



**CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**MONITORAMENTO DE CELULAR POR GPS**

**MARCUS VINICIUS MELO ARAÚJO**

**Orientador:**

Prof. MsC. Francisco Javier de Obaldía Díaz

Brasília

Junho,2011

**MARCUS VINÍCIUS MELO ARAÚJO**

**MONITORAMENTO DE CELULAR POR GPS**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.

Orientador: Prof. MsC. Francisco  
Javier de Obaldía Díaz

Brasília

Junho, 2011

**MARCUS VINÍCIUS MELO ARAÚJO**  
**MONITORAMENTO DE CELULAR POR GPS**

Trabalho apresentado ao Centro  
Universitário de Brasília (UniCEUB)  
como pré-requisito para a obtenção de  
Certificado de Conclusão de Curso de  
Engenharia de Computação.

Orientador: Prof. MsC. Francisco Javier  
de Obaldía Díaz

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação,  
e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -  
FATECS.

---

Prof. Abiezer Amarilia Fernandez  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. MsC. Francisco Javier de Obaldía Díaz  
Orientador

---

Prof. MsC. José Julimá Bezerra Junior  
Instituto Militar de Engenharia - Rio de Janeiro - RJ

---

Prof. MsC. Maria Marony Souza Farias Nascimento  
Mestre em Engenharia Elétrica

## **AGRADECIMENTOS**

Agradeço a todos que contribuíram de alguma forma para a conclusão deste projeto, pela paciência e compreensão dispensadas. A minha namorada Adriana por me ajudar nos teste, me motivar e ter muita paciência, neste período.

Agradeço também ao meu orientador, o professor Francisco Javier, pela pronta ajuda e atenção dispensadas.

## SUMÁRIO

LISTA DE FIGURAS .....	VII
LISTA DE TABELAS .....	IX
LISTA DE EQUAÇÕES .....	X
LISTA DE SÍMBOLOS E ABREVIATURAS .....	XI
RESUMO .....	XII
ABSTRACT .....	XIII
1. INTRODUÇÃO .....	14
1.1. Objetivos .....	15
1.2. Escopo do Trabalho .....	15
1.3. Resultados .....	16
1.4. Estrutura do Trabalho .....	16
1.5. APRESENTAÇÃO DO PROBLEMA .....	17
2. REFERENCIAL TEÓRICO .....	18
2.1. Sistema de Posicionamento Global (GPS).....	18
2.2. Coordenadas Geográficas .....	18
2.2.1. Calculo de distancia entre dois Pontos.....	21
2.3. Banco de Dados MySQL .....	23
2.4. Plataforma Java ME.....	24
2.5. Rede de Transmissão .....	24
2.6. WEB Service.....	25
2.6.1. SOAP .....	25
2.6.2. WSDL .....	27
2.7. PHP .....	28
2.8. Java Script.....	28
2.9. AJAX .....	29
2.10. API Google Maps .....	30
3.1. Modelo Proposto.....	31
3.2. Plataforma Móvel Utilizada.....	32
3.3. Plataforma WEB .....	33
3.4. Software Desenvolvido.....	33
3.5. Interface do Celular .....	35
3.5.1. Pacote Principal .....	36

3.5.1.1. Classe Principal.....	36
3.5.2. Pacote de Dados.....	36
3.5.2.1. Classe getLocalizacao.....	36
3.5.2.2. Classe Dados.....	36
3.5.3. Pacote de Comunicação.....	37
3.6. Aplicativo PHP - Interface Web.....	37
3.6.1. Estrutura.....	37
3.6.2. Banco de dados.....	37
3.6.3. Gerador de XML.....	38
3.6.4. Web Service.....	39
3.7. Apresentação das coordenadas.....	43
3.7.1. Atualização.....	43
3.7.2. Autenticação.....	44
3.7.3. Mapa.....	44
3.7.4. Grupo.....	44
4. APLICAÇÕES E RESULTADOS.....	45
4.1. Descrição da Aplicação do Modelo.....	45
4.2. Avaliação Global da Solução Proposta.....	52
5. CONCLUSÃO.....	55
5.1. Sugestões para projetos futuros.....	55
APÊNDICE I.....	58
APÊNDICE II.....	70
ANEXOS.....	117

## LISTA DE FIGURAS

Figura 1.1 - Topologia do Projeto .....	16
Figura 3.2 - COORDENADAS GEOGRÁFICAS .....	19
Figura 3.3 - Distancia - Rio de Janeiro - São Paulo .....	21
Figura 3.4 - Elementos WSDL .....	27
Figura 2.5 - Chamada Ajax.....	29
Figura 3.1 - Topologia do Projeto .....	31
Figura 3. 2 - Nokia N958G.....	32
Figura 3. 3 - Visão Geral do Software.....	33
Figura 3. 4 - Fluxo de funcionamento da Interface do Celular.....	34
Figura 3.6 - Modelagem do Banco MySQL.....	38
Figura 3. 7 - Resposta da consulta em formato de array. ....	38
Figura 3.8 - Resposta em formato XML .....	39
Figura 3.9- WSDL do web service de posição .....	40
Figura 3.10 - Elemento service.....	41
Figura 3.11 - Elemento binding.....	41
Figura 4.12 - Classe php.....	41
Figura 3. 13 - Elemento portType .....	42
Figura 3. 14 - Elemento message .....	42
Figura 3. 15 - Elemento types.....	42
Figura 3.16 - Classes de entrada e resposta .....	43
Figura 4. 1 - Erro ao criar um usuário duplicado.....	45
Figura 4.2 - Criação de usuário .....	45
Figura 4. 3 Confirmação do uso de recursos do Celular .....	46
Figura 4. 4 - Configuração do software .....	47
Figura 4. 5 - Tela de autenticação.....	47
Figura 4. 6 - Erro ao se autenticar .....	48
Figura 4. 7 - Tela Principal do Projeto .....	48
Figura 4. 8 - Tela Inicial .....	49
Figura 4. 9 - Visão do Grupo.....	50
Figura 4. 10 - Captura de coordenada parado.....	50
Figura 4. 11 - Trajeto Aeroporto até a 407 Sul.....	51
Figura 4. 12 - Trajeto 407 até o UniCeub.....	51
Figura 4. 13 - Coordenadas N95.....	52

Figura 4.14 - Coordenadas GPS TomTom Easy .....53

**LISTA DE TABELAS**

Tabela 3.1- Especificações Técnicas do Nokia N95 8G .....	33
Tabela 4.1 - Distância entre dois pontos sobre distância ( em relação a Brasília) .....	53

**LISTA DE EQUAÇÕES**

Equação 3.1 - Calculo de distância entre dois pontos .....	21
Equação 3.2 - Conversão para graus .....	22

**LISTA DE SÍMBOLOS E ABREVIATURAS**

API	<i>Application Programming Interface</i>
J2ME	Java 2, Micro Edition
GPS	<i>Global Positioning System</i> , Sistema de Posicionamento Global.
MIDP	<i>Mobile Information Device Profile</i>
HTML	<i>HyperText Markup Language</i>
PHP	Acrônimo do inglês - <i>Hypertext Preprocessor</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
IIS	<i>Internet Information Service</i>
WCDMA	<i>Wideband Code Division Multiple Access</i>
HSDPA	<i>High-Speed Downlink Packet Access</i>
RMS	<i>Record Management System (RMS)</i>
WEB	<i>World Wide Web</i>
CSS	<i>Cascading Style Sheets</i>

## RESUMO

Neste projeto é desenvolvido uma solução que envie os dados de posição de um aparelho celular com GPS para um banco de dados na web, sendo que o histórico ficará acessível para consultas posteriores.

A aplicação desenvolvida para aparelho celular faz a leitura dos dados de posição do GPS e em seguida os envia para a internet através da rede de dados 3G. No servidor, os dados são armazenados num banco de dados MySQL e podem ser consultados através da interface WEB, que é responsável por receber e armazenar os dados. Outros aparelhos registrados poderão ser vistos pelos usuário, sabendo assim a distancia entre eles.

A aplicação WEB foi desenvolvida usando a ferramenta IDE Netbeans, escrita na linguagem de programação PHP (Hypertext Preprocessor), fazendo uso do banco de dados Mysql 5. Estas informações podem ser acessadas através de interface visual por intermédio da API (Application Programming Interface ) do Google Maps®. O aplicativo procura mostrar no mapa as coordenadas recebidas do aparelho celular.

A interface instalada no aparelho celular, que é responsável por obter os dados do GPS e enviá-los para WEB foi escrita em J2ME usando IDE Netbeans. A comunicação entre o celular e o servidor foi implementada utilizando *WEB Services*. A mesma possibilita a comunicação entre diversos sistemas distintos independente de qualquer modelo de programação.

**Palavras Chave: GPS, PHP, MySQL, J2ME, Netbeans, SOAP.**

## ABSTRACT

This project aims to develop a solution to send the position data of a mobile phone with GPS to a database on the WEB, which will keep a log accessible for further consultation.

The application developed for the mobile device reads the data of the GPS position, then sends it to the internet using 3G data network. On the server, the data is stored in a MySQL database and can be accessed through the WEB interface, which is responsible for the reception and storage of the data. Other registered devices can be seen by the user, in order to know the distance between them.

The web application was developed using the IDE Netbeans tool, written in the programming language PHP (Hypertext Preprocessor), using MySQL 5 database. This information can be accessed through visual interface by Google Maps© API (Application Programming Interface). The application shows on the map the coordinates received from the mobile device.

The interface installed on the mobile device, which is responsible for obtaining the data from the GPS and send it to the WEB, was written in J2ME using IDE Netbeans. The communication between the mobile device and the server was implemented using *Web Services*. It enables communication among many distinct systems independently of any programming model.

**Keywords:** GPS, PHP, MySQL, J2ME, Netbeans, SOAP.

## 1. INTRODUÇÃO

O Século XXI deixou de ser o futuro e tornou-se presente. O avanço da tecnologia se dá de forma rápida e precisa, antes o que era possível somente nos livros e filmes de ficção científica, agora é parte do cotidiano da humanidade. O ser humano, social, tecnológico, evoluiu da esfera analógica para a esfera digital. A humanidade, com a popularização da Rede Mundial de Computadores (Internet), iniciou uma nova era de uso e acesso a informação. A possibilidade ubíqua de conexão e comunicação propiciou e forçou um avanço social, na forma em que a sociedade moderna se comunica e relaciona. Há na sociedade do século XXI uma demanda criada com estes novos modelos de relações, exemplificada pelo advento das diversas redes sociais. O homem agora está em contato com o mundo, e este ao alcance dos dedos. O novo homem do século XXI, social e tecnológico, tem necessidade de comunicação e informação cada vez maior.

O grande impulso e modificação na forma de comunicação da humanidade surge com a escrita. O salto seguinte foram com as comunicações analógicas, primeiro com o telégrafo seguido pelo telefone, em fins do século XIX, e a invenção do rádio, já no início do século passado. Na segunda metade do século XX as comunicações se tornam digitais e no seu final a comunicação móvel se populariza. Este último passo após a invenção de cientistas Norte-Americanos do laboratório Bell e construído pela empresa Ericsson, que possibilitou a telefonia móvel, o telefone celular. A primeira utilização desta nova tecnologia deu-se em 3 de abril de 1973 em Nova York. Uma ligação feita por Martin Cooper de um aparelho Motorola Dynatac 8000X. A partir desta data, o uso do telefone celular tem se tornado comum. No Brasil o sistema foi inaugurado no Rio de Janeiro em 1990.

O preço inicial dos serviços de telefonia celular nos Estados Unidos estavam em torno de 20 mil dólares no ano de seu lançamento. Hoje, no ano de 2010, temos diversos serviços de telefonia cujos valores são acessíveis a maior parte da população mundial. Este avanço na tecnologia de comunicação, o barateamento de aparelhos e serviços, foram fatores que possibilitaram uma maior disseminação do telefone celular dentre as diversas camadas da população.

A popularização do acesso a telefonia móvel, com o aumento da cobertura das redes de telefonia móvel e conseqüente aumento de oferta de aparelhos de celular, e o desenvolvimento de novas tecnologias possibilitou o a integração de diversas funções a aparelhos de telefonia celular. O celular passou de um objeto de luxo, para um artigo comum,

onde foram agregadas funções, como acesso a internet, e-mail, câmera fotográfica, GPS, que visam suprir a necessidade de comunicação e interação da sociedade do Século XXI, principalmente após o desenvolvimento das redes de terceira geração.

Visando este novo modelo de interrelação entre o homem em sociedade e a tecnologia, o projeto visa apresentar a localização do aparelho celular fazendo uso de seu GPS interno.

### **1.1. Objetivos**

O projeto tem como objetivo geral apresentar uma solução que utilize o GPS do celular, para a obtenção de dados de posicionamento. Estes dados são enviados por meio de conexão de dados 3G para uma interface web onde poderão ser visualizadas. Para tanto, foram desenvolvidas duas interfaces, uma a ser instalada no celular, utilizando a linguagem J2ME, e outra constituída de uma aplicação web responsável pelo recebimento das informações, gravação em um banco de dados, armazenamento das informações e processamento das informações recebidas, disponibilizando sua visualização através do Google Maps.

Como objetivos específicos estão:

- Visualizar as informações enviadas pelo celular através da interface web desenvolvida;
- Criação e leitura do banco de dados para a exibição das últimas coordenadas recebidas.
- Possibilidade de observação através do Google Maps das coordenadas recebidas.
- O cadastro de outros usuários do mesmo software, possibilitado obter a distancia entre os usuários.

Resultados esperados:

1. Visualização das coordenadas capturadas;
2. Localização atual do aparelho;
3. Medição de distâncias entre celulares cadastrados.

### **1.2. Escopo do Trabalho**

No projeto é implementado um sistema que envia coordenadas do GPS do celular para um servidor na WEB. As coordenadas são armazenadas e exibida em uma pagina web através

do Google Maps. O sistema permite a criação de grupos possibilitando saber a distancia que os aparelhos se encontram. Na figura 1.1 é mostrada a visão geral da solução.

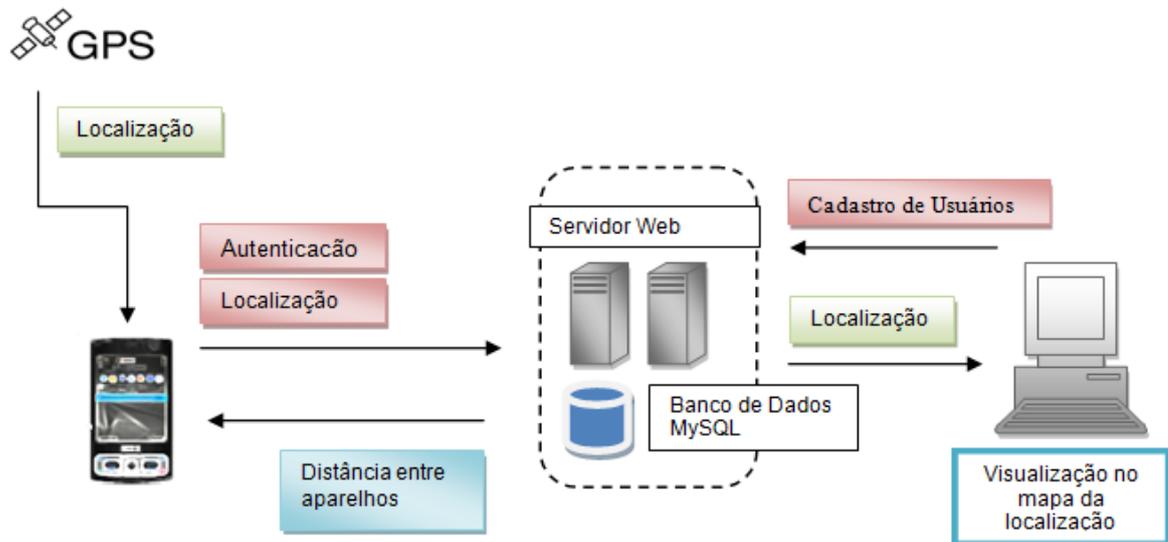


Figura 1.1 - Topologia do Projeto

### 1.3. Resultados

Como resultado espera-se que o projeto seja capaz de enviar as coordenadas para um servidor WEB. Sendo capaz de saber a distância entre aparelhos que estejam cadastrados no mesmo grupo.

### 1.4. Estrutura do Trabalho

Esta monografia está estruturada em cinco capítulos. O primeiro capítulo, Introdução, trata da motivação, objetivos, justificativa e relevância do tema proposto, o escopo do trabalho, os resultados atingidos e a estrutura do trabalho seguida da Apresentação do Problema, trata da apresentação do problema, mostrando soluções existentes em relação ao projeto desenvolvido.

No terceiro capítulo, Bases Metodológicas, são mostradas as tecnologias e ferramentas utilizadas na formação do projeto. Ele está organizado em temas que descrevem as tecnologias e metodologias utilizadas no projeto, sendo eles: J2ME, PHP, MySQL, *Web Services*, SOA, SOAP, Java Script, Rede de transmissão, API Google Maps, GPS, Coordenadas geográficas.

O terceiro capítulo, Proposta e Solução do Modelo, apresenta a arquitetura do projeto e a descrição dos seus elementos. Este capítulo descreve a seqüência de cada etapa da implementação.

No quarto capítulo, Aplicações e Resultados, são mostrados os resultados obtidos, as dificuldades encontradas. São apresentados também os pontos fortes e fracos da solução proposta.

No quinto capítulo é apresentada a conclusão do trabalho e propostas de trabalhos futuros.

## **1.5. APRESENTAÇÃO DO PROBLEMA**

Saber a localização de veículos ou pessoas é algo muito útil. Sistemas de localização são utilizados por diversas empresas de transporte (buscando diminuir o número de roubos de cargas) e logística (para rastrear seus produtos e movimentação de estoques). O campo de geoposicionamento possui um enorme leque de possibilidades, não se restringindo somente ao uso comercial. Voltando-se para o uso pessoal, temos uma constante de preocupação de pessoas em saber onde estão os membros de sua família, em especial suas crianças ou adolescentes, ou pessoas com problemas de memória e idosos.

Muitas soluções de mercado são pensadas visando grandes clientes, utilizando aparelhos e softwares proprietários. Com isso, ficamos vinculados a determinada empresa e pacote de serviços. Citando como exemplo, sistemas de gerenciamento de frotas via satélite. Nele é necessário se instalar um dispositivo na parte de cima do caminhão enviando as informações via rádio, para uma estação que recebe as informações.

A solução desenvolvida nesse projeto serve para uso familiar ou empresarial em pequeno porte, possibilitando monitorar a localização de familiares ou empregados, propondo ser uma alternativa aos sistemas proprietários com vínculos de empresas.

O sistema proposto deverá ser instalado em um aparelho de celular com GPS integrado. As informações de posicionamento serão transmitidas periodicamente utilizando o serviço de dados 3G para um servidor na WEB armazenando-as em um banco de dados.

Por meio de uma interface WEB será possível visualizar no mapa a localização do celular. Um usuário do sistema poderá também ter visibilidade de outros celulares que estiverem registrados, adicionando-o à lista de celulares monitorados, identificando assim a distância uns dos outros.

## **2. REFERENCIAL TEÓRICO**

Neste capítulo são apresentadas as tecnologias empregadas para a construção do projeto. O projeto faz uso do equipamento de GPS, método de comunicação entre as interfaces desenvolvidas e do uso do banco de dados para armazenamento das informações recebidas. As linguagens de programação e as técnicas utilizadas também serão apresentadas.

### **2.1. Sistema de Posicionamento Global (GPS)**

O sistema de GPS do (Acrônimo do inglês – Global Positioning System, Sistema de Posicionamento Global) , criado originalmente para fins militares, sendo desenvolvido e pela Força Aérea Americana. O sistema fornece alta precisão na localização de pontos no globo terrestre.

O sistema é composto por 28 satélites sendo 4 sobressalentes em 6 orbitas diferentes. As órbitas foram calculadas para que ao menos 4 satélites sejam visíveis em qualquer parte do globo terrestre. O GPS usa triangulação para determinar a localização. Os satélites provem dois tipos de sinal, com diferentes precisões. Para o uso civil o sinal é degradado, fornecendo precisão de aproximadamente de 100 metros. No uso militar a precisão é de aproximadamente 20 metros. [Wikipédia – Sistema de Posicionamento Global – 15/09/2010].

O receptor capta os sinais de quatro satélites para determinar as suas próprias coordenadas, e ainda o tempo. Então, o receptor calcula a distância a cada um dos quatro satélites pelo intervalo de tempo entre o instante local e o instante em que os sinais foram enviados (esta distância é chamada pseudodistância). Descodificando as localizações dos satélites a partir dos sinais de microondas (tipo de onda eletromagnética) e de uma base de dados interna, e sabendo a velocidade de propagação do sinal, o receptor, pode situar-se na intersecção de quatro cones, uma para cada satélite. [Wikipédia – Sistema de Posicionamento Global – 15/09/2010].

### **2.2. Coordenadas Geográficas**

O sistema de mapeamento da Terra através de coordenadas geográficas expressa qualquer posição horizontal no planeta através de duas das três coordenadas existentes num sistema esférico de coordenadas, alinhadas com o eixo de rotação da Terra. Herdeiro das

teorias dos antigos babilônios, expandido pelo famoso pensador e geógrafo grego Ptolomeu, um círculo completo é dividido em 360 graus ( $360^\circ$ ). [Wikipédia – Coordenadas geográficas– 22/04/2011].

