Ciências da Computação da Universidade da Califórnia em Berkeley, que teve seu projeto financiado pela DARPA

(Defense Advanced Research Projects Agency). Possui código aberto e suporta grande parte do padrão SQL:2003, além de oferecer funcionalidades mais recentes (Group, 2005).

O PostgreSQL possui um boa performance e robustez, além de tipos de dados que propiciam na utilização de armazenamento de informações pertinentes a posicionamento geográfico, tais como:

- Point – ponto geométrico no plano;
- Polygon – caminho geométrico fechado no plano;
- Path – caminho geométrico no plano.

Um exemplo de utilização de um destes tipos de armazenamento de informações geográficas segue abaixo, onde o tipo point requer um par de coordenadas como entrada:

```
INSERT INTO locais VALUES('Curitiba',(-194.0,53.0));
```

3. Protótipo

Neste capítulo é apresentado utilizando notação UML, à arquitetura proposta e o funcionamento do aplicativo desenvolvido como caso de estudo de um GIS móvel.

3.1 Lista de Requisitos

Segue abaixo a lista dos requisitos selecionados para serem implementados no protótipo utilizado como experimento neste trabalho.

Requisito	Descrição	Tipo
Efetuar login	Esta funcionalidade deve efetuar validação do usuário, mediante a informação de seu login e senha, para que libere acesso ao aplicativo	Funcional
Pesquisar Locais	Funcionalidade que permite ao usuário executar pesquisa dos locais próximos a sua posição, mediante a informação de sua posição e a categoria dos locais desejados	Funcional
Visualizar locais no mapa	Esta funcionalidade deve permitir o usuário visualizar sua posição, além dos locais resultantes da pesquisa realizada, em um mapa	Funcional
Manipular interface por toque	A interface do aplicativo deve permitir ao usuário manipular seus componentes, utilizando toque do dedo (Touch screen), além dos botões de navegação	Não Funcional
Resolução de tela	Interface compatível com manipulação utilizando toque do dedo (Touch screen) e tamanho de tela de um dispositivo móvel.	Não Funcional
Comunicação com servidor de dados	O aplicativo deverá obter as informações de um servidor remoto	Não Funcional
Obter posição atual do usuário	O aplicativo deve obter a posição geográfica do usuário, utilizando dispositivo GPS ou utilizando a rede de comunicação	Não Funcional

3.2 Use Cases

3.2.1 Login

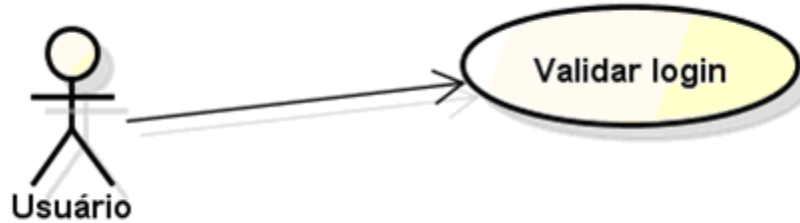


Figura 4 - Validar Login

Descrição	Autenticação do usuário que está acessando o aplicativo
Ator	Usuário (Proprietário da conta)
Fluxo Principal	<ol style="list-style-type: none"> 1. Usuário inicia aplicativo utilizando ícone disponível na área de aplicativos do smartphone; 2. Sistema exibe tela de login contendo os seguintes campos: Login Senha Botão Confirmar Botão Limpar 3. Usuário preenche os campos e pressionar o botão Confirmar; 4. Sistema valida usuário e exibe tela de pesquisa.
Fluxos Alternativos	<ol style="list-style-type: none"> a.1) Usuário preencher formulário e pressiona botão limpar; a.2) Sistema limpa os campos do formulário b.1) Usuário informa login e senha b.2) Sistema invalida usuário e exibe mensagem informando que o usuário informado não existe.
Requisitos Especiais	O smartphone deve possuir acesso a web, para acessar o servidor

Pré-Condições	<ol style="list-style-type: none"> 1. Usuário deve possuir login e senha cadastrada no sistema; 2. O usuário de estar em uma área que possua cobertura de rede de celular ou wifi;
Pós-Condições	O usuário poderá continuar navegando pelas telas do aplicativo

3.2.2 Pesquisa

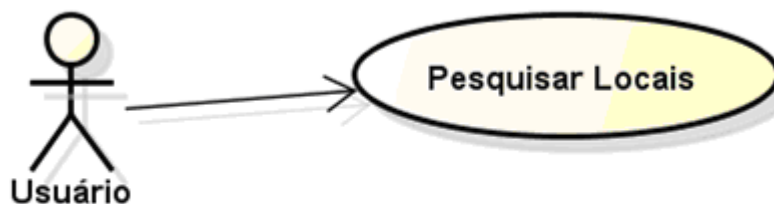


Figura 5 - Pesquisar Locais

Descrição	Pesquisa de locais registrados no servidor que estejam próximos a posição atual do usuário.
Ator	Usuário (Proprietário da conta)
Fluxo Principal	<ol style="list-style-type: none"> 1. Usuário acessa tela de pesquisa, a qual exibirá lista de categorias de locais, além de o botão pesquisar; 2. Usuário seleciona uma categoria e pressiona botão pesquisar; 3. Sistema recupera a posição atual do usuário e efetua pesquisa, no servidor remoto, de locais que pertencem a categoria selecionada; 4. Sistema apresenta lista contendo os locais obtidos na pesquisa, exibindo o nome e a descrição destes; 5. Usuário seleciona item na lista; 6. Local será exibido no mapa (Use complementar)
Fluxos Alternativos	a.1) Usuário pressiona botão retornar (Teclado)

	a.2) Aplicativo retorna a tela anterior; b.1) Sistema não encontra nenhum local registrado nas proximidades da posição atual do usuário; b.2) Aplicativo exibe mensagem informando que não foi encontrado nenhum local com as características pesquisadas;
Requisitos Especiais	O usuário deve estar em uma área que possua cobertura de rede de celular ou wifi;
Pré-Condições	1. O usuário deve ser validado na tela de login.
Pós-Condições	1.Usuário poderá visualizar lista de locais registradas nas proximidades de sua posição e que pertençam a categoria selecionada.

3.2.3 Mapa

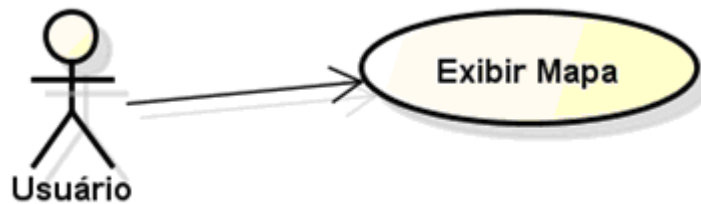


Figura 6 - Exibir Mapa

Descrição	Funcionalidade que exibe a posição atual do usuário, além do local selecionado na pesquisa.
Ator	Usuário (Proprietário da conta)
Fluxo Principal	<ol style="list-style-type: none"> 1. Usuário executa use case de pesquisa 2. Sistema exibe mapa contendo a posição atual do usuário, além da posição do local pesquisado. O sistema apresentará a visualização de rua e zoom.
Fluxos Alternativos	

	a.1) Usuário pressiona botão de zoom in; a.2) Sistema aumenta zoom do mapa. b.1) Usuário desloca o mapa c.1) Usuário seleciona visão de satélite
Requisitos Especiais	O usuário de estar em uma área que possua cobertura de rede de celular ou wifi;
Pré-Condições	1. Usuário deve ter realizado pesquisa de locais e selecionado um local na lista de resultados.
Pós-Condições	1. Usuário pode se localizar para ir até o local desejado.