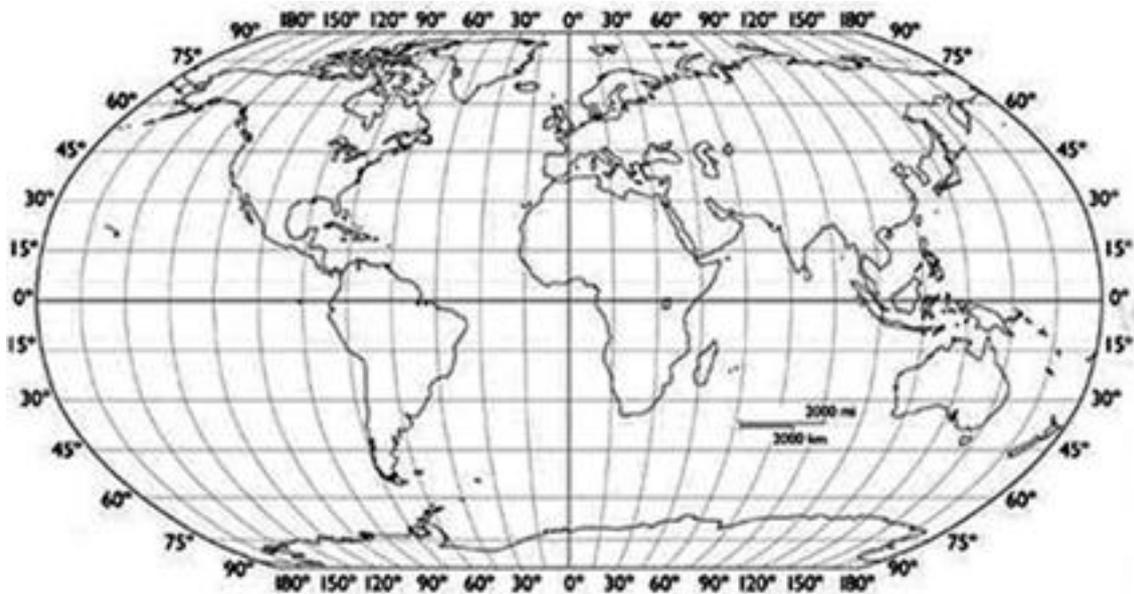


Figura 3.2 - COORDENADAS GEOGRÁFICAS (Fonte: [http://www.geografiaparatos.com.br/img/mapasm/coord\\_geo.jpg](http://www.geografiaparatos.com.br/img/mapasm/coord_geo.jpg))

Coordenadas geográficas são linhas imaginárias verticais e horizontais, que dividem a terra em graus, minutos e segundos, tornando possível definir qualquer ponto no globo terrestre. As latitudes são as linhas paralelas ao Equador, que é o ponto  $0^\circ$  ou ponto de origem. As longitudes são as linhas paralelas ao meridiano de Greenwich que é o ponto  $0^\circ$  ou ponto de origem. (FERREIRA, 2010)

A latitude é dividida em linhas paralelas, que começam no ponto de origem, que é o Equador, como visto ver na figura 3.2, e seguem de  $0^\circ$  a  $90^\circ$ , nos dois sentidos (N=Norte e S=Sul). A longitude é dividida em meridiano que começam no ponto de origem, o meridiano de Greenwich, e segue de  $0^\circ$  a  $180^\circ$  nos dois sentidos (L=Leste e O=Oeste). (FERREIRA, 2010)

Os pontos na superfície terrestre podem ser definidos pelo cruzamento da latitude e longitude. (FERREIRA, 2010)

Existem pelo menos quatro modos de designar uma localização exata para qualquer ponto no globo terrestre. Nos três primeiros sistemas, o globo é dividido em latitudes, que vão de 0 a  $90^\circ$  (Norte ou Sul) e longitudes, que vão de 0 a  $180^\circ$  (Leste ou Oeste). Para efeitos práticos, usam-se as siglas internacionais para os pontos cardeais: N=Norte, S=Sul,

E=Leste/Este, W=Oeste. Para as latitudes, o valor de cada unidade é bem definido, pois o grande círculo tem 20.003,93km, dividindo este último por 180, conclui-se que um grau (°) equivale a 111,133km. Dividindo um grau por 60, toma-se que um minuto (') equivale a 1.852,22m. Dividindo um minuto por 60, tem-se que um segundo (") equivale a 30,87m. Para as longitudes, há um valor específico para cada posição, que aumenta de 0 nos Pólos até a Linha do Equador, onde está o seu valor máximo. (Corrêa, 2011)

Como forma de se demonstrar as diferenças entre cada um dos sistemas, usar-se-á o exemplo para as coordenadas de um lugar específico: a Catedral Metropolitana de Porto Alegre. (Corrêa, 2011)

### **Graus - Minutos - Segundos**

Neste sistema, cada grau é dividido em 60 minutos, que por sua vez se subdividem, cada um, em 60 segundos. A partir daí, os segundos podem ser divididos decimalmente em frações cada vez menores. Deste modo, a localização da Catedral neste sistema é: 30°01'59,512"S e 51°14'07,012"W. (Corrêa, 2011)

### **Graus - Minutos Decimais**

Neste sistema, cada grau é dividido em 60 minutos, que por sua vez são divididos decimalmente. A localização da Catedral fica sendo: 30°01,992'S e 51°14,117'W . (Corrêa, 2011)

### **Graus Decimais**

Neste sistema, cada grau é dividido em frações decimais. A forma de nomeação difere um pouco dos dois primeiros sistemas: a latitude recebe a abreviatura lat e a longitude, lon. Há valores positivos e negativos. Os valores positivos são para o Norte (latitude) e o Leste (longitude) e não recebem um símbolo específico. Os valores negativos são para o Sul (latitude) e o Oeste (longitude), sendo acrescidos do símbolo -. A Catedral tem aqui esta localização: lat -30,0331977° e lon -51,2352811°. (Corrêa, 2011)

### **Universal Transversa de Mercator**

Para efeitos de comparação, o sistema UTM usa três dados em vez de dois. O primeiro é o setor do globo terrestre (Fuso), o segundo é a distância relativa ao centro do meridiano - sempre 500.000,00m - e o terceiro é a distância do Pólo Sul (para lugares situados no Hemisfério Sul) ou da Linha do Equador (para lugares situados no Hemisfério Norte). Nele, a

localização da Catedral é: 22K E=477.826,84m N=6.677.465,44m S (Sistema Carta Geral do Brasil). (Corrêa, 2011) (Corrêa, 2011)

### 2.2.1. Cálculo de distância entre dois Pontos

Para se fazer o cálculo da distância entre duas coordenadas, tomaremos a Terra como uma esfera, e faremos uso da trigonometria esférica. Para isso faremos uso da seguinte fórmula.

$$\begin{aligned} \cos(b) &= \cos(a) \cdot \cos(c) + \sin(a) \cdot \sin(c) \cdot \cos(B) \\ \cos(c) &= \cos(b) \cdot \cos(a) + \sin(b) \cdot \sin(a) \cdot \cos(C) \\ \cos(a) &= \cos(b) \cdot \cos(c) + \sin(b) \cdot \sin(c) \cdot \cos(A) \end{aligned} \quad (\text{Eq. 3.1})$$

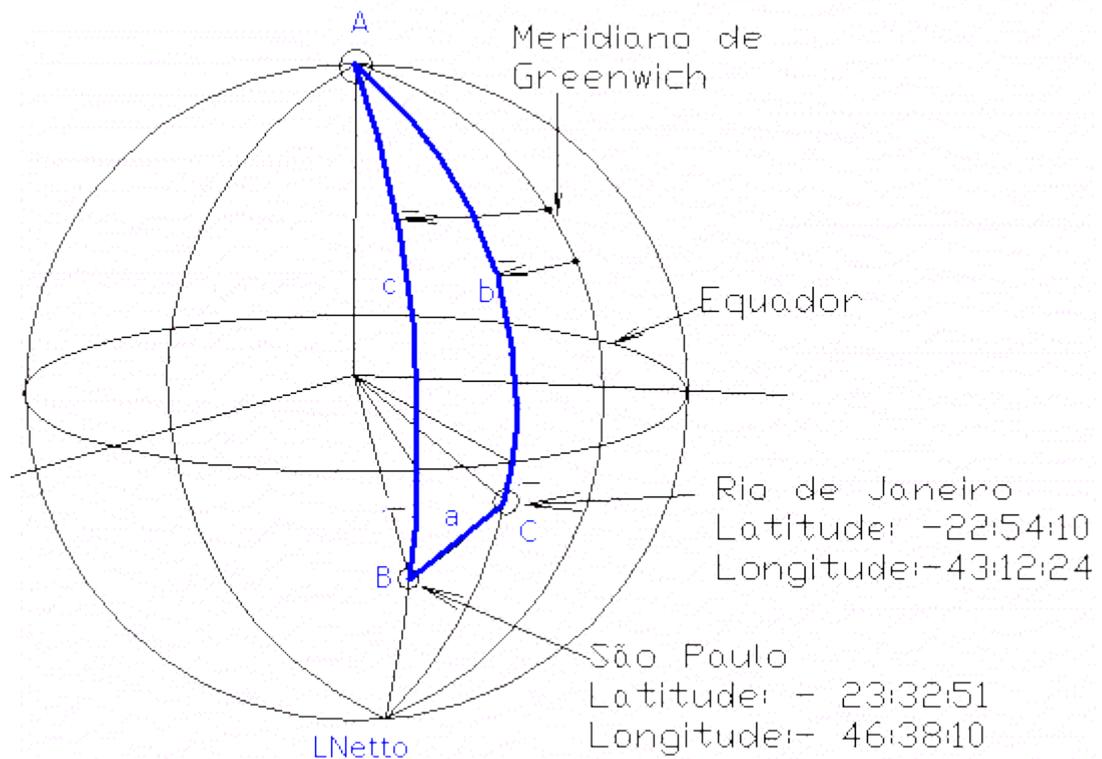


Figura 3.3 - Distância - Rio de Janeiro - São Paulo (Fonte: [http://caraipora.tripod.com/calc\\_dist\\_entre\\_dois\\_pontos.htm](http://caraipora.tripod.com/calc_dist_entre_dois_pontos.htm))

Para exemplificar será utilizada a distância entre Rio de Janeiro e São Paulo. Para o cálculo será necessário primeiramente converter os dados que estão em horas:minutos:segundos para graus.

Transformando-se tudo para segundos

$$\text{Horas} \cdot 60 \cdot 60 = A$$

Minutos . 60 = B

Segundos = C

Converte para graus

$$(A + B + C) / 60 . 60 = \text{Resultado em graus} \quad (\text{Eq 3.2})$$

### **Rio de Janeiro**

Latitude:

$$\begin{aligned} -22:54:10 &= -((22.60.60)+(54.60)+(10)) = -((79.200) + (3.240)+ (10)) = -(82.450) \rightarrow - \\ 82.450/3600 &= -22,902778 \end{aligned}$$

Longitude:

$$- 43:12:24 = -((43.60.60) + (12.60) + (24)) = -155.544 \rightarrow -155.544/3600 = - 43,206667$$

### **São Paulo**

Latitude:

$$\begin{aligned} - 23:32:51 &= -((23.60.60) + (32.60) + (51)) = -((82.800) + (1.920) + (51)) = - (84.774) \rightarrow - \\ 84.774/3600 &= - 23,548333 \end{aligned}$$

Longitude:

$$\begin{aligned} - 46:38:10 &= - ((46.60.60) + (38.60) + (10)) = - (165.600+2.280+10) = - (167.890) \rightarrow - \\ 167.890/3600 &= - 46,636111 \end{aligned}$$

**Arco AB = c**

$$90 - (-23,548333) = 90+23,548333 = 113,54833$$

**Arco AC = b**

$$90 - (-22,902778) = 112,90278$$

**Arco ABC = a**

Diferença das longitudes:

$$-43,206667 - (-46,636111) = -43,206667 + 46,636111 = 3,42944$$

Aplicando a fórmula:

$$\cos(a) = \cos(b) \cdot \cos(c) + \sin(b) \cdot \sin(c) \cdot \cos(A)$$

$$\cos(a) = \cos(112,90278) \cdot \cos(113,54833) + \sin(112,90278) \cdot \sin(113,54833) \cdot \cos(3,42944)$$

$$\cos(a) = -0,3891686 \cdot (-0,3995224) + (0,9211665 \cdot (0,9167234) \cdot (0,9982092))$$

$$\cos(a) = 0,1554815 + 0,8429427$$

$$\cos(a) = 0,9984242$$

$$\arccos(0,9984242) = 3,2169287$$

$$2 \cdot \pi \cdot \text{Raio da Terra} = 6,28 \cdot 6.371 = 40.030 \text{ Km}$$

$$360 \text{ graus} = 40.030 \text{ Km}$$

$$3,2169287 = x$$

$$x = (40.030 \cdot 3,2169287) / 360 = 357,68 \text{ Km}$$

Com o uso destas equações pode-se realizar a transformação dos dados. Os mesmos podem ser armazenados em uma base de dados para posterior consulta. A seguir é apresentada a ferramenta de banco de dados MySQL utilizada neste projeto para o desenvolvimento do banco de dados.

### **2.3. Banco de Dados MySQL**

Um banco de dados é uma coleção organizada de dados. Um banco de dados é gerido por um sistema de gerenciamento de banco de dados (SGBD) que fornece maneiras de armazenar, organizar, modificar e recuperar os dados armazenados. Para o acesso e manipulação dos dados a maioria dos bancos usa a linguagem SQL um padrão adotado praticamente por todos os bancos de dados.

Para este projeto de conclusão será usado o banco de dados MySQL. Um banco de dados de código aberto de livre distribuição. [Livro Tradução (MySQL)]

## 2.4. Plataforma Java ME

A Plataforma Java Micro Edition ( Java ME ) prove uma forma robusta e flexível de se programar para dispositivos moveis. A linguagem e voltada para dispositivos com menos poder computacional e capacidade limitada de memória. Aplicações feitas em Java Me são portáteis entre dispositivos aproveitando as capacidades nativas de cada dispositivo. [ Livre tradução – (Oracle)]

O Java ME se segmenta em duas partes, para atender a características diferentes de dispositivos.

- CDC ( Connected Device Configuration), voltada para dispositivos com poder de processamento e memória mais altos.

CLDC ( Connected Limited Device Configuration) contém uma API mínima para executar em aplicativos com restrição de recursos.

Um aplicativo Java ME inclui uma configuração como CLDC e um profile como o Mobile Information Device Profile (MIDP). Um profile e um conjunto de APIs padrão que suportam uma estreita categoria de dispositivos. Um aplicativo Java Me e comumente chamado de MIDlet.

Para armazenar os dados utilizados no projeto foi utilizado o pacote Record Management System (RMS). Este pacote prove um mecanismo de persistência para o MIDlet.

## 2.5. Rede de Transmissão

A transmissão dos dados e feita através da rede de telefonia móvel, utilizando a 3ª Geração (3G) de rede de celulares. O Universal Mobile Telecommunications System (UMTS) é o termo adotado para designar o padrão de 3ª Geração estabelecido para a rede das operadoras de celular como evolução para operadoras de GSM e que utiliza como interface rádio o Wideband CDMA (WCDMA) e suas evoluções.O mais básico é o WCDMA (Wideband Code Division Multiple Access), que oferece taxas de transmissão de até 384 Kbits. Apesar de no papel o valor ficar próximo dos 236.8 Kbits oferecidos pelo EDGE, na prática o WCDMA oferece tempos de latência muito melhores e uma conexão muito mais utilizável. Um bom exemplo da diferença é o uso de aplicativos de VoIP, que são quase

inutilizáveis no EDGE devido ao lag na transmissão, mas fluem de forma satisfatória no W-CDMA. [Hardware - 3G: UMTS, WCDMA, HSDPA e a questão das frequências, 23/04/2011]

Em seguida temos o HSDPA (*High-Speed Downlink Packet Access*), um protocolo mais recente, que reduz a latência e aumenta a taxa de download da rede de forma expressiva. Usando o HSDPA como protocolo de transporte, o UMTS suporta taxas de 1.8, 3.6, 7.2 e 14.4 Megabits, de acordo com a implementação usada pela operadora (no Brasil a versão de 7.2 Megabits é a mais comum). Naturalmente, a velocidade real variando de acordo com a qualidade do sinal e o número de usuários conectados à mesma estação de transmissão, mas ela é sempre bem mais alta que no WCDMA. [Hardware - 3G: UMTS, WCDMA, HSDPA e a questão das frequências, 23/04/2011]

## 2.6. WEB Service

Um *WEB Service* é uma interface que descreve uma coleção de operadores, que são acessíveis através da rede por uma mensagem XML padronizada. É um serviço descrito usando como padrão, uma notação XML chamada de descrição de serviço. Esta descrição por sua vez, cobre todos os detalhes necessários para interagir com o serviço, incluindo formato de mensagens, protocolos de transporte e localização dos serviços. A interface esconde os detalhes da implementação do serviço, permitindo ser usado independente do *hardware* ou do *software* em que seja utilizado e independentemente também da linguagem de programação que foi escrito. (Kreger, 2011)

### 2.6.1. SOAP

SOAP, originado do acrônimo inglês *Simple Object Access Protocol*, e em português Protocolo Simples de Acesso a Objetos, é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída. [Wikipédia —SOAP 22/04/2011].

O SOAP é um protocolo baseado em XML que permite que dois programas se comuniquem. Ele está no topo da camada de transporte de dados e não está vinculado a nenhum protocolo de transporte em particular. O SOAP serve basicamente para a troca de informações e/ou dados através de objetos criados em diversas linguagens de programação como Visual Basic, Java, C++, etc. Por exemplo, a camada SOAP pode ser adicionada a um objeto Java e CORBA criando um ambiente de computação distribuído comum para os dois

objetos, tornando os objetos CORBA acessíveis pelos objetos Java e vice-versa, sem envolver as particularidades de cada protocolo [W3S].

De acordo com o W3Schools [W3S] a estrutura da mensagem SOAP é definida em um documento XML que contém os seguintes elementos:

- Envelope: Identifica o documento XML como uma mensagem SOAP e é responsável por definir o conteúdo da mensagem;
- Header (opcional): Contém os dados do cabeçalho;
- Body: Contém as informações de chamada e de resposta ao servidor;
- Fault: Contém as informações dos erros ocorridos no envio da mensagem. Esse elemento só aparece nas mensagens de resposta do servidor.

Uma mensagem SOAP possui a seguinte estrutura:

```

<SOAP-ENV:envelope>
<!--Esse é o elemento raiz do SOAP e define que essa é uma mensagem SOAP-->
->
<SOAP-ENV:header>
<!--Especifica informações específicas como autenticação, etc.(é opcional)-->
</SOAP-ENV:header>
<SOAP-ENV:body>
<!--O elemento BODY contém o corpo da mensagem-->
<SOAP-ENV:fault>
<!--O elemento FAULT contém os erros que podem ocorrer-->
</SOAP-ENV:fault>
</SOAP-ENV:body>
</SOAP-ENV:envelope>

```

Abaixo um exemplo de uma mensagem SOAP criado para o projeto

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ws="http://www.e-making.com/projetoweb/ws/">
  <soapenv:Header/>
  <soapenv:Body>

```

```

<ws:envia>
  <usuario>?</usuario>
  <latitude>?</latitude>
  <longitude>?</longitude>
  <dtposicao>?</dtposicao>
</ws:envia>
</soapenv:Body>
</soapenv:Envelope>

```

### 2.6.2. WSDL

Um documento WSDL descreve os serviços do *WEB service* através do XML, fornecendo uma documentação do serviço para que possíveis clientes possam utilizá-lo de forma automatizada. A figura 2.4 mostra os principais elementos de um WSDL:

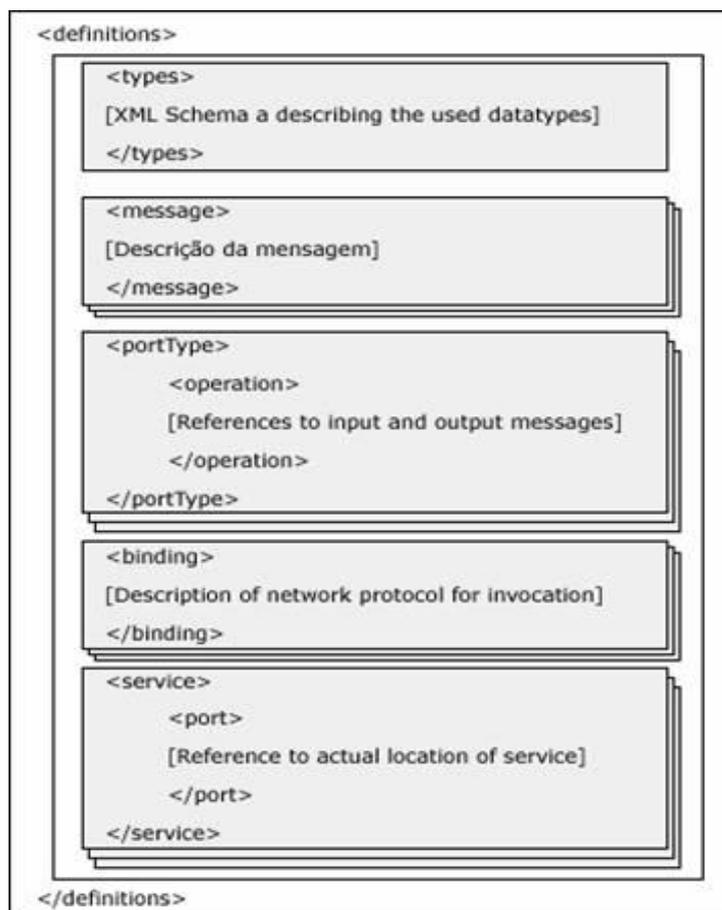


Figura 2.4 - Elementos WSDL (Fonte: [http://imasters.com.br/artigo/4422/webservices/descrevendo\\_um\\_web\\_service\\_-\\_wSDL/](http://imasters.com.br/artigo/4422/webservices/descrevendo_um_web_service_-_wSDL/))

Abaixo e descrito a utilidade de cada trecho do WSDL

<types> - Tipos de dados. Nesta parte são definidos os formatos dos dados de entrada e saída do *web service*.

<message> São os parâmetros de entrada e saída.

<portType> Agrupamento lógico de operações.

<binding> Mapeia os elementos *operation* em um elemento *portType*, para um protocolo específico. Associando o elemento *portType* ao protocolo SOAP, através do parâmetro de transporte.

<service> - Os elementos *service* e *port* definem a localização do serviço, descritos no elemento *binding*.

## 2.7. PHP

PHP, que significa "*PHP: Hypertext Preprocessor*" é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a WEB e pode ser mesclada dentro do código HTML. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente, mas você pode fazer muito mais do que isso com PHP. (PHP.NET)

O PHP "*PHP: Hypertext Preprocessor*" é uma linguagem de script open source voltada para a Web. A linguagem é suportada por diversos servidores de aplicação como IIS ( Internet Information Service) ou Apache.

## 2.8. Java Script

Inicialmente conhecida por LiveScript, JavaScript foi desenvolvido no Netscape para tornar seu browser mais poderoso e funcional, proporcionando a seus usuários uma experiência mais rica na web. Atualmente todos os browsers suportam JavaScript, apesar de haver diferenças de sintaxe entre um browser e outro. Especialmente entre o Internet Explorer e os demais. (Soares, 2006)

Um script JavaScript permite que sejam executadas funções que antes, obrigatoriamente, deviam ser processadas pelo servidor, aberturas de novas janelas, criação e manipulação de camadas, cálculos simples e complexos, etc. (Soares, 2006)

O JavaScript possui uma estrutura de linguagem igual às das outras linguagens clássicas ( Delphi, Visual Basic, C++), ou seja nela existe o conceito de variáveis, funções, estruturas de controle, operadores, além de funções características para manipulação de elementos do browser, tais como escrever no documento HTML, manipular janelas, manipular elementos de um formulário, etc. (Soares, 2006)

## 2.9. AJAX

AJAX é um acrônimo que significa *Asynchronous JavaScript and XML* ( JavaScript e XML assíncrono), porem temos muito mais que a junção do JavaScript com XML, é todo um conceito de navegação e atualização de páginas web. Algumas funções oferecidas pelo AJAX não são novas. O AJAX acima de tudo é uma técnica que oferece a junção de tecnologias usadas há bastante tempo só que de forma separada. O modelo é baseado nos padrões da internet, combinando as funcionalidades CSS (*Cascading Style Sheets*), JavaScript e XML.

O AJAX trouxe à possibilidade de reduzir a transferência de dados entre o servidor e o navegador. As paginas já processadas podem ser atualizadas sem a necessidade de se carregar a pagina novamente. Para isso é feito um acesso ao servidor através do método do JavaScript chamado *XMLHttpRequest* recebendo sua resposta com o método *XMLHttpRequest*. A figura 2.5 abaixo ilustra como funciona uma chamada.

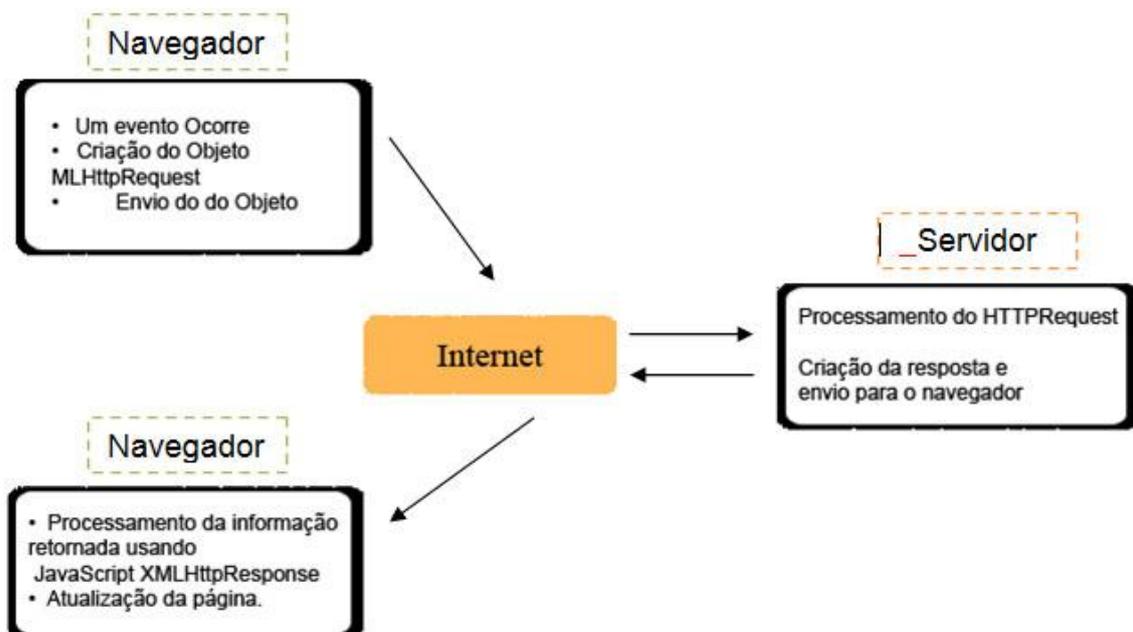


Figura 2.5 - Chamada Ajax

## 2.10. API Google Maps

A empresa Google disponibiliza gratuitamente a API ( *Application Programming Interface* ) do Google Maps, com esta é possível fazer uso dos mapas e fotos de satélites existentes. Sendo possível definir rotas e marcar posições no mapa.

Esta API roda pela Web, sendo necessário conhecimento na linguagem de programação Java Script e programação orientada a Objeto. Com isto temos acesso a vários serviços disponíveis para a manipulação de mapas, cálculo de distância entre dois pontos, melhor rota, e vários outros.

Com o uso destas ferramentas e conceitos apresentados neste capítulo, pode-se especificar o modelo proposto para o projeto, conforme será exposto no capítulo 4 a seguir.

### 3. PROPOSTA E SOLUÇÃO DO MODELO

Neste capítulo é apresentado a arquitetura do projeto, descrevendo os passos do projeto, componentes utilizados e sua montagem. Evidenciando cada parte necessária para o funcionamento do modelo.

#### 3.1. Modelo Proposto

O projeto proposto tem como objetivo desenvolver uma solução que seja capaz de obter as coordenadas geográficas através do GPS do celular e enviá-las para um banco de dados na internet. Com isso é possível acessar uma página na internet e visualizar a localização do aparelho. O modelo proposto contempla também a possibilidade de serem criados grupos. Com isso quem estiver no mesmo grupo poderá saber a distancia na qual outro celular membro do mesmo grupo se encontra.

Para que esse objetivo seja alcançado foi definido o seguinte modelo de implementação, apresentado na figura 3.1:

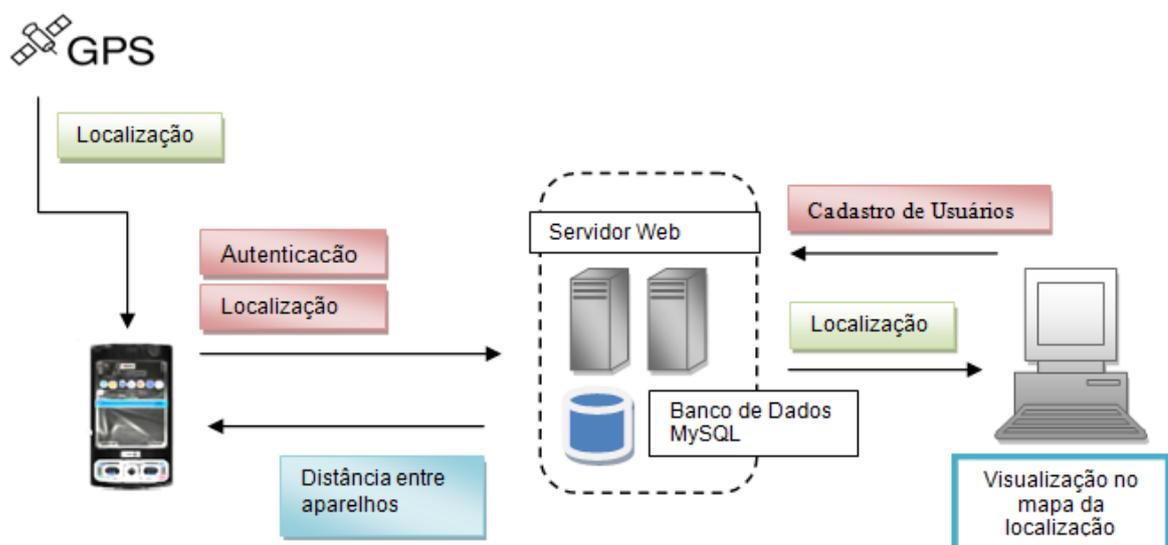


Figura 3.1 - Topologia do Projeto

### 3.2. Plataforma Móvel Utilizada

O equipamento escolhido para o desenvolvimento do projeto foi o *smartphone* da marca Nokia, modelo N95 8GB. Na figura 3.2 é mostrado o aparelho.



Figura 3. 2 - Nokia N958G (Fonte: <http://www.nokia.com.br/produtos/celulares/nokia-n95-8gb>)

Um dos motivos da escolha do aparelho foi sua ampla gama de funções e também por já possuir o aparelho. Eliminando a necessidade de adquirir um aparelho somente para o projeto.

O N95 8G utiliza-se sistema operacional Symbian OS, um sistema operacional bastante estável, e que não exige muitos recursos para funcionar. Na figura 4.3 são mostradas as características do N95 8G.

Tabela 3.1- Especificações Técnicas do Nokia N95 8G

Funções	Característica
Software	<ul style="list-style-type: none"> <li>▪ Sistema operacional: Symbian OS ver. 9.2</li> <li>▪ Interface de usuário: S60 3a Edição, Pacote de Funções (S60 3.1)</li> <li>▪ Java :MIDP 2.0</li> <li>▪ SDKs C++ e Java</li> </ul>
Rede	<ul style="list-style-type: none"> <li>▪ WCDMA HSDPA 850/1900 MHz (na versão N95-4) com conexão simultânea de voz e pacotes de dados (PS velocidade máx. DL/UL= 3.6Mbps/384kbps, CS velocidade máx. 64kbps).</li> <li>▪ GPRS classe B, multi portas classe 32 (5 Rx +3 Tx / Máx Sum 6), velocidade máxima DL/UL= 107 / 64.2 kbits/s</li> </ul>

Fonte: <http://www.nokia.com.br/produtos/celulares/nokia-n95-8gb/especificacoes-tecnicas>. Com adaptações

No celular é instalada a interface WEB. Que por sua vez é responsável por obter as informações de coordenada e enviá-las para o servidor WEB descrito no próximo tópico.

### 3.3. Plataforma WEB

Para receber e armazenar as informações enviadas pelo celular se faz necessário o uso de um servidor, que fique disponível de forma ininterrupta. Assim, para atender essa necessidade especificou-se para o projeto um servidor web, no qual está hospedado, disponível para acesso via internet. O projeto roda em servidor apache<sup>1</sup> que é responsável por executar o software escrito em PHP. No servidor também fica hospedado o banco de dados utilizado no projeto.

### 3.4. Software Desenvolvido

O software desenvolvido para o projeto foi desenvolvido utilizando-se Java Me, JavaScript e PHP. A programação envolvida foi dividida em duas partes fundamentais: a primeira que deve ser instalada no dispositivo de celular, foi programada em Java Me e a segunda parte, que se encontra em um servidor WEB foi programada em PHP. A parte do servidor WEB faz uso de componentes em JavaScript e AJAX, que torna mais dinâmica a atualização dos dados. Para armazenar os dados recebidos, foi utilizado o banco de dados MySQL.

Para realizar a comunicação entre a interface WEB e a PHP, foram utilizados *WEB services*. O que elimina a necessidade reprogramação caso o software seja portado para outras plataformas.

Na figura 3.3 é mostrada a visão geral do software desenvolvido.

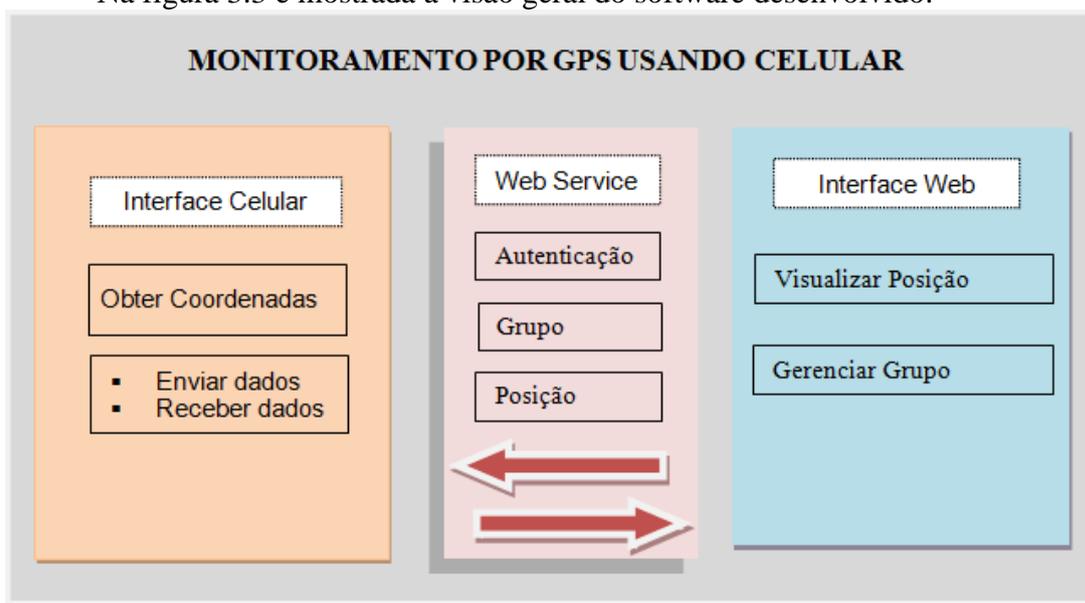


Figura 3. 3 - Visão Geral do Software

<sup>1</sup> Apache um servidor web livre, desenvolvido pela Apache Software Foundation.

O software instalado no celular verifica se já tem informações do usuário salvas em sua memória. Existindo estas informações e solicitado a validação por meio do *WEB service* de autenticação. Caso as informações estejam corretas, recebe a resposta com o número de identificação do usuário.

Com a autenticação feita corretamente o software começa a funcionar, capturando as coordenadas por meio do GPS interno do celular. Após capturadas as coordenadas são enviadas para o servidor web onde serão armazenadas. O software solicita informações ao servidor sobre outros celulares do grupo, exibindo na tela do celular o nome dado ao usuário, a distancia, e o tempo de quando recebeu a ultima coordenada. A figura 3.4 ilustra o diagrama de fluxo do software.

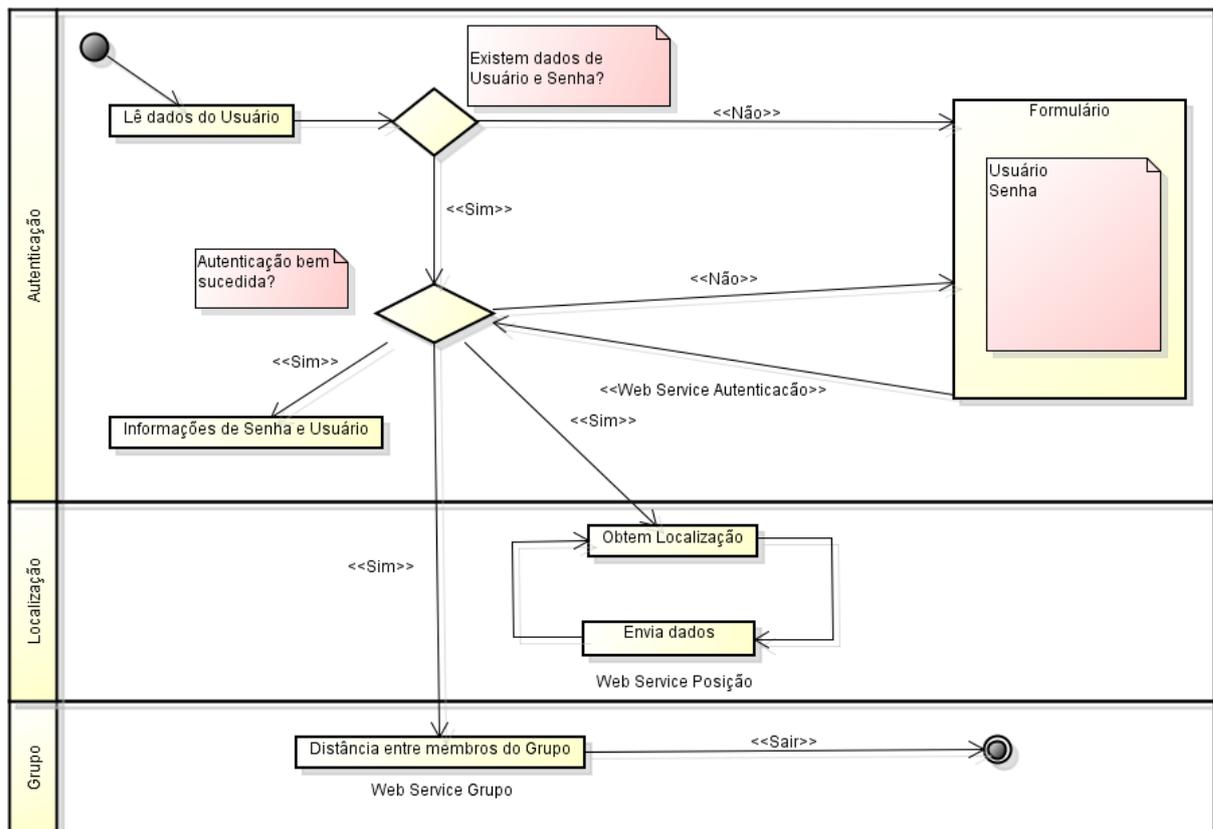


Figura 3.4 - Fluxo de funcionamento da Interface do Celular

Na parte web do software, após fazer a autenticação. O usuário é redirecionado para o mapa onde são mostradas as últimas coordenadas enviadas pelo celular, sendo possível selecionar períodos para a visualização. Selecionando o menu grupos e possível adicionar outros usuários em um grupo. E visualizar qual a última posição do usuário do grupo. Mostrada na figura 3.5 a seguir.

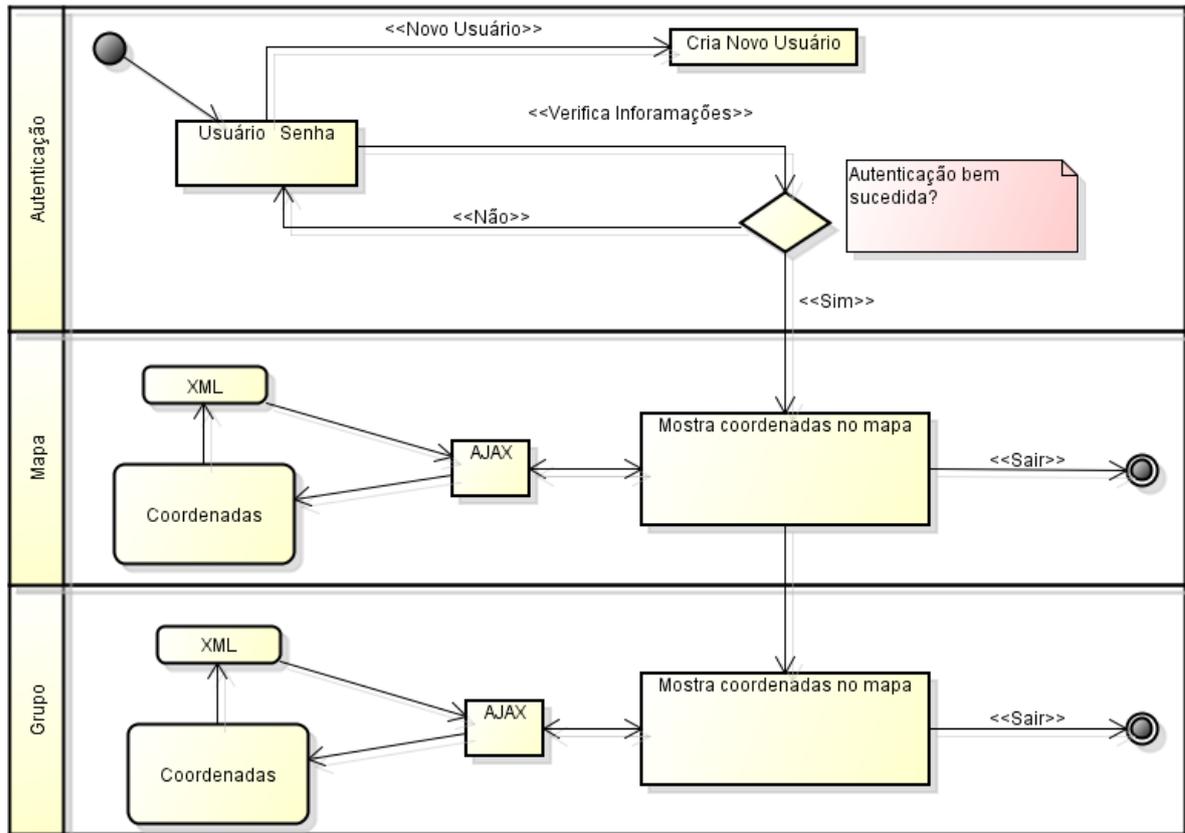


Figura 3.5 - Fluxo de funcionamento da Interface Web

### 3.5. Interface do Celular

O software desenvolvido para o celular foi escrito em Java Me. A linguagem Java Me foi escolhida devido a sua portabilidade da linguagem, tornando possível o uso em outros celulares que além do modelo escolhido para o projeto. No projeto existiu sempre a preocupação de se utilizar apenas bibliotecas nativas do Java Me. Isso possibilita o uso em outros aparelhos com as mesmas características sem necessidade de se modificar o código.

E de responsabilidade do software, efetuar a autenticação do usuário no sistema, salvar usuário e senha, enviar informações de posição e receber a distancia em relação a outros usuários.

Para facilitar a compreensão o software foi organizado em 3 pacotes. São eles:

- principal - classes responsáveis pela apresentação de dados ao usuário.
- comunicação - classes responsáveis pela comunicação do software com o servidor WEB.
- dados - classes responsáveis em obter ou salvar dados.

### **3.5.1. Pacote Principal**

#### **3.5.1.1. Classe Principal**

A classe principal é responsável pelo controle do software e de apresentar graficamente as informações ao usuário. Além disso, é responsável por buscar por dados do usuário salvos, iniciar o GPS, e realizar a comunicação com o servidor web.

Existem diversos métodos na classe principal, porém somente serão apresentados métodos que possuam maior relevância para o entendimento.

Ao se criar um aplicativo em Java Me é necessário que a classe principal seja uma subclasse de `javax.microedition.midlet.MIDlet`. Essa por sua vez define métodos que correspondem ao ciclo de vida da aplicação. Um dos métodos definidos é o `startApp()`, responsável por executar tarefas de inicialização e iniciar a interface com o usuário. Os dados são exibidos na tela através de um *form* que contém os objetos de texto dentro da aplicação.

Para que os comandos do menu sejam capturados implementamos a interface *CommandListener*, essa interface define um método chamado *commandAction* que é chamado sempre que algum elemento do menu sofre uma iteração.

### **3.5.2. Pacote de Dados**

No pacote de dados estão as classes responsáveis por obter os dados do GPS e por salvar as informações e senha de usuário.

#### **3.5.2.1. Classe getLocalizacao**

A classe `getLocalizacao` é responsável por acionar o GPS do celular, e obter as coordenadas. Ela é implementada como um processo separado da classe principal. Isso permite que a classe seja acessada sem concorrer com outros processos.

#### **3.5.2.2. Classe Dados**

A classe de dados é responsável em armazenar e recuperar os dados de senha e usuário. Foi implementada a opção de se salvar o usuário e a senha, para eliminar a necessidade

de se inserir a senha toda vez que o programa for iniciado. Para isso foi utilizado a biblioteca Record Management System (RMS).

Inicialmente se pensou em salvar estas informações utilizando um arquivo de texto. A idéia foi descartada devido ao fato de funcionar somente em aparelhos que contenham sistema de arquivo.

### **3.5.3. Pacote de Comunicação**

No pacote de comunicação estão armazenadas as classes responsáveis pelo funcionamento dos *WEB services*. No projeto são utilizados três *WEB services* são eles posição, autenticação e grupo. O *WEB service* de posição é responsável por enviar as informações de posição do aparelho. O de autenticação cuida da autenticação propriamente dita. O de grupo é responsável em retornar os usuários cadastrados mostrando a distância e o tempo no qual a última posição foi enviada.

## **3.6. Aplicativo PHP - Interface Web**

A interface WEB também chamada de sítio é responsável por, fazer a autenticação do usuário, receber os dados de posição e armazenar em um banco de dados, organização do grupo, calcular a distância entre outros usuários do mesmo grupo.

### **3.6.1. Estrutura**

Para facilitar o entendimento do projeto foram criadas pastas que dividem os códigos em PHP criados conforme sua utilização.

- ws – contém as classes dos *web services* e seus descritores.
- js – Contem os arquivos Java Script.
- include – contém as classes de conexão com o banco, e outras classes necessárias ao funcionamento do projeto.

### **3.6.2. Banco de dados**

O banco de dados utilizado para o armazenamento das informações é o MySQL 5. Nele contém 3 tabelas, são elas:

- tb\_usuario - guarda informações da conta do usuário.

- tb\_grupo - guarda as informações de grupo.
- tb\_posicao - armazena informações de posição.

Na figura 3.6 é mostrada a modelagem realizada para o projeto.

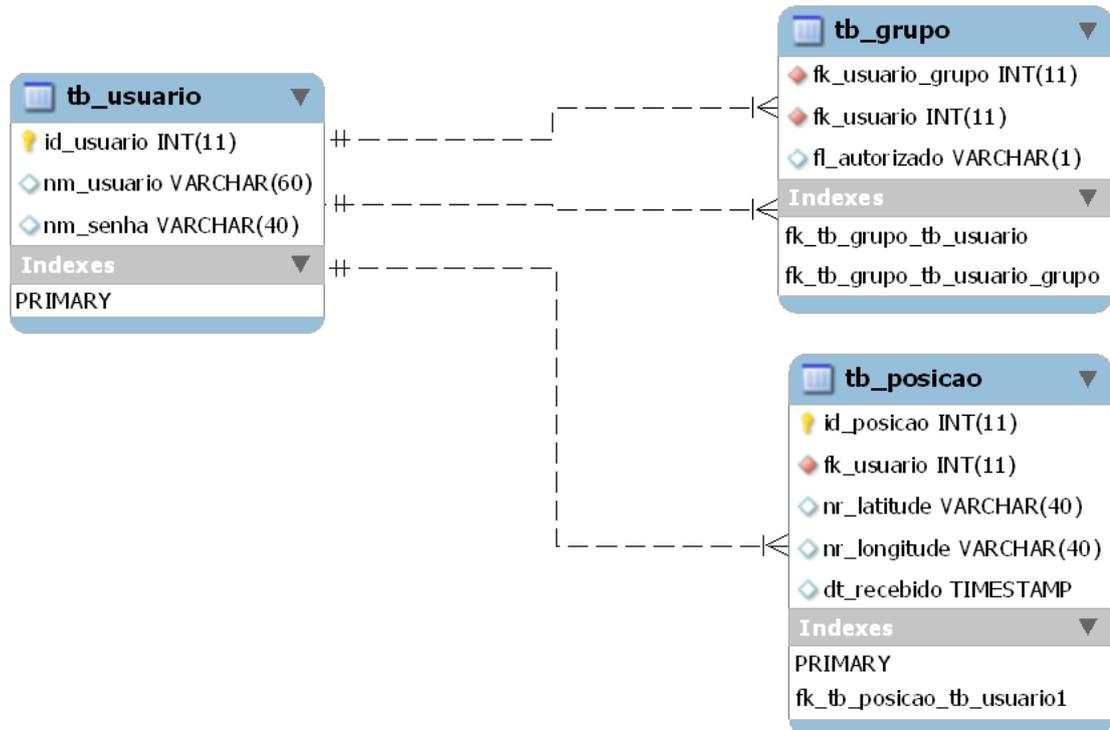


Figura 3.6 - Modelagem do Banco MySQL

### 3.6.3. Gerador de XML

Ao se executar uma consulta no banco de dados por um código em PHP, a resposta fornecida será um *array*. Não é possível acessar esta variável array, porque todo código PHP é executado no servidor. Enviando somente HTML como resposta. Como exemplo, a resposta à consulta da tabela de grupos, transformando o array em HTML fornece uma resposta como na figura 3.7.