3.3 Arquitetura

Esta seção apresenta a arquitetura proposta para utilização no protótipo de uma aplicação GIS móvel para atender os requisitos identificados na seção anterior. Na figura **Figura 7 - Arquitetura do Protótipo** é ilustrada como a arquitetura está dividida e no decorrer do capítulo será descrita e detalhada cada parte da arquitetura.

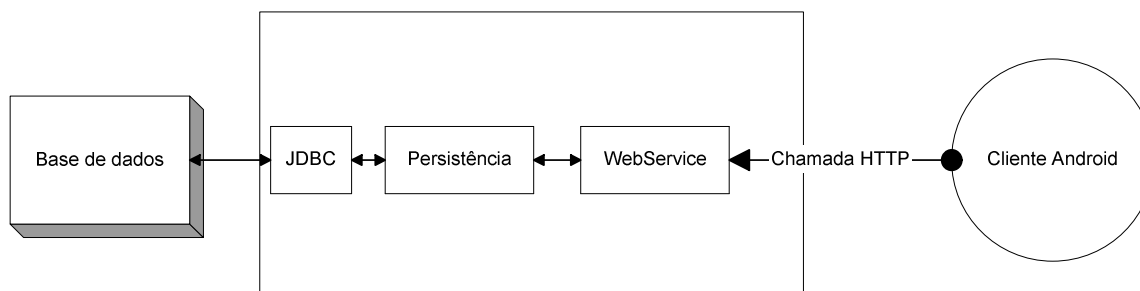


Figura 7 - Arquitetura do Protótipo

Devido ao fato do aplicativo experimental que contempla a lista de requisitos previamente apresentada neste trabalho ter a necessidade de armazenar informações referentes a locais, categorizá-los, armazenar informações do usuário, renderizar mapas e tornar essas informações disponíveis a usuários utilizando um dispositivo móvel, foi identificada a necessidade de dividir a aplicação em partes devidamente responsáveis por determinadas tarefas.

Esta divisão de tarefas se deve ao fato da limitação que dispositivos móveis possuem em relação a armazenamento de grandes volumes de informações e publicação deste

conteúdo. Desta forma para tarefa de **Base de Dados** foi utilizado um SGBD PostgreSQL, possibilitando assim o armazenamento de um grande volume de dado em relação a persistência deste dados em um dispositivo móvel.

Os dados persistidos no SGBD devem ser acessados e processados de alguma forma, para isto foi elaborado um pequeno componente aqui chamado de **Persistência**, pois tem acesso a base de dados, através de funcionalidade de consulta e persistência devidamente padronizadas, conforme será descrito no decorrer desta seção.

O componente **Persistência** pode ser acessado por qualquer tipo de aplicativo Java, mas como o objetivo deste experimento é disponibilizar suas funcionalidades a um aplicativo móvel, houve a necessidade de publicar estas funcionalidades utilizando um **WebService**, pois desta forma o serviço pode ser acessado por um dispositivo móvel de qualquer local que possua cobertura 3G ou WiFi.

O **Cliente Android** por sua vez efetua acesso ao serviço web recuperando apenas as informações necessárias para cada operação que esteja efetuando, ficando com a responsabilidade de manipular a informação, renderizar interface e mapas para o usuário, além de identificar a posição geográfica atual do dispositivo.



Figura 8 - Arquitetura Cliente Servidor (Machado, 2010)

A arquitetura utilizada neste experimento pode ser classificada como cliente servidor devido à característica de possuir um cliente que é responsável pela obtenção dos dados e um processo responsável pela manutenção da informação, que fica a cargo do servidor. (Basttisti, 2005)

3.4 Diagramas de classes

3.4.1 Persistência

A camada ou componente de persistência, responsável por acessar as informações disponíveis na base de dados, foi estruturado utilizando o padrão de projetos DAO (Data Access Object), pois este padrão permite construir uma camada de acesso a base de dados de forma isolada da camada de regra de negócio, tornando dessa forma baixo o acoplamento. O baixo acoplamento permite a mudança do SGBD por outra forma de armazenamento de dados sem que isso implique em mudanças na cada de regra de negócio. (Tacla, 2010)

Para acessar a base de dados foi utilizado o driver JDBC (Java Data Base Connection), pois o driver JDBC permite aos aplicativos Java conectar-se a um banco de dados e manipular a base de dados através da API que este driver oferece (Deitel, 2005). A responsabilidade de controle de acesso ao driver JDBC fica a cargo da classe GerenciadorConexão.

Como o foco principal do experimento é a utilização de informações de posicionamento geográfico, foram implementadas as classes GeoCoordinate e GeoUtils, as quais são destinadas ao armazenamento de coordenada e manipulação dessa informações respectivamente

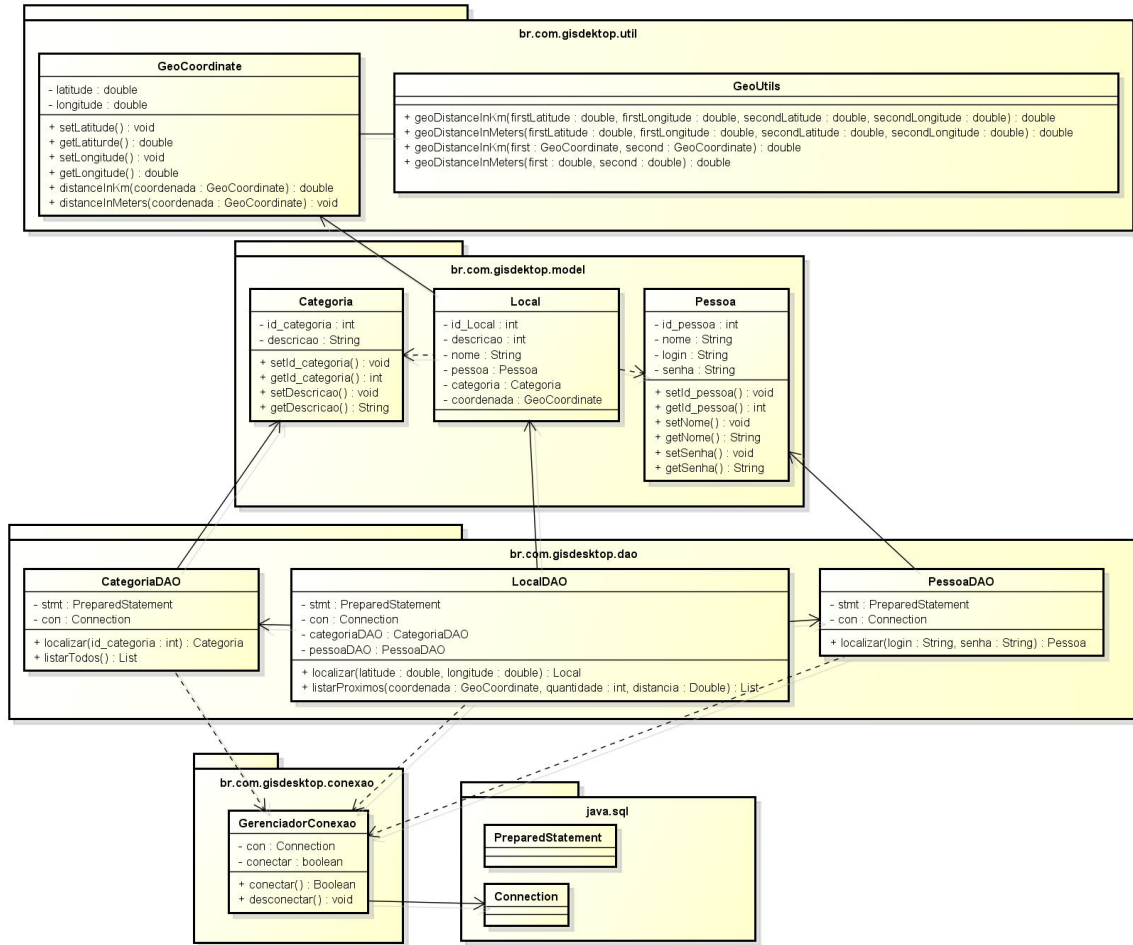


Figura 9 - Diagrama de classe camada persistência

3.4.2 WebService

Para publicação dos métodos de acesso a base de dados, foi implementado o Webservice GISServices, o qual oferece 5 métodos para o cliente utilizar, enquanto há 3 métodos privados, o quais são responsáveis pela montagem das mensagens XML que contém os resultados. Este serviço é disponibilizado em um servidor de aplicação GlassFish.