```
array(6) { [0]=> string(1) "1" ["USUARIO"]=> string(1) "1" [1]=> string(1)
"3" ["MEMBRO"]=> string(1) "3" [2]=> string(1) "S"
["AUTORIZADO"]=> string(1) "S" } array(6) { [0]=> string(1) "1"
["USUARIO"]=> string(1) "1" [1]=> string(1) "4" ["MEMBRO"]=>
string(1) "4" [2]=> string(1) "S" ["AUTORIZADO"]=> string(1) "S" }
```

Figura 3.7 - Resposta da consulta em formato de array.

Esta não é uma forma fácil de ser trabalhada, por isso é necessário sua transformação em algo mais amigável. Como o projeto contempla o uso de AJAX para sua atualização, foi escolhido transformar esta resposta em XML. Surgindo assim a necessidade de se criar um gerador de XML. Sua função consiste em transformar a resposta da consulta ao banco de dados que está em array em um arquivo XML. Na figura 3.8 é ilustrada a mesma resposta em formato XML.

```

- <dados>
  - <linha>
    <usuario>1</usuario>
    <membro>3</membro>
    <autorizado>S</autorizado>
  </linha>
  - <linha>
    <usuario>1</usuario>
    <membro>4</membro>
    <autorizado>S</autorizado>
  </linha>
  - <linha>
    <usuario>1</usuario>
    <membro>5</membro>
    <autorizado>S</autorizado>
  </linha>
+ <linha></linha>
</dados>

```

Figura 3.8 - Resposta em formato XML

Além de mais fácil de ser analisada, pode-se visualmente, podemos usar métodos já existentes que manipulam XML. Com isso pode-se fazer uso das funções do AJAX para manipular os dados como desejarmos.

#### 3.6.4. Web Service

Sem dúvida, a parte que apresentou maior grau de dificuldade no desenvolvimento do projeto foi a do WEB service. Seja pela complexidade do assunto, ou pela grande flexibilidade de linguagens passíveis de utilização. Foi escolhido *WEB service* devido a sua alta portabilidade, e o fato de uma vez pronto outras linguagens podem utilizá-los sem muitas complicações. Utilizando outros métodos seria necessário uma modificação tanto na

programação da parte web quanto no programa escrito para o celular, quando o programa fosse portado para outras linguagens.

Um *WEB service* é composto de duas partes, a primeira é a função propriamente dita, responsáveis pela as ações executadas. A segunda parte a descrição do *WEB service* que é chamado de WSDL.

Agora será explicado com mais detalhes como foi construído o WEB service. No projeto foi adotado o seguinte padrão, para a construção das classes:

- classe entrada - estão os atributos de entrada que devem ser preenchidos.
- classe resposta - a resposta devolvida.

Para o WSDL, o grande problema encontrado foi achar um modelo de descritor que funcionasse com Java Me. Existem diversos programas prontos que fazem o trabalho de gerar o WSDL, mas nenhum funcionava ao se tentar acessar por Java Me. Isso levou a uma pesquisa um pouco mais aprofundada em como criar esse descritor.

O que deve ficar bem claro é a ligação entre o WSDL e código do serviço descrito. Tomando-se como exemplo o WEB service que envia a posição mostrado na figura 3.9.

```
- <wSDL:definitions name="posicao" targetNamespace="http://www.e-making.com/projetoweb/ws/">
+ <wSDL:types></wSDL:types>
+ <wSDL:message name="enviaRequest"></wSDL:message>
+ <wSDL:message name="enviaResponse"></wSDL:message>
+ <wSDL:portType name="posicao"></wSDL:portType>
+ <wSDL:binding name="posicaoSOAP" type="tns:posicao"></wSDL:binding>
+ <wSDL:service name="posicao"></wSDL:service>
</wSDL:definitions>
```

Figura 3 9- WSDL do web service de posição

O elemento *service*, é responsável por dizer onde está localizado o serviço. No caso o arquivo é o `posicao.php`. Este elemento se liga ao elemento *binding*, através do `tns:posicaoSOAP`. O elemento *service* é mostrado na figura 3.10.

```

+ <wsdl:binding name="posicaoSOAP" type="tns:posicao"></wsdl:binding>
- <wsdl:service name="posicao">
  - <wsdl:port binding="tns:posicaoSOAP" name="posicaoSOAP">
    <soap:address location="http://www.e-making.com/projetoweb/ws/posicao.php"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Figura 3.10 - Elemento service

O elemento *binding* mostrado na figura 3.11, especifica qual protocolo de transporte será usado e qual operação será realizada. O elemento *operation* diz o que será executado ao se enviar a mensagem, no caso a função **envia**. Na figura 3.12 é mostrada a classe que o WEB service faz referência.

```

- <wsdl:binding name="posicaoSOAP" type="tns:posicao">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  - <wsdl:operation name="envia">
    <soap:operation soapAction="http://www.e-making.com/projetoweb/ws/envia"/>
  - <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  - <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
+ <wsdl:service name="posicao"></wsdl:service>
</wsdl:definitions>

```

Figura 3.11 - Elemento binding

```

class coordenada implements posicao {

    public function envia($parametros) {

        return $saida;
    }

}

```

Figura 4.12 - Classe php

O elemento *portType* na figura 3.13, especifica que a função de envio vai ter uma entrada e uma saída.

```

- <wsdl:portType name="posicao">
  - <wsdl:operation name="envia">
    <wsdl:input message="tns:enviaRequest"/>
    <wsdl:output message="tns:enviaResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Figura 3. 13 - Elemento portType

Os elementos *message* na figura 3.14, dizem o que objetos serão usados de entrada e saída na função. É fácil observar as mensagens tem elementos *tns:envia* e *tns:enviaResponse*, que são descritos no elemento *types*.

```

- <wsdl:message name="enviaRequest">
  <wsdl:part element="tns:envia" name="parameters"/>
</wsdl:message>
- <wsdl:message name="enviaResponse">
  <wsdl:part element="tns:enviaResponse" name="parameters"/>
</wsdl:message>

```

Figura 3. 14 - Elemento message

No elemento *types* na figura 3.15, neles são definidos os tipos de retorno. No caso ele envia e retorna uma classe, como pode ser visto na figura 3.16.

```

- <wsdl:types>
  - <xsd:schema targetNamespace="http://www.e-making.com/projetoweb/ws/">
    <xsd:element name="envia" type="tns:entrada"> </xsd:element>
    <xsd:element name="enviaResponse" type="tns:resposta"> </xsd:element>
  - <xsd:complexType name="entrada">
    - <xsd:sequence>
      <xsd:element name="usuario" type="xsd:string"/>
      <xsd:element name="latitude" type="xsd:string"/>
      <xsd:element name="longitude" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  - <xsd:complexType name="resposta">
    - <xsd:sequence>
      <xsd:element name="msgResposta" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>

```

Figura 3. 15 - Elemento types

```
class entrada {  
  
    public $usuario;  
    public $latitudo;  
    public $longitudo;  
  
}  
  
class resposta {  
  
    public $msgResposta;  
  
}
```

Figura 3.16 - Classes de entrada e resposta

O WEB service e iniciado da seguinte forma.