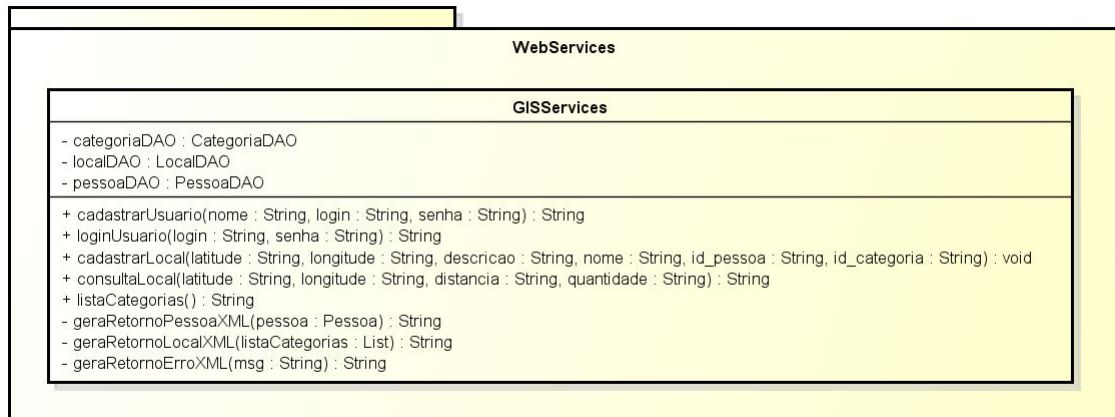


Figura 10 - Diagrama de classe camada web services

3.4.3 Cliente Android

O aplicativo Cliente Android fica embarcado em um dispositivo móvel, sendo estruturado em pacotes com as seguintes atribuições.

br.com.android.gis.activity: Abriga as classes responsáveis pelo controle de interface com o usuário

br.com.android.gis.conexao: Possui a classe GerenciadorConexao, a qual é responsável por estabelecer conexão e comunicação com o Webservice.

br.com.android.gis.util: Neste pacote há classe responsável pelo trabalho de parser das mensagens XML recebidas como resposta do Webservice.

br.com.android.gis.adapter: Abriga classe LocaisAdapter, utilizada pela classe ListaActivity, tendo a responsabilidade de controlar o conteúdo de uma lista, retirando esta responsabilidade da classe de controle de interface.

br.com.android.gis.model: Contém os objetos que recebem os valores das entidades da base de dados, são utilizados como Value Objects.

br.com.android.gis.overlay: Pacote que possui as classes responsáveis por renderizar ícones no mapa.

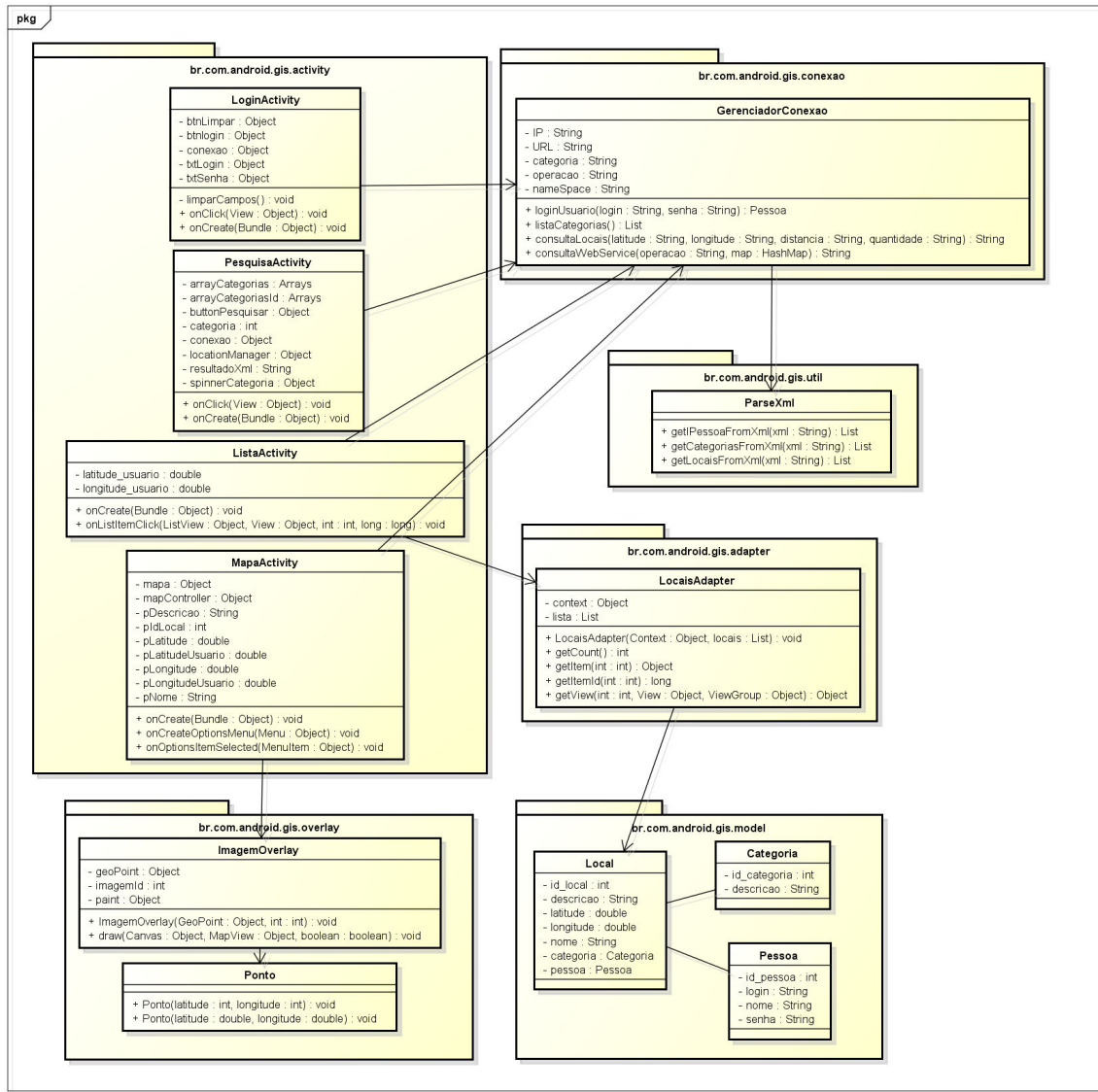


Figura 11 - Diagrama de classe cliente android

3.5 Diagrama de seqüência

Nesta seção são exibidos os principais fluxos do processo de comunicação realizada entre as classes das camadas envolvidas em cada processo

3.5.1 Login Cliente Android

Na figura 12 é ilustrado o fluxo da operação no aplicativo android em que o usuário insere os dados de login e senha e aciona o processo de validação de acesso do usuário, tendo a continuação do fluxo destacado no diagrama da seção seguinte onde é exibido o fluxo realizado a partir do recebimento da solicitação pelo webservice.