```
$server = new SoapServer("autenticar.wsdl");  
$server->setObject(new autenticar());  
$server->handle();
```

O método *SoapServer* inicia o objeto, observando que iniciamos o objeto com um construtor que indica qual WSDL vai descrever o serviço. Depois disso e adicionado a classe que vai processar as informações pela função *setObject*. A função *handle* fica responsável em processar requisições SOAP, chamar as funções necessárias e enviar a resposta de volta.

## 3.7. Apresentação das coordenadas

### 3.7.1. Atualização

As atualizações das coordenadas são refletidas nas paginas através de funções Java Script que utilizam AJAX. Em intervalos de um minuto, a função de atualização é chamada. Essa por sua vez faz a consulta ao banco de dados retornando as últimas coordenadas.

### **3.7.2. Autenticação**

A parte de identificação do usuário é feita por meio do arquivo index.php. Ao se acessar o endereço do projeto este arquivo é chamado, sendo apresentado um formulário com usuário e senha. Ao clicar em entrar o arquivo valida as informações\_redirecionando o usuário para a pagina do mapa. Na pagina também é possível cadastrar um novo usuário.

### **3.7.3. Mapa**

Na pagina do mapa são apresentadas as coordenadas que o aparelho registrou. Nela e possível escolher entre as posições recebidas em um dia.

### **3.7.4. Grupo**

Nesta pagina é mostrada a ultima posição dos membros do grupo, e também a opção de adicionar novos membros.

Para se adicionar um usuário basta inserir o nome do usuário e clicar em adicionar, com isso será solicitada uma permissão ao usuário, ela sendo aceita ambos os usuários podem ver suas posições.

## 4. APLICAÇÕES E RESULTADOS

Neste capítulo é mostrado o funcionamento do projeto, e os resultados obtidos. Outro item abordado neste capítulo trata de alguns problemas enfrentados durante o desenvolvimento.

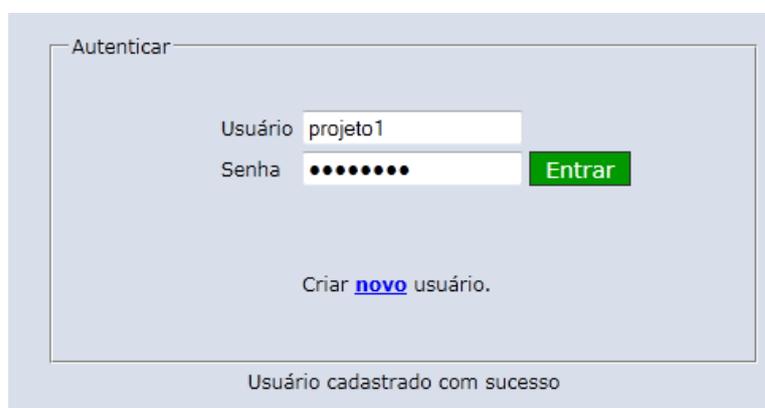
### 4.1. Descrição da Aplicação do Modelo

Após a conclusão da implementação das funcionalidades do software foi possível iniciar os teste. Como foi convencionado que seria necessária á autenticação, o primeiro passo para a execução correta é a criação do usuário no site WEB. Para isso acessa-se a página principal do projeto e criam-se dois usuários. A necessidade do segundo usuário é para que a funcionalidade de grupo possa ser observada. Para criar um usuário basta preencher usuário e senha, depois clicar em novo. Se o usuário já existir será mostrado uma mensagem de erro, como na figura 4.1. A figura 4.2 ilustra a criação do usuário.



A interface de autenticação, intitulada "Autenticar", contém dois campos de entrada: "Usuário" com o valor "projeto1" e "Senha" com caracteres ocultos por pontos. Um botão verde "Entrar" está à direita do campo de senha. Abaixo dos campos, há o texto "Criar [novo](#) usuário.". Na base da interface, uma mensagem de erro indica "Usuário já existe".

Figura 4. 1 - Erro ao criar um usuário duplicado



A interface de autenticação, intitulada "Autenticar", contém dois campos de entrada: "Usuário" com o valor "projeto1" e "Senha" com caracteres ocultos por pontos. Um botão verde "Entrar" está à direita do campo de senha. Abaixo dos campos, há o texto "Criar [novo](#) usuário.". Na base da interface, uma mensagem de sucesso indica "Usuário cadastrado com sucesso".

Figura 4.2 - Criação de usuário

Seguindo a aplicação do modelo devemos instalar o software no celular. Como o software foi desenvolvido em Java, temos um arquivo jar para instalar. Para isso basta copiar o arquivo projeto.jar para o celular e executar a instalação.

Concluída a instalação é necessário executar um procedimento extra. Esse procedimento visa liberar o acesso do software ao GPS e ao envio de dados. Este procedimento contorna um problema encontrado. Ao se acessar funções como GPS é exigida a confirmação do usuário. Como a interface executa consultas periódicas se tornava necessário confirmar a cada leitura de coordenadas como mostrado na figura 4.3.



Figura 4. 3 Confirmação do uso de recursos do Celular

Essa persistência na confirmação a cada leitura, deve-se ao aplicativo não ser assinado digitalmente. Para realizar esta assinatura do aplicativo o custo é de \$299 dólares (Thawte Certificação Fonte: <https://www.thawte.com/code-signing/content-signing-certificates/sun-java/index.html>, acessado em 02/06/2011). O problema foi contornado realizando uma configuração no gerenciador de aplicativos do celular. Para efetuar essa configuração é necessário entrar no gerenciamento de software do celular e realizar a configuração para que seja necessária somente a primeira confirmação. Deve-se permitir o uso do GPS e da rede de dados como mostrado na figura 4.4.



Figura 4. 4 - Configuração do software

Realizados os procedimentos iniciais, pode-se iniciar o uso do aplicativo. O uso do software é bastante simples. Iniciada a interface do celular será solicitado que seja inserido usuário e senha, como na figura 4.5. Neste momento será consultado o banco de dados do servidor web para que os dados sejam verificados. Se os dados não estiverem corretos uma mensagem de erro é apresentada como na figura 4.6.

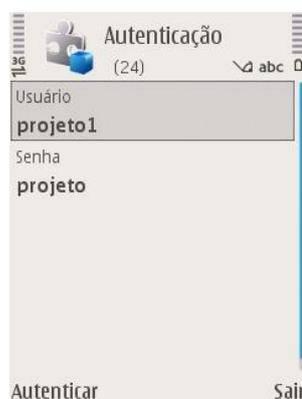


Figura 4. 5 - Tela de autenticação

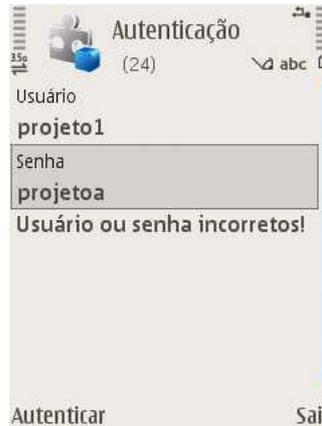


Figura 4. 6 - Erro ao se autenticar

Tendo uma resposta positiva do servidor, é retornado o número de identificação do usuário. Este número será usado posteriormente para identificar que a coordenada enviada pertence ao usuário. Depois que a autenticação é efetuada o usuário é redirecionado para a tela principal do projeto, mostrada na figura 4.7. Neste momento é iniciado um processo paralelo que é responsável por obter as informações do GPS, enviar para o banco de dados e fazer a atualização dos usuários que pertencem aos grupos. Na figura 4.7 é mostrado com os dados de grupo carregados. Com isso o *WEB service* de grupo é utilizado, a distância retornada e calculada através da formula descrita no capítulo 3 sendo retornado o valor em quilômetros, o tempo é definido por uma subtração entre as datas nas quais as coordenadas foram obtidas. Se a diferença de tempo for menor que um dia são exibidas em horas:minutos:segundos, se for maior é exibida em dias.



Figura 4. 7 - Tela Principal do Projeto

Através da interface WEB é possível visualizar as posições enviadas. Para isso é necessário acessar a página principal da aplicação, usa-se o mesmo usuário utilizado para capturar os dados. Depois de efetuada a autenticação, a aplicação redireciona para a tela onde são mostradas as coordenadas obtidas por data. A última coordenada recebida fica sendo o foco do mapa, ao se passar o mouse sobre o ponto e mostrada a ordem que ela foi recebida. Como visto na figura 4.8, ao se passar o mouse em cima do ponto é mostrado em quando foi capturado e suas coordenadas.

Ordem	Latitude	Longitude
1	-46.660213230922004	-23.624645324176
2	-46.660617825235	-23.624975487217
3	-46.660757551508006	-23.625112531282003
4	-46.660210464895	-23.624924776722
5	-47.89103087293	-15.769350093974001
6	-47.886500204523	-15.761026531998
7	-47.893455253686	-15.767954675262

Figura 4. 8 - Tela Inicial

A pagina de controle dos grupos pode ser acessada através do link grupo. Nela são listados outros usuários adicionados, o tempo da última captura e a distância em que se encontra o usuário da posição atual. Para adicionar um usuário basta digitar o nome do usuário e clicar em adicionar. Só será possível visualizar a última coordenada e a distância após a aprovação do usuário. A tela de grupo e mostrada na figura 4.9.

The screenshot displays the 'Projeto Final' application interface. On the left, a map shows a location with a red pin. A tooltip for 'projeto1' provides the following data: Data: 14/06/2011 20:08:14, Latitude: -46.708324834008, and Longitude: -23.608790038317. The right panel features a blue header with the text 'Projeto Final' and 'Monitoramento por GPS usando celular'. Below the header are buttons for 'Mapa', 'Grupo', and 'Sair'. The 'Usuário:' field is set to 'projeto1'. A 'Grupo' table lists 'projeto2' with a distance of 881,897 Km and a duration of 4 dias. There is a 'Convidar' field and an 'Adicionar' button. A 'Pendentes' table lists 'projeto3' with a 'Negar' button.

**projeto1**  
 Data: 14/06/2011 20:08:14  
 Latitude: -46.708324834008  
 Longitude: -23.608790038317

**Projeto Final**  
 Monitoramento por GPS usando celular

Mapa Grupo Sair

Usuário: projeto1

Usuário	Distância	Tempo
projeto2	881,897 Km	4 dias

Convidar Adicionar

Usuário	
projeto3	Negar

Figura 4. 9 - Visão do Grupo

Depois de explicado como manipular o sistema, pode-se iniciar o teste prático. Primeiramente foi testado com o aparelho parado, sendo mostrada a coordenada como na figura 4.10.

The screenshot displays the 'Projeto Final' application interface. On the left, a map shows a location with a red pin. The right panel features a blue header with the text 'Projeto Final' and 'Monitoramento por GPS usando celular'. Below the header are buttons for 'Mapa', 'Grupo', and 'Sair'. The 'Escolha uma data:' dropdown is set to '05.06.2011'. A 'Coordenadas' table lists two entries with their respective latitude and longitude values.

**Projeto Final**  
 Monitoramento por GPS usando celular

Mapa Grupo Sair

Escolha uma data: 05.06.2011

Ordem	Latitude	Longitude
1	-46.706889936547	-23.611075950085002
2	-46.707517657038004	-23.609602914979

Figura 4. 10 - Captura de coordenada parado

A seguir, foi realizado um teste de percurso. Os resultados são satisfatórios, sendo possível observar o percurso percorrido pelo aparelho de forma bem clara. No exemplo da figura 4.11 é mostrado o trajeto em Brasília, desde o aeroporto Juscelino Kubitschek até a 407 sul, no dia 16/04/2011.

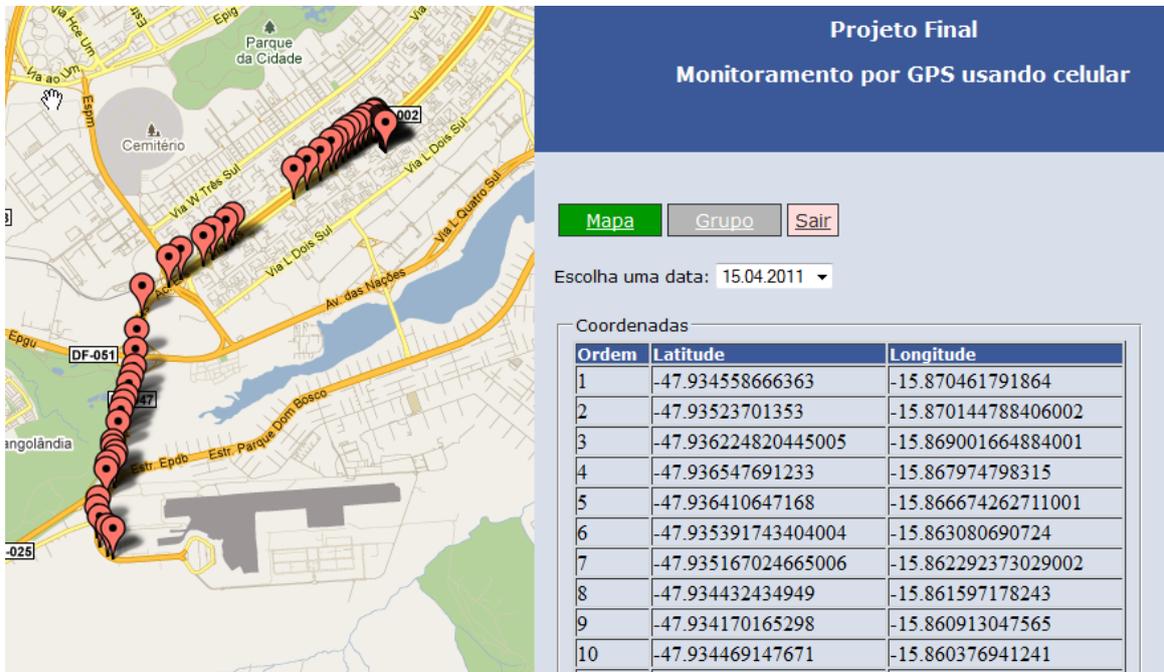


Figura 4. 11 - Trajeto Aeroporto até a 407 Sul

Foi também testado, também um trajeto iniciando da 407 sul até o UniCEUB, mostrado na figura 4.12.

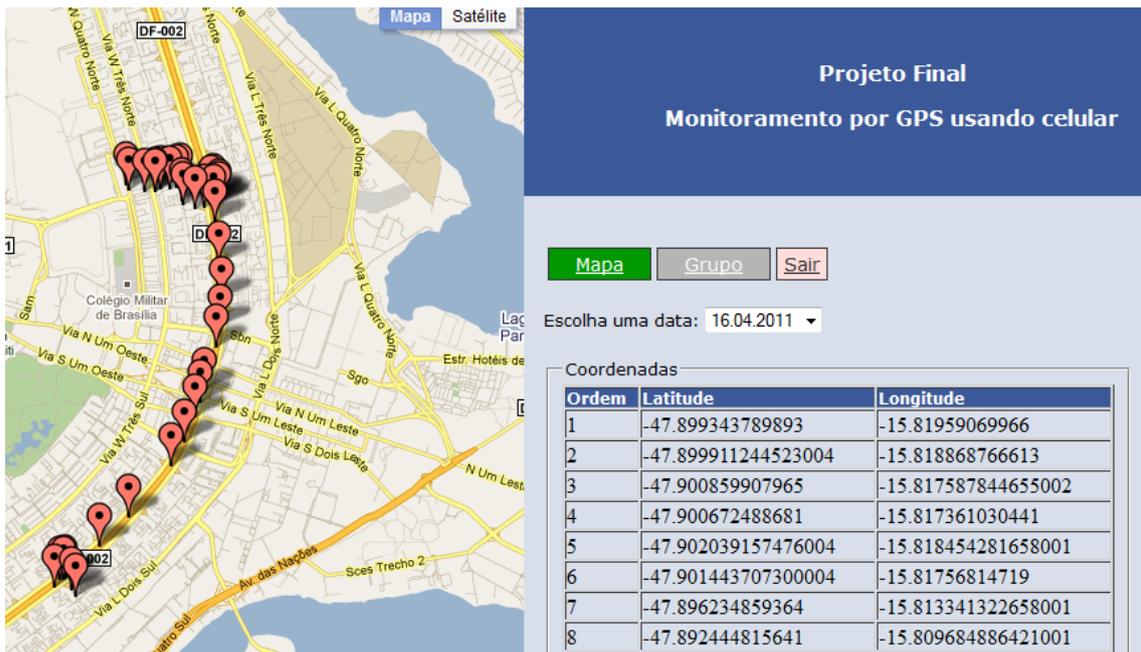


Figura 4. 12 - Trajeto 407 até o UniCeub

Desta forma se conclui o teste do projeto final. Atingindo-se o resultado esperado. Que era de capturar a trajetória do aparelho e apresentar na WEB.

## 4.2. Avaliação Global da Solução Proposta

A implementação do projeto apresentou resultados satisfatórios. Foi possível obter as coordenadas através do GPS e enviá-las para o banco de dados na WEB, disponibilizando as coordenadas através da interface WEB.

A recepção do sinal do GPS se mostrou dependente de fatores externos. Notou-se que em dias com muitas nuvens ou chuva, muitas vezes não é possível se obter uma coordenada. Em ambientes fechados ou sem vista clara para o céu também não era possível obter uma coordenada. Uma sugestão para ser utilizada quando não é possível utilizar o GPS seria utilizar-se da triangulação pelas torres ou identificação da torre no qual se está conectado. A precisão por este métodos é mais baixa, mas não é interrompida tão facilmente.

O envio e recebimento de dados não apresentou problemas, não ocorrendo perda de mensagens ou indisponibilidade para envio ou recebimento.

Apesar da precisão não ser um fator ponderante no projeto, foi comparada a obtenção de coordenadas com GPS para uso veicular modelo EASY da marca TomTom. Como visto na figura 4.13, não existiu variação entre as coordenada obtida.



Figura 4. 13 - Coordenadas N95



Figura 4.14 - Coordenadas GPS TomTom Easy

O método escolhido para calcular a distância entre as coordenadas não mostrou grandes distorções em trajetos curtos. Mas as distorções tendem a aumentar com o aumento da distância entre as coordenadas. Isso ocorre pelo fato do cálculo utilizado fazer a aproximação da Terra como uma esfera perfeita, o que não é uma distância verdadeira. Para se obter uma distância mais próxima com a real seria necessário calcular a distância topográfica. Para exemplificar, foi obtido informações de distância entre as capitais no sítio do Departamento Nacional de Infraestrutura de Transportes (DNIT) (<http://www1.dnit.gov.br/download/distancias.zip>), junto com as coordenadas das capitais no sítio do Observatório Nacional (<http://euler.on.br/ephemeris/escolhasite.php>). Na tabela 4.1 é mostrada o resultado.

Tabela 4.1 - Distância entre dois pontos sobre distância ( em relação a Brasília)

	Distância	Distância Calculada
Goiânia	209 Km	174,06 Km
Uberlândia	435 Km	353,26 Km
São Paulo	1015 Km	877,06 Km
Rio de Janeiro	1148 Km	936,61 Km

Estes resultados estão dentro do que se era esperado, que era de se mostrar a distância aproximada.

A interface WEB se mostrou bastante ágil na atualização de coordenadas nos mapas. O uso de AJAX para realizar a atualização das coordenadas trouxe um ganho de desempenho, por não ser necessário carregar as imagens do mapa novamente.

O uso dos grupos com coordenadas torna o projeto uma alternativa interessante para uso comercial. A possibilidade de saber a coordenada e a distância em que se encontra outro

aparelho possibilita o projeto ser usado para monitoramento de frotas ou pessoas. Por exemplo, poderia ser utilizado para uma frota de taxi, fazendo que o taxista que esteja mais próximo do passageiro seja acionado.

## 5. CONCLUSÃO

Este projeto teve como objetivo obter as coordenadas de um aparelho GPS e enviar para um banco de dados na web, sendo que o histórico fica disponível para consultas posteriores. Para isso foram criadas duas interfaces uma para celular feita em Java Me e outra web escrita em PHP. Através da interface do celular, foi possível capturar coordenadas através do GPS interno e enviá-las para a web. Através do aparelho celular também foi possível saber a distância entre outros usuários cadastrados. Na interface WEB foi possível observar as coordenadas enviadas pelo celular, em determinada data. Sendo possível gerenciar outros usuários, permitindo que se saiba a distância deles entre eles.

O projeto se mostrou um grande desafio, devido às diversas tecnologias empregadas. Sendo necessário um grande número de horas para pesquisas e de como implementar a solução proposta. Existindo a necessidade de recorrer a documentações em sites, devido à dificuldade de se encontrar livros sobre os assuntos abordados.

Concluindo o projeto atingiu o objetivo desejado, mostrando-se uma solução viável, podendo ser utilizada como base para outros projetos.

### 5.1. Sugestões para projetos futuros

Como sugestão para futuros projetos, poderiam ser adicionadas diversas funções e melhorias ao projeto. Entre elas:

- Aprimoramento da interface do celular. Sugere-se a melhoria da interface do celular, possibilitando visualizar um mapa, e não só a distancia entre os usuários.

- Adicionar novos métodos para obter as coordenadas do celular. Apesar do GPS ser mais preciso, nem sempre se consegue obter uma coordenada. Com por exemplo em dias de tempo instável ou em ambientes fechados. Poderiam ser usados métodos complementares ao GPS, como triangulação e identificação das torres.

- Utilizar os dados captados para informar o tráfego. Conseguindo uma amostragem maior de dados seria possível saber quantos aparelhos trafegam em determinados locais durante o dia. Isso poderia ser integrado em aparelhos de GPS existentes ou em celulares. Evitando assim caminhos, onde a probabilidade de estar congestionado e maior. Ou até um sistema que controle o tempo dos semáforos de acordo com o fluxo da região.

-Migração do aplicativo para outras plataformas. Migrar o software do celular para outras plataformas como iPhone OS<sup>2</sup> e Android<sup>3</sup>. A migração para Android em tese não traria grande modificações, pois o sistema operacional é programado em JAVA sendo recompilado pela SDK fornecida pela própria Google.

---

<sup>2</sup> iPhone OS é um sistema operacional móvel. Desenvolvido pela empresa Apple.

<sup>3</sup> Android é um sistema operacional móvel que roda sobre o núcleo Linux. Desenvolvido pela empresa Google.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Corrêa, Iran Carlos Stalliviere. 2011.** [Online] 04 12, 2011. [Citado: 04 12, 2011.] [http://www.ufrgs.br/museudetopografia/Artigos/Coordenadas\\_geogr%C3%A1ficas.pdf](http://www.ufrgs.br/museudetopografia/Artigos/Coordenadas_geogr%C3%A1ficas.pdf).
- Ferreira Daniel Avles. 2010.** SISTEMA DE ALARME INTEGRANDO GPS E CELULAR. 2010.
- Kreger, Heather. 2011.** Web Services Conceptual Architecture (WSCA 1.0). *Web Services Conceptual Architecture (WSCA 1.0)*. 2011.
- Mahmoud, Quasay.** SDN. <http://developers.sun.com/mobility/midp/articles/persist/>. [Online] Oracle. [Citado: 05 01, 2011.] <http://developers.sun.com/mobility/midp/articles/persist/>.
- MySQL Community Edition.** <http://www.mysql.com/products/community/>. [Online] Mysql. [Citado: 05 22, 2011.]
- MySQL.** MySQL Community Edition. <http://www.mysql.com/products/community/>. [Online] MySQL. [Citado: 05 22, 2011.]
- Oracle.** Java Me. *Oracle*. [Online] Oracle. [Citado: 04 08, 2011.] <http://www.oracle.com/technetwork/java/javame/overview/index.html>.
- PHP.NET.** Documentação – Prefácio. *PHP.NET*. [Online] [Citado: 04 11, 2011.] [http://www.php.net/manual/pt\\_BR/preface.php](http://www.php.net/manual/pt_BR/preface.php).
- Soares, Wallace. 2006.** *AJAX ( Asynchronous JavaScript And XML)*. s.l. : Editora Erica, 2006.
- W3S.** W3S - SOAP. [Online] [Citado: 03 15, 2011.] <http://www.w3schools.com/soap/default.asp>.
- Wikipedia.** Sistema de posicionamento global. *Sistema de posicionamento global*. [Online] Sistema de posicionamento. [Citado: 5 15, 2011.] [http://pt.wikipedia.org/wiki/Sistema\\_de\\_posicionamento\\_global](http://pt.wikipedia.org/wiki/Sistema_de_posicionamento_global).
- . SOAP. *Wikipedia*. [Online] [Citado: 04 11, 2011.] <http://pt.wikipedia.org/wiki/SOAP>.

## APÊNDICE I

Códigos escritos para a Interface do celular

### principal.java

```

/*
 * Classe responsável por iniciar o aplicativo e exibir as telas
 *
 */
package principal;

import autenticar.Autentica_Stub;

import dados.dados;
import comunicacao.grupo.Resposta;
import java.rmi.RemoteException;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import org.netbeans.microedition.lcdui.SimpleTableModel;
import dados.getLocalizacao;
import comunicacao.grupo.Grupo_PortType_Stub;
import org.netbeans.microedition.lcdui.TableItem;
import posicao.Posicao_PortType_Stub;

/**
 * @author Marcus Vinicius Melo Araujo
 */
public class principal extends MIDlet implements CommandListener, Runnable {

    private String usuario;
    private boolean autenticado = false;
    private boolean midletPaused = false;
    private Thread thread = null; // variavel que recebe o processo
    private dados data;
    private Form formPrincipal; //tela principal
    private Form formAutenticar; //tela de autenticação
    private TableItem itemTabela; //elemtno tabela
    private StringItem stringStatus; //
    private StringItem stringItem;
    private TextField textFieldSenha;
    private TextField textFieldUsuario;
    private Command exitCommand;
    private Command okCommand;
    private SimpleTableModel modeloTabela; //modelo da tabela

    public principal() {
    }

```

```

private void initialize() {
    //inicia a classe responsável em obter usuário e senha
    data = new dados();
}

public void startMIDlet() {
    switchDisplayable(null, getFormAutenticar());
}

public void resumeMIDlet() {
}
// faz a troca entre as telas

public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
    Display display = getDisplay();
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
}
// Responsável em capturar os comandos do menu do celular

public void commandAction(Command command, Displayable displayable) {

    if (displayable == formAutenticar) {
        if (command == exitCommand) {

            exitMIDlet();

        } else if (command == okCommand) {

            //inicia o web service de autenticação
            Autentica_Stub service = new Autentica_Stub();
            // autenticar_Stub service = new autenticar_Stub();

            //atualiza o texto de aviso
            stringItem.setText("Consultando usuário...");
            try {
                String msg = null;
                //chama a função autenticar do web service
                String res = service.autenticar(textFieldUsuario.getString(),
textFieldSenha.getString());
                //verifica se foi autenticado corretamente
                if (!res.equals("erro")) {

                    autenticado = true;
                    // salva dados do usuário

```

```

        data.setUsuario(textFieldUsuario.getString());
        data.setSenha(textFieldSenha.getString());
        data.setInfo();
        //salva a id do usuario
        usuario = res;
        msg = "Autenticado com sucesso!";

    } else {

        autenticado = false;
        msg = "Usuário ou senha incorretos!";

    }
    //atualiza texto
    stringItem.setText(msg);

} catch (RemoteException ex) {
    ex.printStackTrace();
}

// verifica se está autenticado se, sim redireciona para a tela de principal
autenticado();

}
} else if (displayable == formPrincipal) {
    if (command == exitCommand) {

        exitMIDlet();

    }
}

}

//form principal
public Form getFormPrincipal() {
    if (formPrincipal == null) {
        //inicia a form
        formPrincipal = new Form("Projeto Final", new Item[]{getStringStatus(),
getItemTabela()});
        formPrincipal.addCommand(getExitCommand());
        //adiciona os comandos no listener
        formPrincipal.setCommandListener(this);
        /*
        * Inicia um novo processo, responsável em obter as coordenadas, enviar e receber
dados
        */
        thread = new Thread(this);
        thread.start();
    }
}

```

```

    }
    return formPrincipal;
}

public Command getExitCommand() {
    if (exitCommand == null) {
        exitCommand = new Command("Sair", Command.EXIT, 0);
    }
    return exitCommand;
}

public StringItem getStringStatus() {
    if (stringStatus == null) {
        stringStatus = new StringItem("Status", "<nulo>", Item.BUTTON);
    }
    return stringStatus;
}

public SimpleTableModel getTableModel1() {
    if (modeloTabela == null) {

        //inicia a tabela
        modeloTabela = new SimpleTableModel(new java.lang.String[][]{
            new java.lang.String[]{"", "", ""},
            new java.lang.String[]{"", "", ""}}, new java.lang.String[]{"Usu\u00E1rio",
"Distancia", "Tempo"});

    }
    return modeloTabela;
}

//verifica se a autenticação foi bem sucedida
// redireciona para a janela
public void autenticado() {

    if (autenticado) {
        switchDisplayable(null, getFormPrincipal());
    } else {
        switchDisplayable(null, getFormAutenticar());
    }
}

//form de autenticação
public Form getFormAutenticar() {
    if (formAutenticar == null) {
        //inicia a form

```

```

        formAutenticar = new Form("Autentica\u00E7\u00E3o", new
Item[]{getTextFieldUsuario(), getTextFieldSenha(), getStringItem()});
        //adciona os comandos
        formAutenticar.addCommand(getExitCommand());
        formAutenticar.addCommand(getOkCommand());
        formAutenticar.setCommandListener(this);
    }
    return formAutenticar;
}

public TextField getTextFieldUsuario() {
    if (textFieldUsuario == null) {

        textFieldUsuario = new TextField("Usu\u00E1rio", data.getUsuario(), 32,
TextField.ANY);

    }
    return textFieldUsuario;
}

public TextField getTextFieldSenha() {
    if (textFieldSenha == null) {

        textFieldSenha = new TextField("Senha", data.getSenha(), 32, TextField.ANY);

    }
    return textFieldSenha;
}

public Command getOkCommand() {
    if (okCommand == null) {
        okCommand = new Command("Autenticar", "Autenticar", Command.OK, 0);
    }
    return okCommand;
}

public StringItem getStringItem() {
    if (stringItem == null) {
        stringItem = new StringItem("", null);
    }
    return stringItem;
}

public TableItem getItemTabela() {
    if (itemTabela == null) {
        itemTabela = new TableItem(getDisplay(), "Grupo");
        itemTabela.setLayout(ImageItem.LAYOUT_DEFAULT |
ImageItem.LAYOUT_NEWLINE_BEFORE | ImageItem.LAYOUT_NEWLINE_AFTER |
Item.LAYOUT_EXPAND | Item.LAYOUT_VEXPAND);
        itemTabela.setModel(getTableModel1());
    }
}

```

```

        itemTabela.setPreferredSize(-1, -1);
    }
    return itemTabela;
}

//Returns a instancia do display
public Display getDisplay() {
    return Display.getDisplay(this);
}

//sai do midlet
public void exitMIDlet() {
    switchDisplayable(null, null);
    destroyApp(
        true);
    thread = null;
    notifyDestroyed();
}

//inicial o aplicativo

public void startApp() {
    if (midletPaused) {
        resumeMIDlet();
    } else {
        initialize();
        data.getInfo();
        startMIDlet();
    }
    midletPaused = false;
}

public void pauseApp() {
    midletPaused = true;
}

}

public void grupo() {
    stringStatus.setText("Obtendo grupo");
    //inicia o web service de grupo
    Grupo_PortType_Stub service = new Grupo_PortType_Stub();
    try {

        // Executa o webservice de grupo
        Resposta[] ss = service.lista(usuario);
        // cria a variavel que vai receper a resposta do array
        Resposta r = new Resposta();
        //cria o array que vai armazenar a resposta do web service
        String[][] tabela2 = null;

```

```

// Inicia o tamanho do array
tabela2 = new java.lang.String[ss.length][];
for (int i = 0; i < ss.length; i++) {
    //adiciona a classe
    r = ss[i];
    tabela2[i] = new java.lang.String[]{r.getUsuario(), r.getDistancia(), r.getTempo()};
}
modeloTabela = new SimpleTableModel(tabela2, new
java.lang.String[]{"Usu\u00E1rio", "Dist\u00e2ncia", "Tempo"});
itemTabela.tableModelChanged(modeloTabela);
itemTabela.setModel(modeloTabela);

} catch (RemoteException ex) {
    ex.printStackTrace();
}
}

public void atualizar() {
    try {
        stringStatus.setText("Inicializando");

        //atualiza a tabela de dist\u00e2ncia de grupo
        grupo();
        //inicia o web service que envia posi\u00e7\u00e3o
        Posicao_PortType_Stub service = new Posicao_PortType_Stub();

        stringStatus.setText("Obtendo Localiza\u00e7\u00e3o...");
        getLocalizacao loc = new getLocalizacao();
        //inicia a leitura do GPS
        loc.run();

        stringStatus.setText("Enviando Coordenadas ...");
        //impede que coordenadas nulas sejam enviadas
        if (loc.latitudeString.length() > 4) {
            String s = service.envia(usuario, loc.latitudeString, loc.longitudeString);
            stringStatus.setText(s);

        } else {
            stringStatus.setText("Coordenadas n\u00e3o obtidas");
        }

    } catch (RemoteException ex) {
        ex.printStackTrace();
    }
}
}

```

```

public void destroyApp(boolean unconditional) {
}

public void run() {
    //executa o processo de forma indefinida
    while (true) {
        try {
            atualizar();
            //coloca o processo para dormir por 45 segundos
            thread.sleep(45000);
        } catch (InterruptedException ie) {
            ie.printStackTrace();
        }
    }
}
}
}
}

```

### **dados.java**

```

package dados;
import javax.microedition.rms.RecordEnumeration;
import javax.microedition.rms.RecordStore;
import javax.microedition.rms.RecordStoreException;
import javax.microedition.rms.RecordStoreNotOpenException;

/**
 *
 * @author marcus vinicius melo araujo
 * Classe responsável por
 * Recupera e grava as informacoes de senha e usuário
 *
 */
public class dados {

    private RecordStore gravado;
    private String dados;
    public String usuario;
    public String senha;

    public String getDados() {
        return dados;
    }

    public void setDados(String dados) {
        this.dados = dados;
    }
}

```

```

public String getSenha() {
    return senha;
}

public void setSenha(String senha) {
    this.senha = senha;
}

public String getUsuario() {
    return usuario;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

public void dados() {
}

/*
 * grava as informações de usuário e senha
 */
public void setInfo() {
    try {
        //Apaga os dados antes de executar a gravação
        deleteData();
        //Abre o recordStore
        gravado = RecordStore.openRecordStore("info", true);
        //grava o usuario
        byte[] strb = getUsuario().getBytes();
        gravado.addRecord(strb, 0, strb.length);
        //grava a senha
        byte[] strb2 = getSenha().getBytes();
        gravado.addRecord(strb2, 0, strb2.length);
        //fecha o recordStore
        fechaRecordStore();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/*
 * recupera as informações de usuário e senha
 */

public void getInfo() {

```

```

try {
    // marca o número de registros para zero
    int numrec = 0;
    try {

        //Abre o recordStore
        gravado = RecordStore.openRecordStore("info", true);

        //pega o numero de registros
        numrec = gravado.getNumRecords();

    } catch (RecordStoreException ex) {
        ex.printStackTrace();
    }
    // verifica a quantidade de registros gravados
    if (numrec > 1) {

        byte[] record;

        RecordEnumeration  enum = gravado.enumerateRecords(null, null, false);

        try {
            // verifica se existe usuário e senha salvos
            if (numrec == 2) {
                //recupera usuário
                record = gravado.getRecord(1);
                setUsuario(new String(record));
                // recupera senha
                record = gravado.getRecord(2);
                setSenha(new String(record));

            } else if (numrec == 1) {
                //recupera usuário
                record = gravado.getRecord(1);
                setUsuario(new String(record));
            }

        } catch (RecordStoreException ex) {

            ex.printStackTrace();
        }

    }
} catch (RecordStoreNotOpenException ex) {
    ex.printStackTrace();
}

```

```

        fechaRecordStore();

    }

    /*
    * Fecha o recordStore
    */
    private void fechaRecordStore() {
        try {
            gravado.closeRecordStore();
        } catch (RecordStoreException ex) {
            ex.printStackTrace();
        }
    }
    /*
    * Apaga as informações do recordStore
    */

    public void deleteData() {
        try {
            RecordStore.deleteRecordStore("info");

        } catch (RecordStoreException ex) {
            ex.printStackTrace();

        }

    }
}

```

### **getLocalizacao.java**

```

package dados;
import javax.microedition.lcdui.Font;
import javax.microedition.location.Coordinates;
import javax.microedition.location.Criteria;
import javax.microedition.location.Location;
import javax.microedition.location.LocationException;
import javax.microedition.location.LocationProvider;

/**
* Class responsável em acessar o GPS e obter as coordenadas
*/
public class getLocalizacao {

    public String latitudeString = "";
    public String longitudeString = "";

```

```
private void obterLocalizacao() {
    try {

        Criteria criteria = new Criteria();

        LocationProvider lp = LocationProvider.getInstance(criteria);
        if (lp != null) {
            //
            Location l = lp.getLocation(300);
            if (l.isValid()) {
                Coordinates c = l.getQualifiedCoordinates();
                if (c != null) {
                    double latitude = c.getLatitude();
                    double longitude = c.getLongitude();
                    latitudeString = "" + latitude;
                    longitudeString = "" + longitude;
                }
            }
        }
    } catch (LocationException le) {
        System.out.println(le.getMessage());
    } catch (InterruptedException ie) {
        System.out.println(ie.getMessage());
    }
}

public void run() {
    obterLocalizacao();
}
}
```

## APÊNDICE II

### Interface WEB

#### index.php

```

<?php
/*
 * Importa autenticacao.class.php
 */
include_once("include/autenticacao.class.php");
$autenticar = (boolean) $_POST['autenticar'];
//se verdadeiro efetua a autenticação
if ($autenticar) {

    $autenticacao = new autenticacao($_POST['usuario'], $_POST['password']);
    $autenticado = $autenticacao->verificaUsuario();
    // consulta ao banco
    if ($autenticacao->verificaUsuario()) {
        $autenticado = true;

        $id = $autenticacao->verificaUsuario();
    } else {
        $autenticado = false;
        $msg = "Verifique se o usu&aaacute;rio e a senha digitados est&atilde;o corretos.";
    }
}

//verifica se o usuario esta autenticado
if ($autenticado) {
    //redireciona para a pagina do mapa em caso de positivo

    header("Location: http://e-making.com/projetoweb/mapa.php?id=$id");

    //se nao estiver exhibe a autenticação
} else {
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<link rel="stylesheet" href="projeto.css" type="text/css" media="screen">

<title>Projeto Final</title>

<script language="javascript" src="js/ajax.js"></script>
<script type="text/javascript" >

function novo(){
    var usuario = document.getElementById('usuario').value;
    var senha = document.getElementById('senha').value;

```

```

        if (usuario.length < 1 && senha.length < 1) {
            alert("Os campos usuário e senha são obrigatórios!");
        } else {

ajax("include/ajaxAutenticar.php?acao=novo&usuario="+usuario+"&senha="+senha,"resposta");
        } //fim else
    } //fim function novo
</script>

</head>

<body marginheight="0" marginwidth="0">
    <div id="corpo" style="background:#D8DFEA; width:100%;">

        <table width="100%" align="center" bgcolor="#3B5998" border="0">
            <tr ><td><p align="center" class="textoTitulo">&nbsp;</p>
                <p align="center" class="textoTitulo">Projeto Final </p>
                <p align="center" class="textoTitulo">Monitoramento por GPS usando celular</p>
                <p align="center" class="textoTitulo">&nbsp;</p></td>
            </tr>
        </table>

        <br />
        <br />
        <br />
        <br />

        <table width="420" border="0" align="center" bgcolor="#D8DFEA" >

            <tr><td>

                <div id="form" style="width: 450px;">
                    <fieldset><legend class="texto">Autenticar</legend>
                    <form action="index.php" method="post" enctype="multipart/form-data"
name="formautentica" target="_self">
                        <input name="autenticar" id="autenticar" value="true" type="hidden" />
                        <table width="100%" border="0" align="center">
                            <tr>
                                <td width="25%">&nbsp;</td>
                                <td width="12%">&nbsp;</td>
                                <td width="29%">&nbsp;</td>
                                <td width="34%">&nbsp;</td>
                            </tr>
                            <tr>
                                <td>&nbsp;</td>
                                <td><span class="texto">Usu&aacute;rio</span></td>
                                <td><label>
                                    <input name="usuario" type="text" id="usuario" />
                                </label></td>
                                <td>&nbsp;</td>
                            </tr>
                            <tr>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                            </tr>
                        </table>
                    </form>
                </div>
            </td>
        </tr>
    </table>

```

```

        <td class="texto">Senha</td>
        <td><label>
            <input name="senha" type="password" id="senha"/>
        </label></td>
        <td><input name="Submit" type="submit" class="botao"
value="Entrar" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td><label></label></td>
    </tr>
    <tr>
        <td width="25%">&nbsp;</td>
        <td width="12%">&nbsp;</td>
        <td width="29%">&nbsp;</td>
        <td width="34%">&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td class="texto">Criar <strong><a href="javascript:novo();"
>novo</a></strong> usu&aacute;rio. </td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td class="texto">&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>

    </form>
</fieldset></div>
<table width="100%" align="center">
    <tr>
        <td align="center"><span class="texto" > <div id="resposta"
name="resposta"><?=$msg; ?>&nbsp;</div></span></td>
    </tr>
</table>

</td></tr></table>
<br />
<br />
<br />
<br />

</div>

```

```

<table width="100%" align="center" bgcolor="#3B5998" border="0" class="rodape">
  <tr ><td><p align="center" class="textoTitulo">Marcus Vinicius Melo Araujo </p>
    </td>
  </tr>
</table>
</body>
</html>
<?php
} //fim else
?>

```

### mapa.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <title>Projeto Final - Mapa</title>
    <link
href="http://code.google.com/apis/maps/documentation/javascript/examples/default.css"
rel="stylesheet" type="text/css" />
    <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
    <script type="text/javascript" src="js/mapa.js"></script>
    <script type="text/javascript" src="js/posicao.js" ></script>
  </head>
  <body onLoad="initialize();" marginheight="0" marginwidth="0" leftmargin="0"
rightmargin="0" topmargin="0" bgcolor="#D8DFEA">
    <link rel="stylesheet" href="projeto.css" type="text/css" media="screen">
    <input type="hidden" id="grupoId" name="grupoId" value="<?php echo
$_REQUEST['id']; ?>">
    <table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#D8DFEA">
      <tr>
        <td>
          <p align="center">&nbsp;</p>
          <p align="center">&nbsp;</p></td>
          <td bgcolor="3B5998">&nbsp;</td>
          <td colspan="3" bgcolor="3B5998">
            <p align="center" class="textoTitulo">&nbsp;</p>
            <p align="center" class="textoTitulo">Projeto Final </p>
            <p align="center" class="textoTitulo">Monitoramento por GPS usando
celular</p>
          <p align="center" class="textoTitulo">&nbsp;</p>
        </td>
      </tr>
    </table>

```

```

</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td colspan="2">&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td colspan="2">&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td width="0%">&nbsp;</td>
  <td width="61%">&nbsp;</td>
  <td colspan="2">
    <table width="200" border="0">
      <tr>
        <td>
          <a href="mapa.php?id=<?= $_REQUEST['id']; ?>"
class="botaoMapaA">Mapa</a></td>
        <td><a href="grupo.php?id=<?= $_REQUEST['id']; ?>"
class="botaoGrupoI">Grupo</a></td>
        <td><a href="index.php" class="botaoSair">Sair</a></td>
      </tr>
    </table>

    </td>
  <td width="6%">&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td width="3%">&nbsp;</td>
  <td width="30%">&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>
    <div id="map_canvas" style="width: 60%;"></div>
  </td>
  <td colspan="2"><span class="texto">Escolha uma data:</span>
    <select name="data" id="data" onChange="loop();">
      <?php
        require_once("include/conexao.class.php");
        //inicia a conexão com o banco
        conexao :: conectar();
        //seleciona as datas

```

```

$resultado = mysql_query("
SELECT DISTINCT DATE( `dt_recebido` ) as us,
DATE_FORMAT(DATE( `dt_recebido` ),
GET_FORMAT(DATE,'EUR')) as dt_recebido
FROM `tb_posicao` WHERE fk_usuario = " . $_REQUEST['id'] . "");

```

```

while ($dados = mysql_fetch_array($resultado)) {

    $latitude = $dados['nr_latitude'];
    $longitude = $dados['nr_longitude'];
    $us = $dados['us'];
    $dt_recebido = $dados['dt_recebido'];
    //popula a lista
    echo("<option value='$us'$>$dt_recebido</option>");
} ?>
</select>

```

```

</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td colspan="2">
<br>
<fieldset><legend class="texto">Coordenadas</legend>
<div id="resposta" >&nbsp;</div></fieldset>
</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td colspan="2">&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td colspan="2"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td colspan="2">&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td colspan="2">&nbsp;</td>
<td>&nbsp;</td>
</tr>

```

```

        <td>&nbsp;</td>
        <td bgcolor="#3B5998">&nbsp;</td>
        <td colspan="3">
            <table width="100%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#3B5998" class="rodape">
                <tr><td><p align="center" class="textoTitulo">Marcus Vinicius Melo
Araujo </p>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</table>

</body>

</html>

```

### grupo.php

```

<?php
$susu = $_REQUEST["id"];
?>
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
        <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
        <title>Projeto Final</title>
        <link
href="http://code.google.com/apis/maps/documentation/javascript/examples/default.css"
rel="stylesheet" type="text/css" />
        <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
        <script type="text/javascript" src="js/mapa.js"></script>
        <script type="text/javascript" src="js/ajax.js"></script>
        <script type="text/javascript" src="js/grupo.js"></script>
    </head>

    <body onLoad="initialize();loop();" marginheight="0" marginwidth="0" leftmargin="0"
rightmargin="0" topmargin="0" bgcolor="#D8DFEA">
        <link rel="stylesheet" href="projeto.css" type="text/css" media="screen">
        <input type="hidden" id="grupoId" name="grupoId" value="<?php echo
$_REQUEST['id']; ?>">
        <table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#D8DFEA">
            <tr>
                <td>

```

```

        <p align="center">&nbsp;</p>
        <p align="center">&nbsp;</p></td>
<td bgcolor="3B5998">&nbsp;</td>
<td colspan="2" bgcolor="3B5998">

        <p align="center" class="textoTitulo">&nbsp;</p>
        <p align="center" class="textoTitulo">Projeto Final </p>
        <p align="center" class="textoTitulo">Monitoramento por GPS usando celular</p>
        <p align="center" class="textoTitulo">&nbsp;</p>        </td>
</tr>
<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td width="0%">&nbsp;</td>
        <td width="61%">&nbsp;</td>
        <td>
                <table width="200" border="0">
                        <tr>
                                <td> <a href="mapa.php?id=<?=$_REQUEST['id']; ?>"
class="botaoMapaI">Mapa</a></td>
                                <td><a href="grupo.php?id=<?=$_REQUEST['id']; ?>"
class="botaoGrupoA">Grupo</a></td>
                                <td><a href="index.php" class="botaoSair">Sair</a></td>
                        </tr>
                </table>
        </td>
        <td width="6%">&nbsp;</td>
</tr>
<tr>
        <td>&nbsp;</td>
        <td> <div id="map_canvas" style="width: 61%;"></div> </td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td class="texto">
                <br>
                <br>Usuário: <div id="usuarioPrincipal">&nbsp;</div>
                <fieldset><legend class="texto">Grupo</legend>

```

```

        <div id="grupo">&nbsp;</div>
        </fieldset></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td><span class="texto">Convidar</span>
    </tr>

    <input type="text" name="usuario" id="usuario"><button onClick="adicionar();"
class="botao"> Adicionar</button>
    <br>
    <div id="resposta">&nbsp;</div>
    </td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>
        <fieldset><legend class="texto">Pendentes</legend>
        <div id="pendentes">&nbsp;</div>
        </fieldset>
    </td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td bgcolor="#3B5998">&nbsp;</td>
    <td colspan="2"><table width="100%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#3B5998" class="rodape">

```

```

        <tr ><td><p align="center" class="textoTitulo">Marcus Vinicius Melo Araujo
</p>
                </td>
        </tr>
</table>    </td>
</tr>
</table>
</body>
</html>

```

## WEB SERVICE

### autenticar.php

#### <?php

```

/*
 * Web Service Responsável por processar as informações de autenticação
 */
//desabilita a cache do web service
ini_set("soap.wsdl_cache_enabled", 0);
include_once("../include/autenticacao.class.php");

class entrada {

    public $usuario;
    public $senha;

}

class resposta {

    public $msgResposta;

}

interface autentica {

    public function autenticar($parametros);
}

class webAutentica implements autentica {

    public function autenticar($parametros) {
        $parametros;
        $saida = new resposta();
        $saida->msgResposta = $parametros->usuario;
        $saida->msgResposta = $parametros->senha;
        $autenticar = new autenticacao($parametros->usuario, $parametros->senha);
    }
}

```

```

//verifica o usuário
if ($autenticar->verificaCredenciais()) {
    $saida->msgResposta = $autenticar->verificaUsuario();
} else {
    $saida->msgResposta = "erro";
}
return $saida;
}
}

```

```

//inicia o web service
$server = new SoapServer("autenticar.wsdl");
$server->setObject(new webAutentica());
$server->handle();
?>

```

### **autenticar.wsdl**

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.e-making.com/projetoweb/ws/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="autenticar"
targetNamespace="http://www.e-making.com/projetoweb/ws/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.e-making.com/projetoweb/ws/">
      <xsd:element name="autenticar" type="tns:entrada">

      </xsd:element>
      <xsd:element name="autenticarResponse" type="tns:resposta">

      </xsd:element>
      <xsd:element name="novo" type="tns:entrada">

      </xsd:element>
      <xsd:element name="novoResponse" type="tns:resposta">

      </xsd:element>

      <xsd:complexType name="resposta">
        <xsd:sequence>
          <xsd:element name="msgResposta" type="xsd:string"></xsd:element>
        </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="entrada">
        <xsd:sequence>
          <xsd:element name="usuario" type="xsd:string"></xsd:element>
          <xsd:element name="senha" type="xsd:string"></xsd:element>

```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="autenticarRequest">
    <wsdl:part element="tns:autenticar" name="parameters"/>
</wsdl:message>
<wsdl:message name="autenticarResponse">
    <wsdl:part element="tns:autenticarResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="novoRequest">
    <wsdl:part name="parameters" element="tns:novo"></wsdl:part>
</wsdl:message>
<wsdl:message name="novoResponse">
    <wsdl:part name="parameters" element="tns:novoResponse"></wsdl:part>
</wsdl:message>
<wsdl:portType name="autentica">
    <wsdl:operation name="autenticar">
        <wsdl:input message="tns:autenticarRequest"/>
        <wsdl:output message="tns:autenticarResponse"/>
    </wsdl:operation>

</wsdl:portType>
<wsdl:binding name="autenticarSOAP" type="tns:autentica">

    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="autenticar">

        <soap:operation
            soapAction="http://www.e-making.com/projetoweb/ws/autenticar" />
        <wsdl:input>

            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>

            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>

</wsdl:binding>
<wsdl:service name="autenticar">
    <wsdl:port binding="tns:autenticarSOAP" name="autenticarSOAP">
        <soap:address location="http://www.e-making.com/projetoweb/ws/autenticar.php"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

**grupo.php**  
**<?php**

```
//desabilita a cache do web service
ini_set("soap.wsdl_cache_enabled", 0);

//importa as classes
include_once("../include/grupo.class.php");
include_once("../include/xml.class.php");
include_once("../include/coordenadas.class.php");

class entrada {

    public $usuario;

}

class resposta {

    public $usuario;
    public $distancia;
    public $tempo;

}

interface grupo {

    public function lista($param);
}

class webGrupo implements grupo {

    //inicial o array com a resposta
    public $linha = array();

    public function lista($param) {
        //inicia a classe de resposta
        // $p = new resposta();

        $lista = new ClassGrupo();
        $lista->id_usuario = $param->usuario;

        //pega a ultima posição do usuário de referencia
        $coordenadasIni = ( $lista->posicao());
        $latitudeIni = $coordenadasIni['nr_latitude'];
        $longitudeIni = $coordenadasIni['nr_longitude'];
        $dataIni = $coordenadasIni['dt_recebido'];

        //inicia classe para calculo da distância
```

```

$ins = new Coordenadas();

//inserir dados de posição inicial
$p1lat = $latitudeIni;
$p1long = $longitudeIni;

//inicia o array
$data = array();

//carrega variavel de resultados
$rs = $lista->listar();

$i = 0;
//executa
while ($dados = mysql_fetch_assoc($rs)) {
    $p2lat = $dados['nr_latitude'];
    $p2long = $dados['nr_longitude'];
    //calcula de ab
    $ab = $ins->calculaArco($p2lat);
    //Calcula bc
    $ac = $ins->calculaArco($p1lat);
    //Calcula A
    $A = $ins->diferencaLong($p1long, $p2long);
    $ins->esfera($A, $ac, $ab);
    //calcula a distância
    $ins->calcDistanciaKm($ins->esfera($A, $ac, $ab));
    $p = new resposta();
    $p->usuario = $data[$i]['usuario'] = $dados['nm_usuario'];
    //efetua o calculo de distancia e carrega na variavel
    $p->distancia = $data[$i]['distancia'] = number_format($ins->calcDistanciaKm($ins-
>esfera($A, $ac, $ab)),3,',','.')." Km";
    $p->tempo = $lista->tempo($dataIni, $dados['dt_recebido']);
    $i++;
    //carrega o array de resposta
    $this->linha[] = $p;
}
return $this->linha;
}

}

//Inicia o Web Service
$server = new SoapServer("grupo.wsdl");
$server->setObject(new webGrupo());
$server->handle();
?>

```

### **grupo.wsdl**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.e-making.com/projetoweb/ws/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="grupo"
targetNamespace="http://www.e-making.com/projetoweb/ws/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.e-making.com/projetoweb/ws/">
      <xsd:element name="lista" type="tns:entrada">

        </xsd:element>
        <xsd:element name="listaResponse">

          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="saida"
                type="tns:resposta" minOccurs="1" maxOccurs="unbounded">
                </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>

        <xsd:complexType name="entrada">
          <xsd:sequence>

            <xsd:element name="usuario" type="xsd:string"></xsd:element>
          </xsd:sequence>
        </xsd:complexType>

        <xsd:complexType name="resposta">
          <xsd:sequence>
            <xsd:element name="usuario" type="xsd:string"></xsd:element>
            <xsd:element name="distancia" type="xsd:string"></xsd:element>
            <xsd:element name="tempo" type="xsd:string"></xsd:element>
          </xsd:sequence>
        </xsd:complexType>

      </xsd:schema>
    </wsdl:types>
    <wsdl:message name="listaRequest">
      <wsdl:part element="tns:lista" name="parameters"/>
    </wsdl:message>
    <wsdl:message name="listaResponse">
      <wsdl:part element="tns:listaResponse" name="parameters"/>
    </wsdl:message>
    <wsdl:portType name="grupo">
      <wsdl:operation name="lista">
        <wsdl:input message="tns:listaRequest"/>
        <wsdl:output message="tns:listaResponse"/>
      </wsdl:operation>
    </wsdl:portType>
  </wsdl:definitions>

```

```

<wsdl:binding name="grupoSOAP" type="tns:grupo">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="lista">
    <soap:operation soapAction="http://www.e-making.com/projetoweb/ws/lista"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="grupo">
  <wsdl:port binding="tns:grupoSOAP" name="grupoSOAP">
    <soap:address location="http://www.e-making.com/projetoweb/ws/grupo.php"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

### **posicao.php**

#### **<?php**

```

/*
 * Web Service Responsável por processar as informações de Grupo
 */

```

```

ini_set("soap.wsdl_cache_enabled", 0);
require_once("../include/posicao.class.php");

```

```

class entrada {

```

```

    public $usuario;
    public $latitude;
    public $longitude;

```

```

}

```

```

class resposta {

```

```

    public $msgResposta;

```

```

}

```

```

interface intposicao {

```

```

    public function envia($parametros);

```

```

}

```

```

class webPosicao implements intposicao {

```

```

public function envia($parametros) {

    $saida = new resposta();

    $ins = new posicao();
    $ins->id_usuario = $parametros->usuario;
    $ins->nr_latitude = $parametros->latitude;
    $ins->nr_longitude = $parametros->longitude;