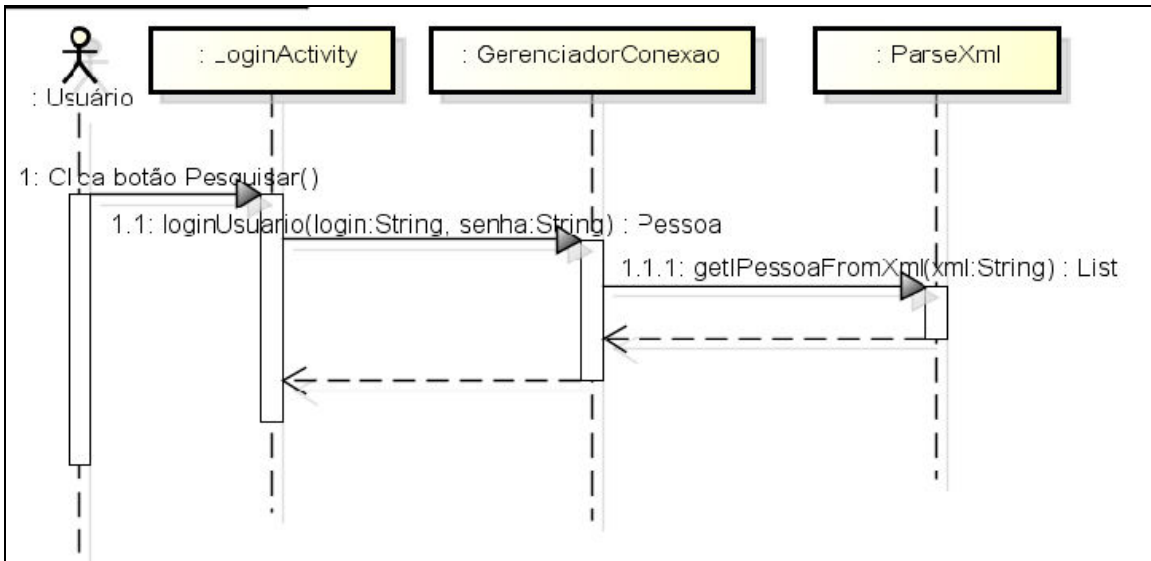


Figura 12 - Diagrama de seqüência login - Cliente Android

3.5.2. Login WebServices

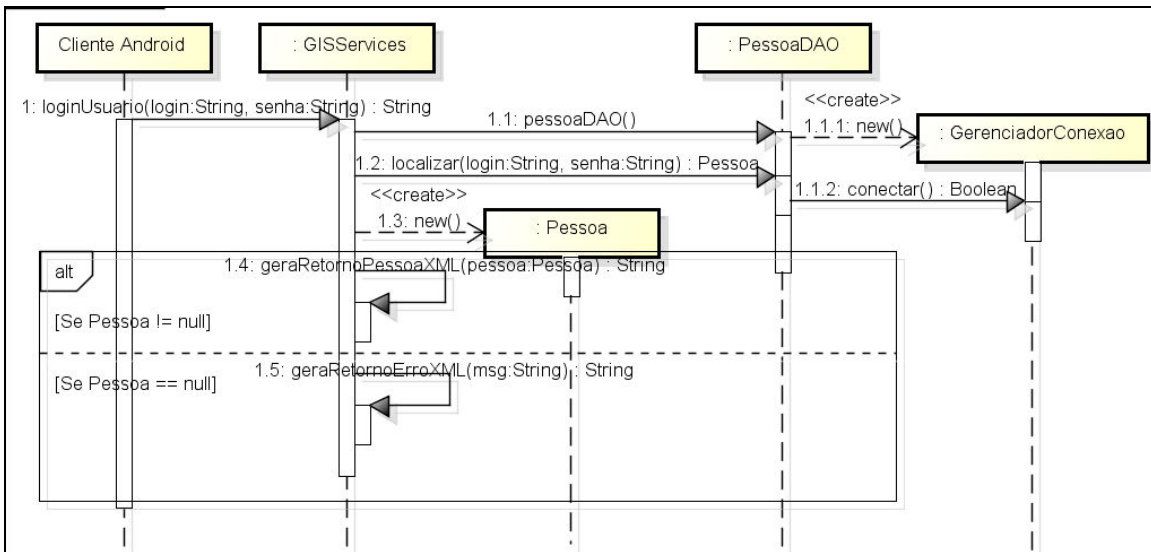


Figura 13 - Diagrama de seqüência login – WebServices

3.5.3 Pesquisar Local Cliente Android

Este diagrama exibe o fluxo realizado durante o processamento da requisição da solicitação de pesquisa de locais que estejam próximos da posição atual do usuário.

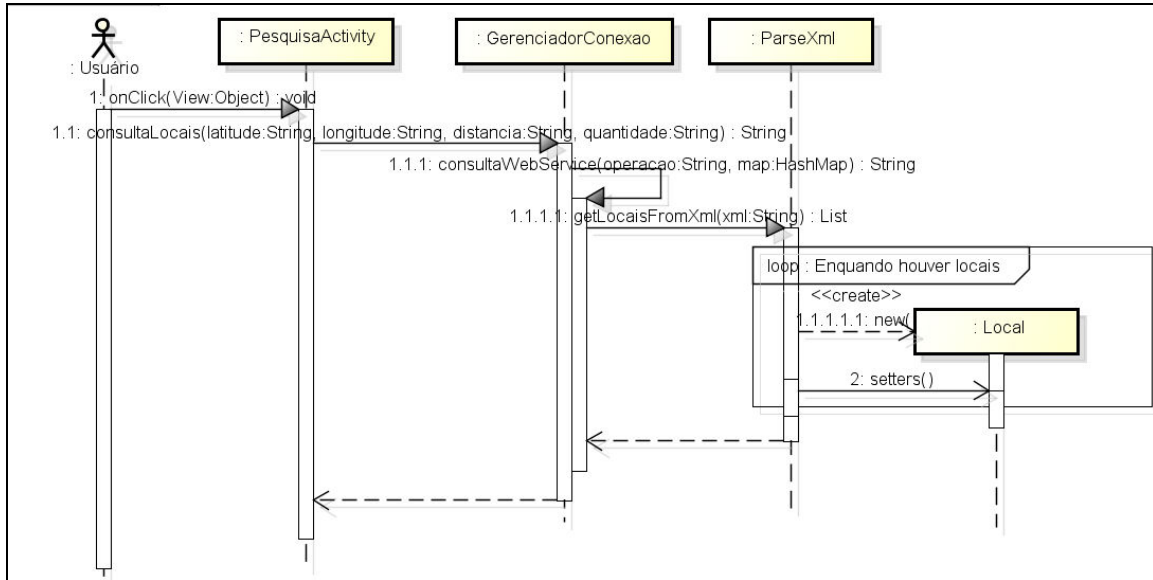


Figura 14 - Diagrama de sequência Pesquisar local - Cliente Android

3.5.4 Pesquisar Local

O diagrama desta seção ilustra o fluxo onde o webservice recebe e processa a solicitação de pesquisa de locais dando continuidade ao processo da seção anterior.

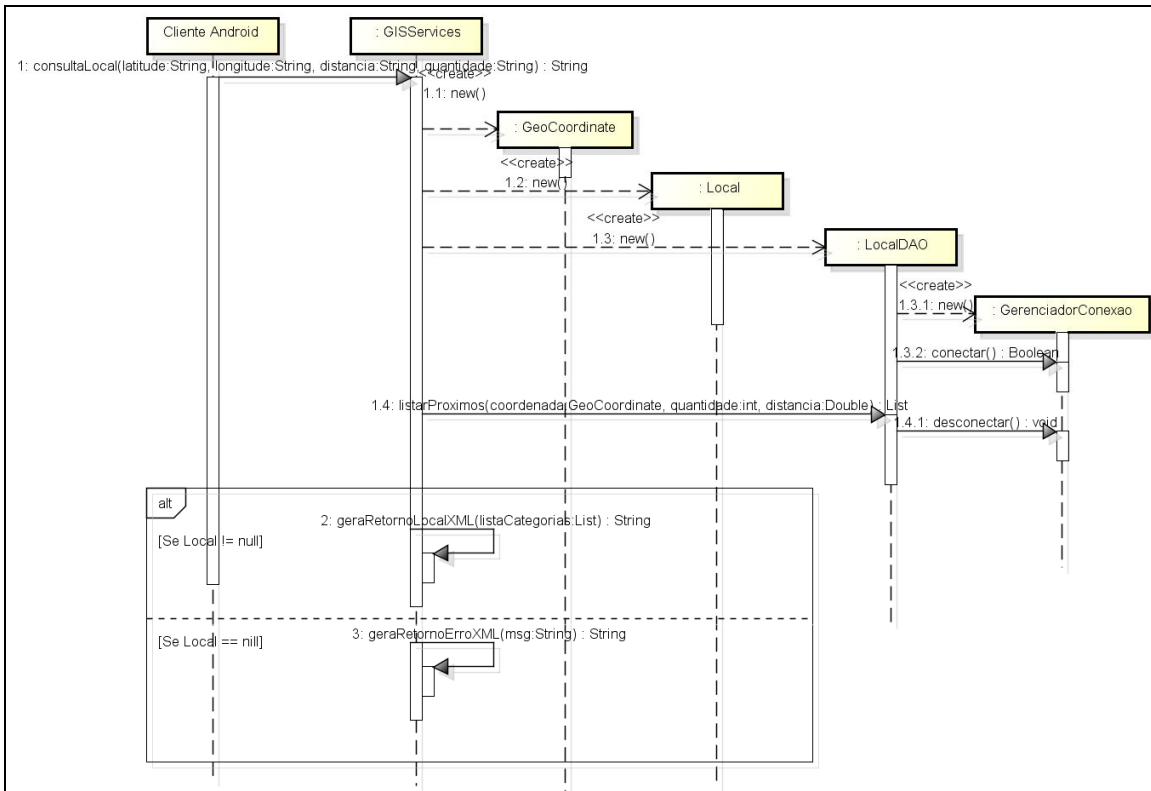


Figura 15 - Diagrama de seqüência Pesquisar local - Web Services

3.6 Diagrama de Entidade e Relacionamento

A estrutura de base de dados criada para o protótipo é baseada no modelo ER (Entidade Relacionamento) conforme pode ser visualizado na **Figura 16 - Diagrama de entidade e relacionamento**

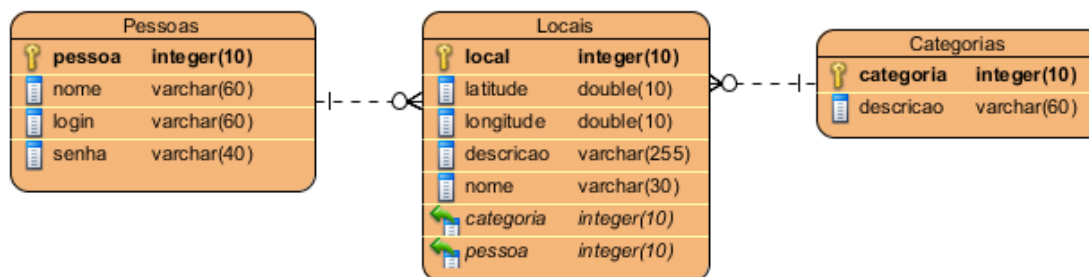


Figura 16 - Diagrama de entidade e relacionamento

A ilustração do modelo utilizado mostra como a estrutura é simples sendo composta por apenas 3 entidades, porém é o suficiente para simular uma base de dados utilizada por uma aplicação com as características de um GIS móvel.

A entidade **Categorias** é utilizada, assim como seu nome sugere, para categorizar os locais registrados na base de dados, facilitando dessa forma a pesquisa por categorias específicas.

Entidade **Locais** representa a estrutura de armazenamento das informações referentes aos locais persistidos na base de dados, os quais possuem vínculo com **Categorias** e **Pessoas**, permitindo dessa forma efetuar pesquisa por categorias e por pessoas que inseriram as informações do local. Apesar do SGBD utilizado no experimento (PostgreSQL) permitir o armazenamento de informação de posicionamento geográfico de forma específica, foi utilizado o tipo Double para armazenar informações sobre latitude e longitude, devido a existência de uma classe específica para efetuar o tratamento destas informações nas camadas de aplicação.

Por sua vez a entidade **Pessoas**, descreve os dados armazenados referentes ao usuário do aplicativo, permitindo ao aplicativo efetuar consulta sobre a permissão de acesso do usuário, além de registrar vínculo entre o usuário que inseriu as novas informações sobre determinado local.

4. Experimento

Neste capítulo é descrito o funcionamento do protótipo construído e apresentado o passo a passo de sua utilização. Além disso, serão discutidos os recursos utilizados para o desenvolvimento, os pontos fortes e as dificuldades encontradas durante o desenvolvimento.

Login

Para a utilização do aplicativo android foi desenvolvido e publicado um serviço web, o qual armazena os dados disponíveis para consulta, além de validar o usuário que está efetuando login.

O serviço web utilizado no experimento teve sua base de dados armazenada em um SGBD PostgreSQL, a qual era acessada por um aplicativo contendo as regras de pesquisa, que por sua vez foi publicado em um servidor GlassFish.