    //inserir a coordenada e retorna a resposta
    $saida->msgResposta = $ins->inserir();
    return $saida;
}

}

//Inicia o Web Service
$server = new SoapServer("posicao.wsdl");
$server->setObject(new webPosicao());
$server->handle();
?>

```

### **posicao.wsdl**

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.e-making.com/projetoweb/ws/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="posicao"
targetNamespace="http://www.e-making.com/projetoweb/ws/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.e-making.com/projetoweb/ws/">
      <xsd:element name="envia" type="tns:entrada">

        </xsd:element>
        <xsd:element name="enviaResponse" type="tns:resposta">

          </xsd:element>

        <xsd:complexType name="entrada">
          <xsd:sequence>
            <xsd:element name="usuario" type="xsd:string"></xsd:element>
            <xsd:element name="latitude" type="xsd:string"></xsd:element>
            <xsd:element name="longitude" type="xsd:string"></xsd:element>

          </xsd:sequence>
        </xsd:complexType>

        <xsd:complexType name="resposta">

```

```

        <xsd:sequence>
            <xsd:element name="msgResposta" type="xsd:string"/></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="enviaRequest">
    <wsdl:part element="tns:envia" name="parameters"/>
</wsdl:message>
<wsdl:message name="enviaResponse">
    <wsdl:part element="tns:enviaResponse" name="parameters"/>
</wsdl:message>
<wsdl:portType name="posicao">
    <wsdl:operation name="envia">
        <wsdl:input message="tns:enviaRequest"/>
        <wsdl:output message="tns:enviaResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="posicaoSOAP" type="tns:posicao">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="envia">
        <soap:operation soapAction="http://www.e-making.com/projetoweb/ws/envia"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="posicao">
    <wsdl:port binding="tns:posicaoSOAP" name="posicaoSOAP">
        <soap:address location="http://www.e-making.com/projetoweb/ws/posicao.php"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

## ajaxAutenticar.php

### <?php

```

/*
 * Responsavel em receber as chamadas em AJAX de autenticação
 *
 */
require_once("autenticacao.class.php");
require_once("xml.class.php");

$acao = $_REQUEST['acao'];
$usuario = $_REQUEST['usuario'];

```

```

$senha = $_REQUEST['senha'];

switch ($acao) {

    //novo cadastro
    case 'novo':

        $lista = new autenticacao();
        $lista->usuario = $usuario;
        $lista->senha = $senha;
        // Tenta efetuar o cadastro do usuário
        $rs = $lista->cadastrar();

        //verifica se o resultado retornou o número de identificação do usuário
        if (is_numeric($rs)) {
            echo ("Usuário cadastrado com sucesso");
        } else {
            echo ("Usuário já existe");
        }

        break;
    default:
        break;
}
?>

```

### **ajaxGrupo.php**

```
<?php
```

```

/*
 * Responsável em receber as chamadas em AJAX da página de Grupo
 *
 */
require_once("grupo.class.php");
require_once("xml.class.php");
require_once("coordenadas.class.php");

$acao = $_REQUEST['acao'];
$usuario = $_REQUEST['usuario'];
$grupoId = $_REQUEST['grupoId'];
$usuario2 = $_REQUEST['usuario2'];

switch ($acao) {
    //cadastra usuário no grupo
    case 'cadastrar':

```

```

$cadastrar = new ClassGrupo();
$cadastrar->id_usuario = $grupoId;
$cadastrar->usuario = $cadastrar->getID($usuario);
// verifica se retornou um usuário valido
if (is_numeric($cadastrar->usuario)) {

    echo $cadastrar->cadastrar();
} else {
    echo ("Usuário não existe!");
}
break;
// lista os usuarios do grupo e a distancia
case 'listar':
    //obtem a ultima posicao do usuario
    $lista = new ClassGrupo();
    $lista->id_usuario = $grupoId;
    $coordenadasIni = ( $lista->posicao());
    $latitudeIni = $coordenadasIni['nr_latitude'];
    $longitudeIni = $coordenadasIni['nr_longitude'];
    $dataIni = $coordenadasIni['dt_recebido'];

    $ins = new Coordenadas();

    $p1lat = $latitudeIni;
    $p1long = $longitudeIni;

    $data = array();
    $rs = $lista->listar();
    //construir o array de distancias
    $i = 0;
    while ($dados = mysql_fetch_assoc($rs)) {

        $data[$i]['latitude'] = $p2lat = $dados['nr_latitude'];
        $data[$i]['longitude'] = $p2long = $dados['nr_longitude'];
        //calcula de ab
        $ab = $ins->calculaArco($p2lat);
        //calcula de ac
        $ac = $ins->calculaArco($p1lat);
        //calcula de A
        $A = $ins->diferencaLong($p1long, $p2long);
        $data[$i]['usuario'] = $dados['nm_usuario'];
        //calcula a distância
        $data[$i]['distancia'] = number_format($ins->calcDistanciaKm($ins->esfera($A, $ac,
        $ab)),3,',','.') . " Km";
        //calcula o tempo entre as medições
        $data[$i]['tempo'] = $lista->tempo($dataIni, $dados['dt_recebido']);
        $i++;
    }
    //gera uma resposta XML
    xml::getXML_Array($data);

```

```

    break;
//lista as coordenadas no mapa
case 'listarMapa':

    $lista = new ClassGrupo();
    $lista->id_usuario = $grupoId;

    //gera uma resposta XML
    xml::getXML($lista->listar());

    break;
//lista de usuários pendentes de aprovação
case 'listarPendentes':
    //inicia a classe
    $lista = new ClassGrupo();
    $lista->id_usuario = $grupoId;

    //gera uma resposta XML
    xml::getXML($lista->listarPendentes());

    break;
//nega o convite removendo da tabela
case 'negar':

    $lista = new ClassGrupo();
    $lista->id_usuario = $grupoId;
    $lista->id_usuario2 = $usuario2;
    //remove o registro da tabela do primeiro usuário
    $lista->remover($param);

    $lista->id_usuario = $usuario2;
    $lista->id_usuario2 = $grupoId;
    //remove o registro da tabela do segundo usuário
    $lista->remover($param);
    echo ("Usuário removido");
    break;
//aprova o convite
case 'aprovar':
    $lista = new ClassGrupo();

    $lista->id_usuario = $grupoId;
    $lista->id_usuario2 = $usuario2;
    // aprova para primeiro usuário
    $lista->aprovar($param);

    $lista->id_usuario = $usuario2;
    $lista->id_usuario2 = $grupoId;
    // aprova para o segundo usuário

```

```

        $lista->aprovar($param);
        break;
//retorna a ultima coordenada do usuario
case 'main':

        $lista = new ClassGrupo();
        $lista->id_usuario = $grupoId;

        //gera uma resposta XML
        xml::getXML($lista->main());

        break;
default:
        break;
}
?>

```

### **ajaxMapa.php**

```
<?php
```

```

/*
 * Responsavel em receber as chamadas em AJAX da atualizaçao de coordenadas na página
 de mapa
 *
 */
require_once("posicao.class.php");
require_once("xml.class.php");

$acao = $_REQUEST['acao'];
$usuario = $_REQUEST['usuario'];
$grupoId = $_REQUEST['grupoId'];
$data = $_REQUEST['data'];

switch ($acao) {
    case 'listar':
        $lista = new posicao();
        $lista->id_usuario = $grupoId;
        $lista->data = $data;
        //gera o XML
        xml::getXML($lista->listar());

        break;

    default:
        break;
}
?>

```

**autenticacao.class.php****<?php**

```

/*
 * Classe de autenticação
 *
 */
require_once("conexao.class.php");
conexao :: conectar();

class Autenticacao {

    public $usuario;
    public $senha;
    public $resultado;

    //construtor
    public function autenticacao($usuario, $senha) {
        $this->usuario = $usuario;
        $this->senha = $senha;
    }

    //cadastra usuário
    public function cadastrar() {
        //se o usuário não existir faz a inserção
        if ($this->verificaUsuario() == false) {

            $this->resultado = mysql_query("
            INSERT INTO `tb_usuario` (
            `id_usuario` ,
            `nm_usuario` ,
            `nm_senha`
            )
            VALUES (
            NULL ,
            '" . $this->usuario . "',
            '" . $this->senha . "'
            )");
            //verifica se o usuário foi inserido corretamente retornando o id_usuario gerado
            if ($this->resultado) {
                return mysql_insert_id();
            } else {
                return false;
            }
        } else {
            return "Usuário já existe!! Para acessar você deve fazer sua autenticação!";
        }
    }
}

```

```

//verifica se o usuário existe
public function verificaUsuario() {
    $this->resultado = mysql_query("SELECT id_usuario AS rs FROM tb_usuario WHERE
nm_usuario = " . $this->usuario . " limit 1");
    $dados = mysql_fetch_array($this->resultado);
    if ($dados["rs"] != null) {
        return $dados["rs"];
    } else {
        return false;
    }
}

//verifica senha e usuario
public function verificaCredenciais() {

    $this->resultado = mysql_query("SELECT id_usuario AS rs FROM tb_usuario WHERE
nm_senha = " . $this->senha . " AND nm_usuario = " . $this->usuario . " limit 1");
    $dados = mysql_fetch_array($this->resultado);

    if ($dados["rs"] != null) {
        return $dados["rs"];
    } else {
        return false;
    }
}

}

?>

```

### **conexao.class.php**

```
<?php
```

```

/*
 * Classe responsável em realizar a conexão com o banco
 */

class Conexao {

    public $banco = ""; // banco de dados a ser acessado
    public $root = ""; // usuário do banco de dados
    public $senha = ""; // senha do banco de dados
    public $local = ""; //endereço do banco de dados
    public $porta = '3306'; //porta usada pelo banco
    public static $link; // Guarda conexão com o banco
    static protected $instance; // Guarda uma instância da classe

    public function __construct() { // Um construtor privado

```

```

        self::ConectaBanco();
        self::SelecionaBanco();
    }

    //conecta com o banco
    public function ConectaBanco() {
        $this->link = mysql_connect($this->local . ':' . $this->porta, $this->root, $this->senha);
        if (!$this->link) {
            die('Não foi possível conectar: ' . mysql_error() . "<br><br>");
        }
    }

//Fim ConectaBanco

    public function SelecionaBanco() {
        $db_selected = mysql_select_db($this->banco, $this->link);
        if (!$db_selected) {
            die('Erro de conexão com o banco : ' . mysql_error());
        }
    }
// Gera a instancia da classe de conexão
    public static function conectar() {
        try {

            if (!isset(self::$instance)) { //verifica se ja foi instaciada a classe
                $c = __CLASS__; //pega o nome da classe que fez a chamada
                self::$instance = new $c (); //cria objeto
            }
        } catch (Exception $exc) {
            echo $exc->getTraceAsString();
        }
        return self::$instance;
    }
//fecha a conexao com o banco de dados
    public static function fecharConexao() {
        try {
            mysql_close(self::$link);
        } catch (Exception $exc) {
            echo $exc->getTraceAsString();
        }
    }
}

?>

coordenadas.class.php

<?php

```

```
// Constante
define("ARCO", "40076"); // circunferencia da terra 2.PI.raio da terra
/*
 * Classe responsável por manipular as coordenadas
 */

class Coordenadas {

    public $DMS;
    public $DD;
    public $grau;
    public $Graus;
    public $minutos;
    public $segundos;
    public $sinal;

    public function getDMS() {
        return $this->DMS;
    }

    public function setDMS($DMS) {
        $this->DMS = $DMS;
    }

    public function getDD() {
        return $this->DD;
    }

    public function setDD($DD) {
        $this->DD = $DD;
    }

    public function getGrau() {
        return $this->grau;
    }

    public function setGrau($grau) {
        $this->grau = $grau;
    }

    public function getMinutos() {
        return $this->minutos;
    }

    public function setMinutos($minutos) {
        $this->minutos = $minutos;
    }

    public function getSegundos() {
        return $this->segundos;
    }
}
```

```
}

public function setSegundos($segundos) {
    $this->segundos = $segundos;
}

public function getSinal() {
    return $this->sinal;
}

public function setSinal($sinal) {
    $this->sinal = $sinal;
}

public function coordenadas() {
    $this->DD = $coordenadas;
}

public function formataNumer($numero) {
    return number_format($numero, 7);
}

public function calculaGraus($hora, $minutos, $segundos) {
    $graus = 0;
    $graus = (($hora * 60 * 60) + $minutos * 60 + ($segundos)) / 3600;
    return $graus;
}

public function calculaArco($grau) {
    return $this->formataNumer($grau = 90 - ($grau));
}

public function getInteiro($valor) {
    return substr($valor, '.');
}

public function getCasa($valor) {
    return strpos($valor, '.');
}

public function valor($valor) {
    return (int) "0" . stristr($valor, '.');
}

public function multiplicar($valor) {

    return $valor * 60;
}

public function calcDistanciaKm($grau) {
```

```

        return number_format(( ($grau * ARCO) / 360), 3);
    }

    public function diferencaLong($long1, $long2) {
        return $long1 - ( $long2);
    }

    public function calcCos($param) {
        return(( cos(deg2rad($param))));
    }

    public function calcSeno($param) {
        return( ( sin(deg2rad($param))));
    }

    public function calcArcCos($param) {
        return(( acos(($param))));
    }

    public function getGraus() {
        return $this->Graus;
    }

    public function setGraus($Graus) {
        $this->Graus = $Graus;
    }

    public function esfera($a, $b, $c) {
        //usa a formula da circunferencia
        $cos = ( ($this->calcCos($b) * $this->calcCos($c) ) + ( $this->calcSeno($b) * $this->calcSeno($c) * $this->calcCos($a)));

        return $this->formataNumer(rad2deg($this->calcArcCos($cos)));
    }
}

?>

```

### **grupo.class.php**

```
<?php
```

```

/*
 * Classe responsável em realizar operacoes com grupo
 */
require_once("conexao.class.php");
//inicia a conexão com o banco
conexao :: conectar();

```

```

class ClassGrupo {

    public $id_usuario;
    public $id_usuario2;
    public $usuario;
    public $posicao1;
    public $posicao2;

    public function cadastrar() {

        if ($this->verificar() == false) {
            //cria a primeira entrada
            $this->inserir($this->id_usuario, $this->usuario, 'N');
            //cria a primeira entrada
            $this->inserir($this->usuario, $this->id_usuario, 'A');
            return ("Convite solicitado!");
        } else {
            return "Convite já cadastrado";
        }
    }

    //insere na tabela de grupo
    function inserir($usuario_grupo, $usuario, $flag) {

        $this->resultado = @ mysql_query(
            "INSERT INTO `tb_grupo` ( `fk_usuario_grupo` , `fk_usuario` ,
            `fl_authorized` ) VALUES ( '$usuario_grupo', '$usuario ', '$flag' )"
        );
        if ($this->resultado) {
            return mysql_insert_id();
        } else {
            return "false";
        }
    }

    //lista os usuarios do grupo com a ultima coordenada recebida
    public function listar() {
        $this->resultado = mysql_query("
        SELECT distinct
        gr.fk_usuario_grupo,usu.id_usuario,usu.nm_usuario,nr_latitude,nr_longitude,dt_recebido,DATE_FORMAT(( `dt_recebido` ), '%e/%m/%Y %H:%i:%s') as dt_formatada
        FROM tb_grupo gr
        INNER JOIN tb_usuario usu on usu.id_usuario = gr.fk_usuario
        INNER JOIN (SELECT * FROM tb_posicao where id_posicao in (SELECT
        max(id_posicao) as id_posicao FROM tb_posicao group by fk_usuario )) po on po.fk_usuario
        = usu.id_usuario
        WHERE gr.fk_usuario_grupo = '$this->id_usuario' and gr.fl_authorized = 'S' ");
        return $this->resultado;
    }
}

```

```

//lista os pendentes
public function listarPendentes() {
    $this->resultado = mysql_query("
    SELECT *
    FROM tb_grupo grupo
    INNER JOIN tb_usuario usuario
    ON usuario.id_usuario = grupo.fk_usuario
    WHERE fl_autorizado <> 'S' AND fk_usuario_grupo = '$this->id_usuario' ");
    return $this->resultado;
}

//calcula o tempo
public function tempo($dataInicial, $dataFinal) {
    $this->resultado = mysql_query(" SELECT DATEDIFF( '$dataInicial', '$dataFinal' ) AS
dias");
    $dados = mysql_fetch_array($this->resultado);
    $tempo = $dados['dias'];
    //se a diferenca for igual a zero o tempo deve ser calculado em horas
    if ($tempo == 0) {

        $this->resultado = mysql_query(" SELECT TIMEDIFF( '$dataInicial', '$dataFinal' )
AS horas");
        $dados = mysql_fetch_array($this->resultado);
        $tempo = $dados['horas'];
        //remove sinal de negativo
        return strlen($tempo) > 8 ? $tempo = substr($tempo, 1, 8) : $tempo;
    } else {
        //remove sinal de negativo
        return $tempo < 0 ? ($tempo * -1) . ' dias' : $tempo . ' dias';
    }
}

//recupera a ultima posicao recebida
public function posicao() {
    $this->resultado = mysql_query("
    SELECT (`id_posicao`,`fk_usuario`,`nr_latitude`,`nr_longitude`,`dt_recebido`
    FROM `tb_posicao`
    WHERE `fk_usuario` ='$this->id_usuario'
    ORDER by id_posicao desc limit 1
    ");
    $dados = mysql_fetch_array($this->resultado);
    return $dados;
}

//verifica se ja existe convite solicitado
public function verificar() {

    $this->resultado = mysql_query("
    SELECT 1 AS rs
    FROM tb_grupo

```

```

        WHERE `fk_usuario_grupo` = '$this->id_usuario' AND `fk_usuario` = '$this-
>usuario' limit 1");
        $dados = mysql_fetch_array($this->resultado);
        if ($dados["rs"] != null) {
            return true;
        } else {
            return false;
        }
    }

    //atualiza o status para aprovado
    public function aprovar($param) {
        $this->resultado = mysql_query("
        UPDATE `tb_grupo` SET
        `fl_authorized` = 'S'
        WHERE `fk_usuario_grupo` = '$this->id_usuario' AND `fk_usuario` = '$this-
>id_usuario2'"
        );
        $dados = mysql_fetch_array($this->resultado);
        if ($dados["rs"] != null) {
            return true;
        } else {
            return false;
        }
    }

    //remove usuário do grupo
    public function remover($param) {
        $this->resultado = mysql_query("DELETE FROM tb_grupo WHERE
`fk_usuario_grupo` = '$this->id_usuario' AND `fk_usuario` = '$this->id_usuario2'"
        );
    }

    //retorna o número do usuario
    public function getID($usuario) {

        $this->resultado = mysql_query("SELECT id_usuario AS rs FROM tb_usuario WHERE
nm_usuario = '$usuario' ");
        $dados = mysql_fetch_array($this->resultado);
        if ($dados["rs"] != null) {
            return $dados["rs"];
        } else {

            return "false";
        }
    }

    //retorna as coordenadas do usuário logado
    public function main() {
        $this->resultado = mysql_query("

```

```

        SELECT *,DATE_FORMAT(( `dt_recebido` ), '%e/%m/%Y %H:%i:%s') as
dt_formatada
        FROM tb_posicao
        INNER JOIN tb_usuario ON id_usuario= fk_usuario
        WHERE fk_usuario = '$this->id_usuario'
        ORDER BY `id_posicao` desc limit 1 ");

        return $this->resultado;
    }

}

//fim classe ClassGrupo
?>

```

### **posicao.class.php**

**<?php**

```

/*
 * Classe responsável em realizar operacoes com a posicao
 */
require_once("conexao.class.php");
conexao :: conectar();

class Posicao {

    public $id_usuario;
    public $usuario;
    public $posicao1;
    public $posicao2;
    public $data;
    public $nr_latitude;
    public $nr_longitude;

    public function situacaoCadastro($param) {
        // se aceitar faca
        if ($param instanceof ClassName) {

        } else {
            //se negado faca
        }
    }

    function inserir() {

        $this->resultado = @ mysql_query(
            "
            INSERT INTO `tb_posicao` (

```

```

`id_posicao`,
`fk_usuario`,
`nr_latitude`,
`nr_longitude`,
`dt_recebido`
)
VALUES (
NULL,
$this->id_usuario, $this->nr_latitude, $this->nr_longitude,
CURRENT_TIMESTAMP
)
");

    if ($this->resultado) {
        return "OK";
    } else {

        return "ERRO";
    }
}

public function listar() {
    $this->resultado = mysql_query("
SELECT *,DATE_FORMAT(`dt_recebido`), '%e/%m/%Y %H:%i:%s') as dt_formatada
FROM tb_posicao
where fk_usuario = '$this->id_usuario' and DATE(`dt_recebido`) = '$this->data' ");

    return $this->resultado;
}

public function verificar() {

    $this->resultado = mysql_query("SELECT 1 AS rs FROM tb_grupo WHERE
`fk_usuario_grupo` = '$this->id_usuario' AND `fk_usuario` = '$this->usuario' limit 1");
    $dados = mysql_fetch_array($this->resultado);
    if ($dados["rs"] != null) {
        return true;
    } else {
        return false;
    }
}

public function getID($usuario) {

    $this->resultado = mysql_query("SELECT id_usuario AS rs FROM tb_usuario WHERE
nm_usuario = '$usuario' ");
    $dados = mysql_fetch_array($this->resultado);
    if ($dados["rs"] != null) {
        return $dados["rs"];
    } else {

```

```

        return "false";
    }
}

}

?>

```

### **xml.class.php**

**<?php**

```

/*
 * classe responsável em gerar o XML
 */
require_once("conexao.class.php");
conexao :: conectar();

class xml {

    public static $xml; //armazena o XML que vai ser gerado

    public static function XMLInicio() {
        self::$xml = new XMLWriter;
        //Cria área na memória para armazenar o XML
        self::$xml->openMemory();
        //Inicia o cabeçalho XML
        self::$xml->startDocument('1.0', 'iso-8859-1');
        //Inicia o primeiro elemento do XML
        self::$xml->startElement("dados");
    }

    public static function XMLFim() {
        //encerra o elemento </dados>
        self::$xml->endElement();
        header('Content-type: text/xml');
        # Imprime os dados armazenados
        print self::$xml->outputMemory(true);
    }

    public static function getXML($sql) {
        $resultado = $sql;
        self::XMLInicio();
        while ($dados = mysql_fetch_assoc($resultado)) {
            //inicia o elemnto <linha>
            self::$xml->startElement("linha");
            //obtem os nomes das tags
            $n = array_keys($dados);
            $c = count($dados);

```

```

    for ($i = 0; $i < $c; $i++) {
        //inserir elemento
        self::$xml->writeElement(strtolower($n[$i]), $dados[$n[$i]]);
    }
    //encerra o elemento </linha>
    self::$xml->endElement();
}