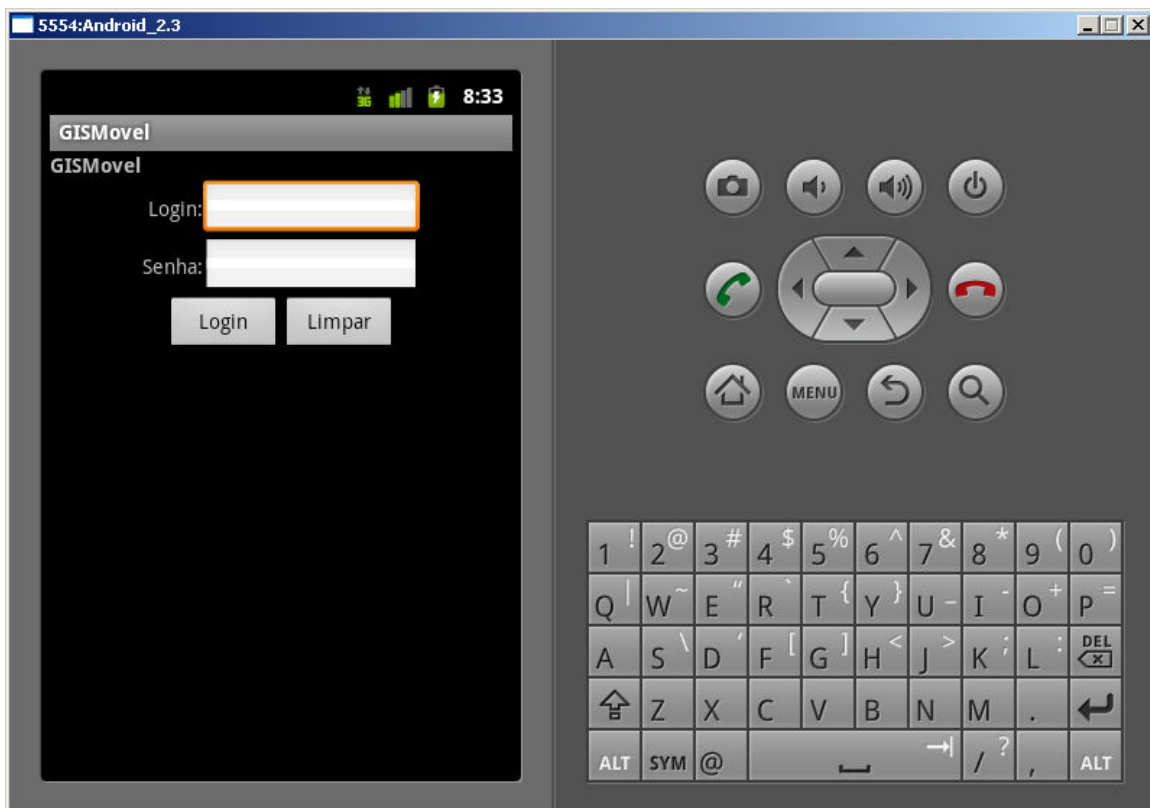


Figura 17 - Tela de login

A requisição de login é efetuada utilizando a chamada de serviço Webservice, passando como parâmetro o login e senha do usuário. O Webservice valida o login do usuário, enviando no seu corpo de resposta uma string seguindo uma estrutura XML, a qual contém a resposta de erro ou as informações pertinentes ao usuário. O XML recebido pelo cliente android é processado e utilizadas às informações de acordo com a necessidade.

Pesquisa

Após validar o login do usuário, o cliente android recebe os dados pertinentes ao usuário logado, além de uma lista de categorias de locais que estão cadastrados na base de dados

O aplicativo móvel utiliza a lista de categorias para preencher o campo Categoria e apresentar à tela de pesquisa, possibilitando ao usuário selecionar a categoria desejada para consulta.

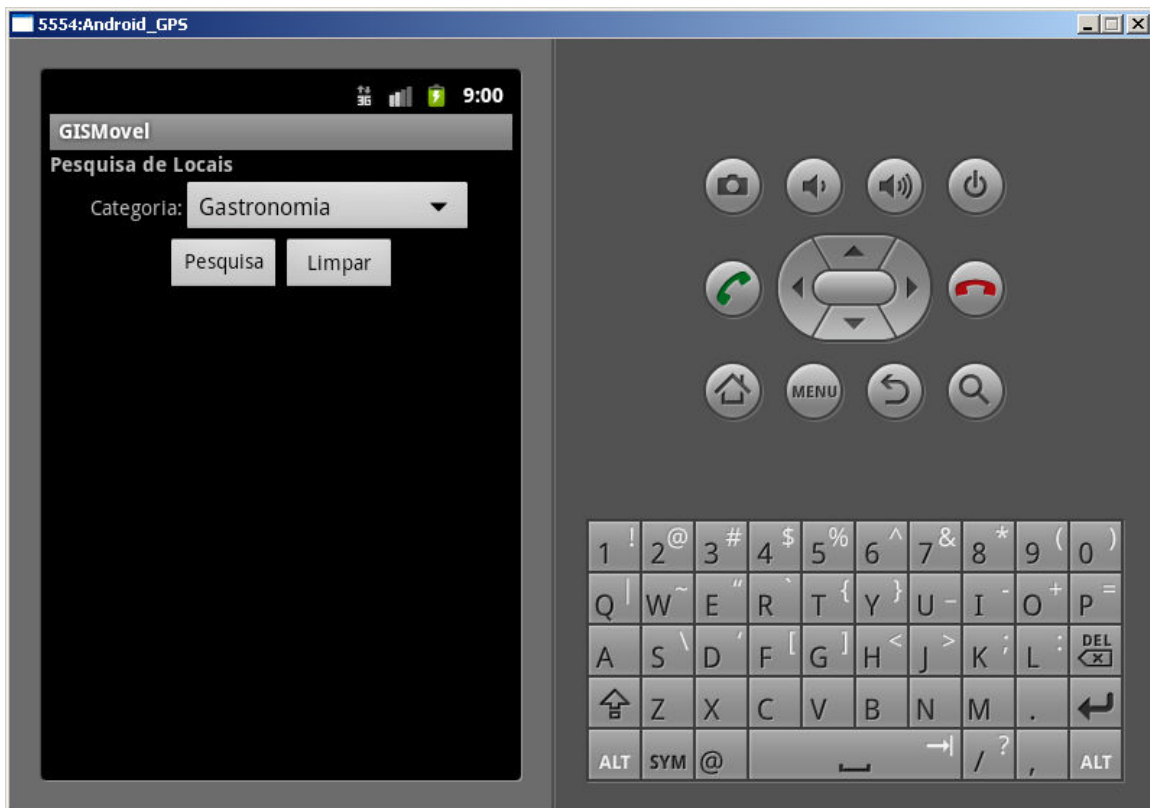


Figura 18 - Tela de pesquisa

Visualização de resultado de pesquisa

Após o usuário selecionar uma categoria de confirmar pesquisa, o aplicativo móvel consulta o webservice solicitando todos os locais disponíveis na base de dados, que pertençam à categoria informada e que estejam próximos em um raio de 3 km do local em que o usuário se encontra.

O resultado obtido é exibido em uma lista, a qual apresenta o nome e uma descrição rápida do local.



Figura 19 - Lista de resultado

Mapa

O local que o usuário seleciona é automaticamente exibido em um mapa, o qual apresenta a posição atual do usuário, além da posição do local selecionado.

O mapa tem como ponto central a posição geográfica do usuário e apresenta visualização de mapa como padrão, além de ter exibir zoom de 16 x, o que possibilita a visualização de aproximadamente 8 quadras ao redor de sua posição.

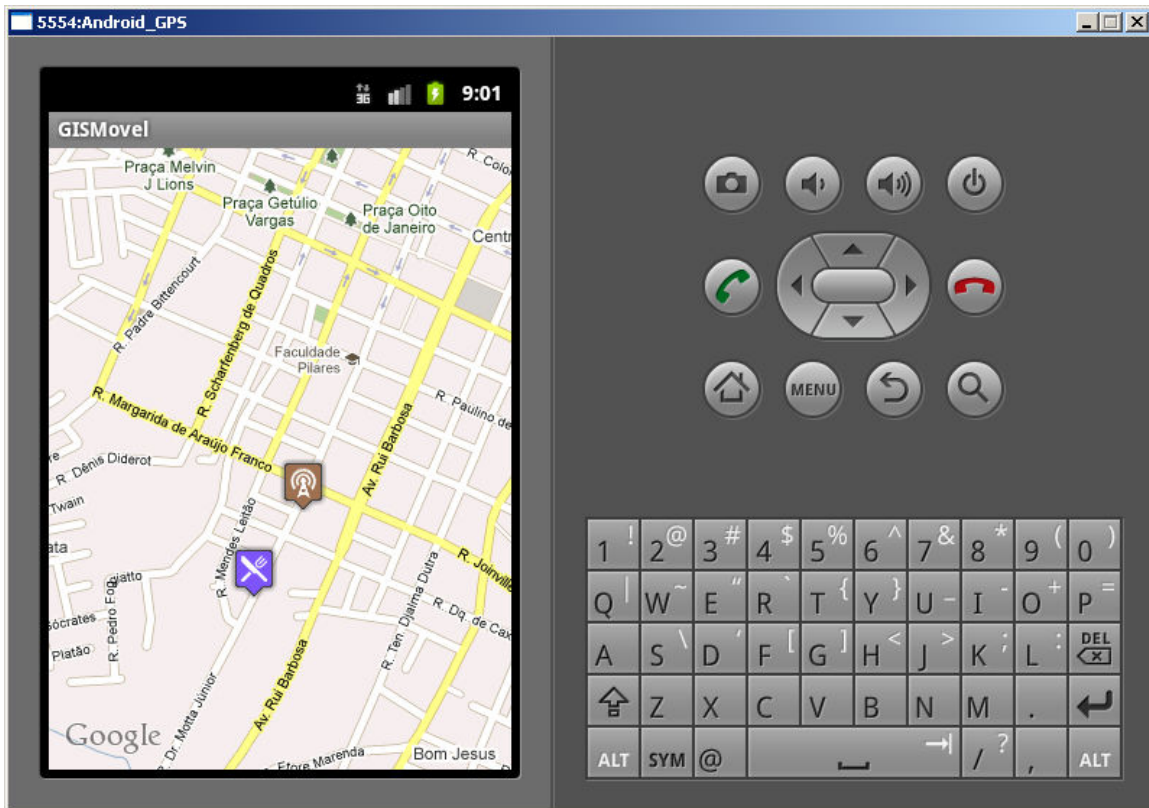


Figura 20 - Mapa

O mapa dispõe de algumas configurações de visualização que o usuário pode manipular caso deseje explorar o local exibido.

Ao pressionar o botão menu do dispositivo móvel, o aplicativo exibe um menu na parte inferior do mapa, contendo quatro botões com as seguintes funcionalidades: aumentar zoom, diminuir zoom, visualizar o mapa com fotos de satélite e visualizar o mapa com representação gráfica de mapa (padrão Google).

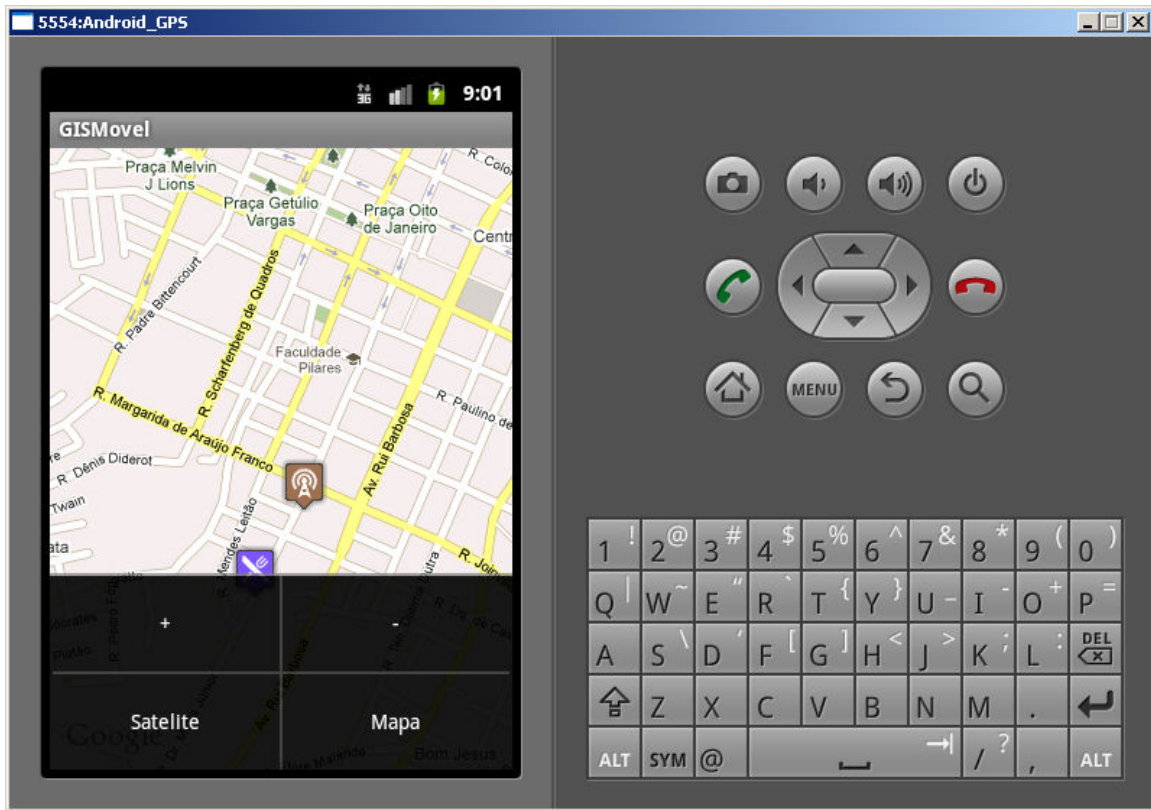


Figura 21- Menu do mapa

A visualização por imagens de satélite é uma boa opção para localizar pontos de referência e identificar possíveis obstáculos no caminho.

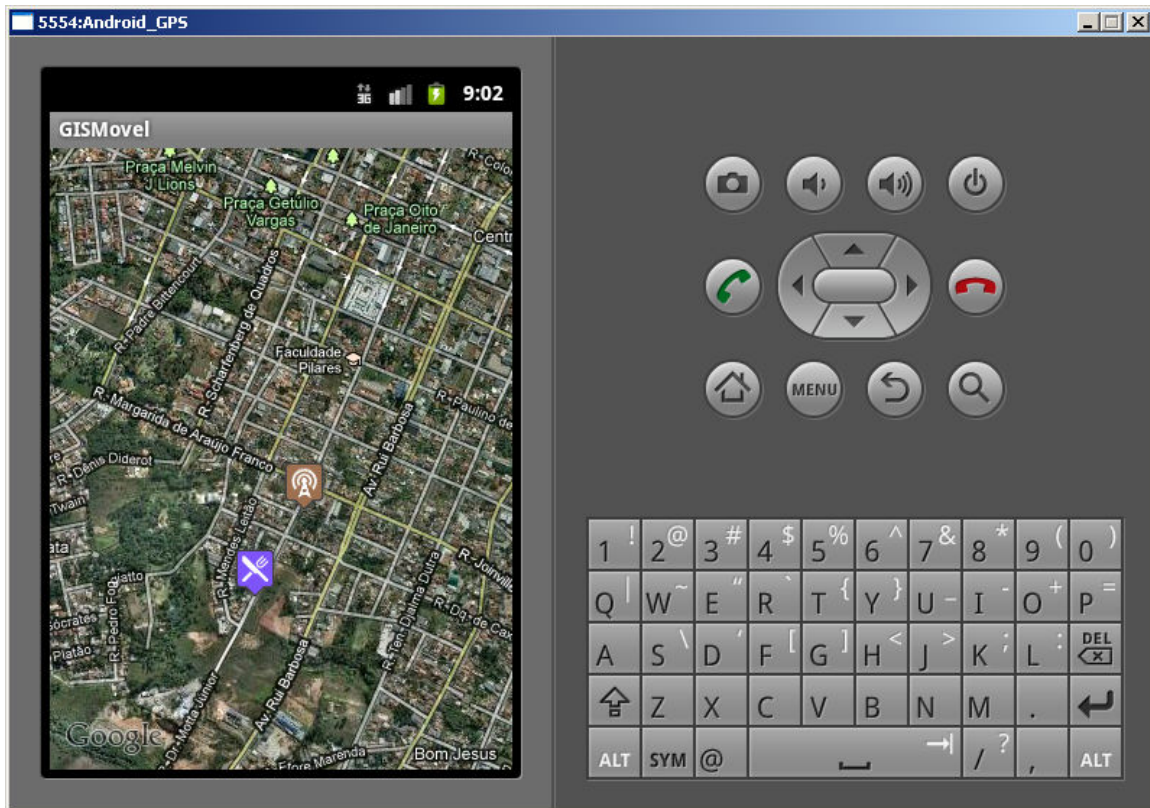


Figura 22 -Visualização de satélite

4.1 Resultados obtidos

Durante a construção do aplicativo utilizado como experimento deste estudo foram encontradas dificuldades em relação à utilização do emulador Android, quanto à funcionalidade de simulação de recebimento de posicionamento geográfico do GPS do dispositivo móvel, mas em contrapartida foram identificadas várias facilidades que a plataforma android disponibiliza ao desenvolvedor.

Como foi descrito no capítulo anterior, o aplicativo de consulta a base de dados utilizado neste experimento necessitava disponibilizar os dados e funcionalidades de pesquisa de forma fácil ao aplicativo móvel, desta forma foi utilizada arquitetura que emprega a tecnologia WebService, a qual possibilita que qualquer tipo de dispositivo ou sistema possa acessar e utilizar as funcionalidades publicadas.

Para construção do WebService foi utilizado o servidor Glassfish e a IDE Netbeans, a qual disponibiliza ao usuário interface que permite o rápido desenvolvimento de um WebService a partir de um aplicativo que possua métodos que serão publicados na internet. Outro ponto positivo do Netbeans em conjunto com o Glassfish seria a funcionalidade de teste do WebService, funcionalidade que agiliza e auxilia o desenvolvedor durante o desenvolvimento.

Para envio de resposta do webservices ao aplicativo requisitante, é utilizado por padrão um “envelope” SOAP, que por sua vez encapsula uma mensagem XML. Neste experimento foi analisada a hipótese de utilização de uma biblioteca para escrita de arquivos XML e parser de conteúdo, tais como JAXB, JAXP, DOM4J, etc. Porém devido às estruturas utilizadas neste experimento representarem pequenos objetos, as mensagens XML foram criadas dinamicamente com a inserção das informações em um template string.

Partindo do princípio que não foi utilizada nenhuma biblioteca para manipulação de mensagens XML no aplicativo back-end, na construção do aplicativo Android também não foi empregado este tipo de recurso, pois o arquivo de instalação do aplicativo móvel deve ser o mais enxuto possível. Como alternativa a este tipo de recurso, foram implementados métodos para realizar o trabalho de parser do arquivo XML, nestes métodos foi empregado recurso de expressão regular disponibilizado pela plataforma Java no pacote java.util.

O ponto que gerou dificuldade neste experimento foi a utilização do emulador android durante o desenvolvimento do aplicativo móvel. Devido o emulador ter a responsabilidade de emular um sistema operacional completo para um aplicativo móvel, a sua inicialização demanda um tempo considerável, o que pode desencorajar o desenvolvedor que está conhecendo a plataforma android. Outro ponto que de certa forma exige paciência e tempo do desenvolvedor é o fato de que o emulador não funciona direito dependendo do sistema operacional instalado na máquina do desenvolvedor e o idioma utilizado, o emulador em certos casos simplesmente não exibe mapa ou ignora a emulação de envio de posição geográfica do GPS.

Em contrapartida ao desempenho questionável do emulador, a plataforma android oferece ao desenvolvedor sua API, possibilitando fácil configuração para localização do posicionamento geográfico do dispositivo móvel, podendo ser alternado entre localização via GPS ou por NetWork, além deste recurso é simples a implementação de funcionalidade de atualização do mapa e inserção de ícones e figuras ilustrativas sobre a posição desejada, recurso o qual deixa mais rico e bonito a funcionalidade de visualização de mapas.

API da plataforma possibilita alterar a forma de visualização do mapa, alternando entre visualização de imagens de satélite ou mapa gráfico, o que é fantástico, pois a menos de 10 anos atrás não tínhamos acesso a mapas dinâmicos e muito menos a imagens de satélite atualizadas. A possibilidade de controle de zoom sobre o mapa é um ponto forte, pois facilita a visualização melhor dos detalhes do mapa, principalmente quando combinado com a visualização de satélite.

Conclusões

Como foi observado nos capítulos anteriores a plataforma Android possibilita ao desenvolvedor construir aplicações que possuam características de um GIS para dispositivos móveis, além de combinar estas características com as facilidades que os dispositivos oferecem atualmente, tais como GPS, acesso a redes 3G e WIFI, etc..

Apesar de em alguns aspectos ser custoso o desenvolvimento utilizando as versões da plataforma Android disponíveis até o momento da realização deste trabalho, é perfeitamente possível produzir aplicações com qualidade necessária para disponibilizar no mercado.

O grande sucesso dos dispositivos móveis utilizando a plataforma Android é a resposta do mercado para o Google, pois o mercado aguardava por uma solução de qualidade que possibilitasse a participação de qualquer desenvolvedor interessado em trabalhar com aplicação móvel, ou seja, uma solução democrática. Talvez isso seja uma característica herdada do Linux, mas de qualquer forma conta com grande aceitação do mercado, o que possibilita o investimento em projetos para criação de soluções que utilizam características de geoprocessamento para dispositivos móveis.

Trabalhos Futuros

Conforme o aplicativo apresentado como experimento neste trabalho nota-se a abrangência de possibilidades de utilização das funcionalidades apresentadas. Devido ao tempo disponível para execução do trabalho apresentado ser inversamente proporcional a quantidade de possibilidades de utilização e detalhamento da tecnologia utilizada, seria possível dar continuidade em um próximo trabalho de estudo, focando no complemento do protótipo apresentado, inserindo funcionalidades tais como:

- Usuário gravar novos locais e suas respectivas informações;
- Vincular fotos aos dados dos locais registrados;
- Criar perfil de usuário contendo locais registrados pelo mesmo, além de locais favoritos;
- Compartilhamento e sugestões de locais para amigos, ou até mesmo compartilhamento utilizando redes sociais;
- Compartilhamento do seu local atual com outro usuário, por exemplo, um pai rastreando o filho;
- Estudo/comparativo entre GIS móvel e LBS (Serviços baseados em localização)
- Avaliar utilização de arquitetura SOA (Arquitetura Orientada a Serviços) em um aplicativo GIS móvel.

Referências Bibliográficas

- Alliance, O. H. Android - Open Source Project. <http://source.android.com/index.html>
- Basttisti, J. Administração e Desenvolvimento – Curso Completo. Rio de Janeiro: Axcell Books. 2005
- Deitel, H. M. Java: como programar. São Paulo: Pearson Prentice Hall. 2005
- Ferreira, A. B. D. H. Minidicionário da Língua Portuguesa. São Paulo: Editora Nova Fronteira. 1985
- Gilberto Câmara, A. M. M. E. C. D. Geoprocessamento: Teoria e Aplicações. Embrapa 2006.
- Group, T. P. G. D. Documentação do PostgreSQL 8.0.0 2005.
- Huerta, E. GPS: posicionamento satelital. Rosario: UNR Editora - Universidad Nacional de Rosario. 2005
- Lecheta, R. R. Google Android – Aprenda a criar aplicações para dispositivos móveis com Android SDK. São Paulo: Novatec Editora. 2010
- Machado, R. D. S. F. Modelagem e prototipação de uma aplicação LBS utilizando a plataforma Android. Curso de Ciência da Computação da Universidade Federal do Rio Grande do Sul, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.
- Marzullo, F. P. SOA na prática: inovando seu negócio por meio de soluções orientadas a serviços. São Paulo: Novatec Editora. 2009
- Piroli, E. L. Introdução ao geoprocessamento Ourinhos: Experimental de Ourinhos. 2010
- Rick Rogers, J. L., Zigurd Mednieks, Blake Meike. Desenvolvimento de aplicações Android. São Paulo: Novatec. 2009
- Rocha, C. H. B. Geoprocessamento - Tecnologia Transdisciplinar. Juiz de Fora: Editora do Autor. 2007. 219 p.
- Tacla, Cesar Augusto. Design Patterns Parte 4: Padrões Factory e DAO. Curitiba 2010. <http://www.dainf.ct.utfpr.edu.br/~tacla/DesignPatterns/0040-JavaDP-DAO.pdf>