//encerra o xml e exibe as informações
self::XMLFim();
}

public static function getXML_Array($sql) {
    $resultado = $sql;
    //inicia o XML
    self::XMLInicio();
    for ($k = 0; $k < count($resultado); $k++) {
        //inicia o elemento <linha>
        self::$xml->startElement("linha");
        //obtem os nomes das tags
        $n = array_keys($resultado[$k]);
        //
        $c = count($resultado[$k]);
        for ($i = 0; $i < $c; $i++) {
            //inserir elemento
            self::$xml->writeElement(strtolower($n[$i]), $resultado[$k][$n[$i]]);
        }
        //encerra o elemento </linha>
        self::$xml->endElement();
    }
    //encerra o xml e exibe as informações
    self::XMLFim();
}

}

// fim class xml
?>

```

### grupo.js

```

//inserir os pontos dos usuarios no mapa
function ajaxXMLGrupoMapa(url,retorno){
    var retorno = document.getElementById(retorno);
    var xmlhttp;
    var txt,x,xx,i;
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
}

```

```

else
{
  // code for IE6, IE5
  xmlhttp=new XMLHttpRequest("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function()
{
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
  {
    txt="<table
border='1'><tr><tr><th>Usuário</th><th>Latitude</th><th>Longitude</th></tr>";
    x=xmlhttp.responseXML.documentElement.getElementsByTagName("linha");
    for (i=0;i<x.length;i++)
    {
      txt=txt + "<tr>";
      xx=x[i].getElementsByTagName("fk_usuario_grupo");
      {
        try
        {
          txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";
        }
        catch (er)
        {
          txt=txt + "<td> </td>";
        }
      }
      xx=x[i].getElementsByTagName("nr_longitude");
      {
        try
        {
          var lon = xx[0].firstChild.nodeValue ;
          txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";
        }
        catch (er)
        {
          txt=txt + "<td> </td>";
        }
      }
      xx=x[i].getElementsByTagName("nr_latitude");
      {
        try
        {
          txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

          var lat = xx[0].firstChild.nodeValue ;
        }
        catch (er)
        {

```

```

        txt=txt + "<td> </td>";
    }
}
xx=x[i].getElementsByTagName("nm_usuario");
var usuario = xx[0].firstChild.nodeValue;
xx=x[i].getElementsByTagName("dt_formatada");

addLatLng(lat ,lon,usuario+ "<br>Data:"+xx[0].firstChild.nodeValue);
txt=txt + "</tr>";
}
txt=txt + "</table>";

}
}
xmlhttp.open("GET",url,true);
xmlhttp.send();
}
//insere no mapa o usuário principal
function ajaxXMLMain(url,retorno){
    var retorno = document.getElementById(retorno);
    var xmlhttp;
    var txt,x,xx,i;
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {

            txt="<table
border='1'><tr><tr><th>Usuário</th><th>Latitude</th><th>Longitude</th></tr>";
            x=xmlhttp.responseXML.documentElement.getElementsByTagName("linha");
            for (i=0;i<x.length;i++)
            {
                txt=txt + "<tr>";

                xx=x[i].getElementsByTagName("nr_longitude");
                {
                    try
                    {
                        var lon = xx[0].firstChild.nodeValue ;
                        txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";
                    }
                }
            }
        }
    }
}

```

```

        catch (er)
        {
            txt=txt + "<td> </td>";
        }
    }
    xx=x[i].getElementsByTagName("nr_latitude");
    {
        try
        {
            txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

            var lat = xx[0].firstChild.nodeValue ;
        }
        catch (er)
        {
            txt=txt + "<td> </td>";
        }
    }
    xx=x[i].getElementsByTagName("nm_usuario");
    {
        try
        {
            var usuario = xx[0].firstChild.nodeValue;
            txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

        }
        catch (er)
        {
            txt=txt + "<td> </td>";
        }
    }
    document.getElementById('usuarioPrincipal').innerHTML = "<button
onClick='foco(" +lat+"," +lon+ ")' class='botaoFoco'>" + xx[0].firstChild.nodeValue +
"</button>";
    xx=x[i].getElementsByTagName("fk_usuario");
    xx=x[i].getElementsByTagName("dt_formatada");
    var data = xx[0].firstChild.nodeValue ;
    addLatLng(lat ,lon,usuario+"<br>Data: "+data);
    txt=txt + "</tr>";
}
txt=txt + "</table>";

}
}
xmlhttp.open("GET",url,true);
xmlhttp.send();
}
//lista usuarios do grupo
function ajaxXMLGrupo(url,retorno){
    var retorno = document.getElementById(retorno);

```

```

var xmlhttp;
var txt,x,xx,i;
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.onreadystatechange=function()
{
  if (xmlhttp.readyState===4 && xmlhttp.status===200)
  {

    txt="<table width='100%' border='1' cellpadding='0' cellspacing='1'
bordercolor='#666666'>"
    +" <tr class='textoBranco' >"
    +"<td bgcolor='3B5998'>Usuário</td>"
    +" <td bgcolor='3B5998' >Distância</td>"
    +" <td bgcolor='3B5998' >Tempo</td>"
    +" </tr>";
    x=xmlhttp.responseXML.documentElement.getElementsByTagName("linha");
    for (i=0;i<x.length;i++)
    {
      txt=txt + "<tr>";
      xx=x[i].getElementsByTagName("latitude");
      var lat = xx[0].firstChild.nodeValue ;
      xx=x[i].getElementsByTagName("longitude");
      var lon = xx[0].firstChild.nodeValue ;
      xx=x[i].getElementsByTagName("usuario");

      {
        try
        {
          txt=txt + "<td><button onClick='foco(" +lat+"," +lon+ ")' class='botaoFoco'>"
+ xx[0].firstChild.nodeValue + "</button></td>";

          }
        catch (er)
        {
          txt=txt + "<td> </td>";
        }
      }
      xx=x[i].getElementsByTagName("distancia");
      {
        try
        {
          var lon = xx[0].firstChild.nodeValue ;

```

```

        txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

    }
    catch (er)
    {
        txt=txt + "<td> </td>";
    }
}
xx=x[i].getElementsByTagName("tempo");
{
    try
    {
        txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

    }
    catch (er)
    {
        txt=txt + "<td> </td>";
    }
}

    txt=txt + "</tr>";
}
txt=txt + "</table>";
returno.innerHTML=txt;

}
}
xmlhttp.open("GET",url,true);
xmlhttp.send();
}
//lista os usuários pendentes
function ajaxXMLPendentes(url,retorno){
    var retorno = document.getElementById(retorno);
    var xmlhttp;
    var txt,x,xx,i;
    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {

```

```

txt="<table width='100%' border='1' cellpadding='0' cellspacing='1'
bordercolor='#666666'>"
+" <tr class='textoBranco' >"
+"<td bgcolor='3B5998'>Usuário</td>"
+" <td bgcolor='3B5998' ></td>"
+" <td bgcolor='3B5998' ></td>"
+" </tr>";
x=xmlhttp.responseXML.documentElement.getElementsByTagName("linha");
for (i=0;i<x.length;i++)
{
txt=txt + "<tr>";

xx=x[i].getElementsByTagName("nm_usuario");
{
try
{
txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";
}
catch (er)
{
txt=txt + "<td> </td>";
}
}
//
xx=x[i].getElementsByTagName("fl_authorized");

xb=x[i].getElementsByTagName("fk_usuario");
xa=x[i].getElementsByTagName("fk_usuario_grupo");

{
try
{
//remove a opcao aprovar do usuario que fez o convite
if(xx[0].firstChild.nodeValue == 'N'){
txt=txt + "<td> </td>";
}else{
txt=txt + "<td><button onClick=' aprovar(" + xa[0].firstChild.nodeValue
+"," + xb[0].firstChild.nodeValue+")' class='botaoAprovar'>Aprovar</button></td>";
}
txt=txt + "<td><button onClick=' negar(" + xa[0].firstChild.nodeValue +"," +
xb[0].firstChild.nodeValue+")' class='botaoRemover'>Negar</button></td>";
}
catch (er)
{
txt=txt + "<td> EE</td>";
}
}
txt=txt + "</tr>";
}
txt=txt + "</table>";

```

```

        //grava no elemento de resposta
        retorno.innerHTML=txt;

    }
}
xmlhttp.open("GET",url,true);
xmlhttp.send();
}
//aprova convite
function ajaxXMLStatus(url,retorno){
    var retorno = document.getElementById(retorno);
    var xmlhttp;
    var txt,x,xx,i;
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            x=xmlhttp.responseText;
            retorno.innerHTML=x;

        }
    }
    xmlhttp.open("GET",url,true);
    xmlhttp.send();
}
function listar(){
    var grupoId = document.getElementById('grupoId').value;
    //carrega os pontos no mapa

ajaxXMLGrupoMapa("include/ajaxGrupo.php?acao=listarMapa&grupoId="+grupoId,"grupo
");
    //carrega tabela do grupo
    ajaxXMLGrupo("include/ajaxGrupo.php?acao=listar&grupoId="+grupoId,"grupo");
    //carrega o ponto do usuario no mapa
    ajaxXMLMain("include/ajaxGrupo.php?acao=main&grupoId="+grupoId,"grupo");

}
// lista os usuários com os convites pendentes
function listarPendentes(){

```

```

    var grupoId = document.getElementById('grupoId').value;

    ajaxXMLPendientes("include/ajaxGrupo.php?acao=listarPendientes&grupoId="+grupoId,"pendentes");

}
//aprova convite
function aprovar(usuario,usuario2){

    var grupoId = document.getElementById('grupoId').value;
    var usuario = usuario2;

    ajaxXMLStatus("include/ajaxGrupo.php?acao=aprovar&grupoId="+grupoId+"&usuario2="+usuario,"pendentes");

}
//foca o mapa na coordenada
function foco (lat ,lon){
    map.setCenter(new google.maps.LatLng(lat ,lon));
}
//nega o convite
function negar(usuario,usuario2){

    var grupoId = document.getElementById('grupoId').value;
    var usuario = usuario2;

    ajaxXMLStatus("include/ajaxGrupo.php?acao=negar&grupoId="+grupoId+"&usuario2="+usuario,"pendentes");

}
// adiciona usuário ao grupo
function adicionar(){
    var grupoId = document.getElementById('grupoId').value;
    var usuario = document.getElementById('usuario').value;
    if (usuario.length != 0){

        ajax("include/ajaxGrupo.php?acao=cadastrar&usuario="+usuario+"&grupoId="+grupoId,"resposta");
    }else{
        alert( "Preencha o campo de convite");
    }

}

}
//realiza a atualizacao
function loop(){
    //atualiza a lista de usuarios

```

```

listar();
//atualiza a lista de usuarios pendentes
listarPendentes();
//limpa os pontos do mapa
deletePontos();
//atualiza a cada 45 segundos
setTimeout('loop()',60000);
}

```

### mapa.js

```

var map; //recebe o mapa
var markersArray = []; //array de coordenadas

function initialize() {
  //define o centro do mapa
  var sp = new google.maps.LatLng(-23.608586, -46.707921);
  var myOptions = {
    zoom: 7,
    center: sp,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };

  map = new google.maps.Map(document.getElementById('map_canvas'), myOptions);

}
//adiciona uma posicao no mapa
function addLatLng(longitude,latitude,label) {
  var marker = new google.maps.Marker({
    position: new google.maps.LatLng(longitude,latitude),

    map: map
  });
  //mostra uma janela com as coordenadas ao passar sobre um ponto
  google.maps.event.addListener(marker, 'mouseover', function() {
    infowindow.open(map,marker);
  });
  //remove a janela
  google.maps.event.addListener(marker, 'mouseout', function() {
    infowindow.close(map,marker);
  });
  var conteudo = '<div id="conteudo">'+
  "+(label) + "<br> Latitude:"+latitude+"<br> Longitude:"+longitude+
  '</div>';
  // vincula o conteudo
  var infowindow = new google.maps.InfoWindow({

```

```

        content: conteudo
    });
    markersArray.push(marker);
}

function deletePontos() {
    if (markersArray) {
        for (i in markersArray) {
            markersArray[i].setMap(null);
        }
        markersArray.length = 0;
    }
}
}

```

### **posicao.js**

```

function ajaxXMLMapa(url,retorno){
    var retorno = document.getElementById(retorno);
    var xmlhttp;
    var txt,x,xx,i;
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            //apaga pontos do mapa
            deletePontos();
            txt="<table width='100%' border='1' cellpadding='0' cellspacing='1'
bordercolor=#666666>"
            +" <tr class='textoBranco' >"
            +"<td bgcolor='3B5998'>Ordem</td>"
            +" <td bgcolor='3B5998' >Latitude</td>"
            +" <td bgcolor='3B5998' >Longitude</td>"
            +" </tr>";
            x = xmlhttp.responseXML.documentElement.getElementsByTagName("linha");
            for (i=0;i<x.length;i++)
            {
                txt=txt + "<tr>";
                xx=x[i].getElementsByTagName("fk_usuario");
                {
                    try

```

```

        {
            txt=txt + "<td>" + (i+1) + "</td>";
        }
        catch (er)
        {
            txt=txt + "<td> </td>";
        }
    }
    xx=x[i].getElementsByTagName("nr_longitude");
    {
        try
        {
            var lon = xx[0].firstChild.nodeValue ;
            txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

        }
        catch (er)
        {
            txt=txt + "<td> </td>";
        }
    }
    xx=x[i].getElementsByTagName("nr_latitude");
    {
        try
        {
            txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

            var lat = xx[0].firstChild.nodeValue ;
        }
        catch (er)
        {
            txt=txt + "<td> </td>";
        }
    }
    xx=x[i].getElementsByTagName("dt_formatada");
    addLatLng(lat ,lon,"Posição:" + (i+1)+"<br> Data: " + xx[0].firstChild.nodeValue );

    txt=txt + "</tr>";
}
//marca a ultima coordenada como o centro do mapa
map.setCenter(new google.maps.LatLng(lat ,lon));
txt=txt + "</table>";
//inserir na div
retorno.innerHTML=txt;

}

}
xmlhttp.open("GET",url,true);

```

```
    xmlhttp.send();
}

function listar(){
    var grupoId = document.getElementById('grupoId').value;
    var selecteddata = document.getElementById('data').value;
    //chamada ajax

    ajaxXMLMapa("include/ajaxMapa.php?acao=listar&grupoId="+grupoId+"&data="+selected
data,"resposta");

}
//atualiza a pagina a cada 60 segundos
function loop(){
    listar();

    setTimeout('loop()',600000);
}
```

## ANEXOS

Código fonte do programa que não são de minha autoria.

Esta no site W3C

([http://www.w3schools.com/ajax/tryit.asp?filename=tryajax\\_first](http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_first)) Acessado 20/01/2011

### **ajax.js**

```
function ajax(url,retorno){
    var retorno = document.getElementById(retorno);
    if (window.XMLHttpRequest)
        { // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp=new XMLHttpRequest();
        }
    else
        { // code for IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }

    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
            {
                retorno.innerHTML=xmlhttp.responseText;
            }
    }
    xmlhttp.open("GET",url,true);
    xmlhttp.send();
}
```

Código fonte gerado automaticamente pelo NetBeans para conexão com o WEB service.

### **autentica.java**

// This class was generated by 172 StubGenerator.

// Contents subject to change without notice.

// @generated

```
package comunicacao.autenticar;
```

```
public interface Autentica extends java.rmi.Remote {
    public java.lang.String autenticar(java.lang.String usuario, java.lang.String senha)
    throws java.rmi.RemoteException;
```

```
}
```

### **autentica\_stub.java**

// This class was generated by 172 StubGenerator.

// Contents subject to change without notice.

// @generated

```

package autenticar;

import javax.xml.rpc.JAXRPCException;
import javax.xml.namespace.QName;
import javax.microedition.xml.rpc.Operation;
import javax.microedition.xml.rpc.Type;
import javax.microedition.xml.rpc.ComplexType;
import javax.microedition.xml.rpc.Element;

public class Autentica_Stub implements comunicacao.autenticar.Autentica, javax.xml.rpc.Stub
{
    private String[] _propertyNames;
    private Object[] _propertyValues;

    public Autentica_Stub() {
        _propertyNames = new String[] {ENDPOINT_ADDRESS_PROPERTY};
        _propertyValues = new Object[] {"http://www.e-
making.com/projetoweb/ws/autenticar.php"};
    }

    public void _setProperty(String name, Object value) {
        int size = _propertyNames.length;
        for (int i = 0; i < size; ++i) {
            if (_propertyNames[i].equals(name)) {
                _propertyValues[i] = value;
                return;
            }
        }
        // Need to expand our array for a new property
        String[] newPropNames = new String[size + 1];
        System.arraycopy(_propertyNames, 0, newPropNames, 0, size);
        _propertyNames = newPropNames;
        Object[] newPropValues = new Object[size + 1];
        System.arraycopy(_propertyValues, 0, newPropValues, 0, size);
        _propertyValues = newPropValues;

        _propertyNames[size] = name;
        _propertyValues[size] = value;
    }

    public Object _getProperty(String name) {
        for (int i = 0; i < _propertyNames.length; ++i) {
            if (_propertyNames[i].equals(name)) {
                return _propertyValues[i];
            }
        }
        if (ENDPOINT_ADDRESS_PROPERTY.equals(name) ||
        USERNAME_PROPERTY.equals(name) || PASSWORD_PROPERTY.equals(name)) {
            return null;
        }
    }
}

```

```

    }
    if (SESSION_MAINTAIN_PROPERTY.equals(name)) {
        return new java.lang.Boolean(false);
    }
    throw new JAXRPCException("Stub does not recognize property: "+name);
}

protected void _prepOperation(Operation op) {
    for (int i = 0; i < _propertyNames.length; ++i) {
        op.setProperty(_propertyNames[i], _propertyValues[i].toString());
    }
}

//
// Begin user methods
//

    public java.lang.String autenticar(java.lang.String usuario, java.lang.String senha)
throws java.rmi.RemoteException {
    // Copy the incoming values into an Object array if needed.
    Object[] inputObject = new Object[2];
    inputObject[0] = usuario;
    inputObject[1] = senha;

    Operation op = Operation.newInstance(_qname_autenticar, _type_autenticar,
_type_autenticarResponse);
    _prepOperation(op);
    op.setProperty(Operation.SOAPACTION_URI_PROPERTY, "http://www.e-
making.com/projetoweb/ws/autenticar");
    Object resultObj;
    try {
        resultObj = op.invoke(inputObject);
    } catch (JAXRPCException e) {
        Throwable cause = e.getLinkedCause();
        if (cause instanceof java.rmi.RemoteException) {
            throw (java.rmi.RemoteException) cause;
        }
        throw e;
    }
    java.lang.String result;
    // Convert the result into the right Java type.
    // Unwrapped return value
    Object msgRespostaObj = ((Object[])resultObj)[0];
    result = (java.lang.String)msgRespostaObj;
    return result;
}
//
// End user methods
//

```

```

    protected static final QName _qname_msgResposta = new QName("",
"msgResposta");
    protected static final QName _qname_senha = new QName("", "senha");
    protected static final QName _qname_usuario = new QName("", "usuario");
    protected static final QName _qname_autenticar = new QName("http://www.e-
making.com/projetoweb/ws/", "autenticar");
    protected static final QName _qname_autenticarResponse = new
QName("http://www.e-making.com/projetoweb/ws/", "autenticarResponse");
    protected static final Element _type_autenticar;
    protected static final Element _type_autenticarResponse;
    static {
        // Create all of the Type's that this stub uses, once.
        Element _type_usuario;
        _type_usuario = new Element(_qname_usuario, Type.STRING);
        Element _type_senha;
        _type_senha = new Element(_qname_senha, Type.STRING);
        ComplexType _complexType_entrada;
        _complexType_entrada = new ComplexType();
        _complexType_entrada.elements = new Element[2];
        _complexType_entrada.elements[0] = _type_usuario;
        _complexType_entrada.elements[1] = _type_senha;
        _type_autenticar = new Element(_qname_autenticar, _complexType_entrada);
        Element _type_msgResposta;
        _type_msgResposta = new Element(_qname_msgResposta, Type.STRING);
        ComplexType _complexType_resposta;
        _complexType_resposta = new ComplexType();
        _complexType_resposta.elements = new Element[1];
        _complexType_resposta.elements[0] = _type_msgResposta;
        _type_autenticarResponse = new Element(_qname_autenticarResponse,
_complexType_resposta);
    }
}

```

### **Entrada.java**

```

// This class was generated by the JAXRPC SI, do not edit.
// Contents subject to change without notice.
// JSR-172 Reference Implementation wscompile 1.0, using: JAX-RPC Standard
Implementation (1.1, build R59)

```

```
package autenticar;
```

```

public class Entrada {
    protected java.lang.String usuario;
    protected java.lang.String senha;

    public Entrada() {
    }

    public Entrada(java.lang.String usuario, java.lang.String senha) {

```

```

        this.usuario = usuario;
        this.senha = senha;
    }

    public java.lang.String getUsuario() {
        return usuario;
    }

    public void setUsuario(java.lang.String usuario) {
        this.usuario = usuario;
    }

    public java.lang.String getSenha() {
        return senha;
    }

    public void setSenha(java.lang.String senha) {
        this.senha = senha;
    }
}
Resposta.java
// This class was generated by the JAXRPC SI, do not edit.
// Contents subject to change without notice.
// JSR-172 Reference Implementation wscompile 1.0, using: JAX-RPC Standard
Implementation (1.1, build R59)

package autenticar;

public class Resposta {
    protected java.lang.String msgResposta;

    public Resposta() {
    }

    public Resposta(java.lang.String msgResposta) {
        this.msgResposta = msgResposta;
    }

    public java.lang.String getMsgResposta() {
        return msgResposta;
    }

    public void setMsgResposta(java.lang.String msgResposta) {
        this.msgResposta = msgResposta;
    }
}

package comunicacao.grupo;

```

**Entrada.java**

```

public class Entrada {
    protected java.lang.String usuario;

    public Entrada() {
    }

    public Entrada(java.lang.String usuario) {
        this.usuario = usuario;
    }

    public java.lang.String getUsuario() {
        return usuario;
    }

    public void setUsuario(java.lang.String usuario) {
        this.usuario = usuario;
    }
}

```

**Grupo\_PortType.java**

```

// This class was generated by 172 StubGenerator.
// Contents subject to change without notice.
// @generated

```

```

package comunicacao.grupo;

```

```

public interface Grupo_PortType extends java.rmi.Remote {
    public comunicacao.grupo.Resposta[] lista(java.lang.String usuario) throws
    java.rmi.RemoteException;
}

```

**Grupo\_PortType\_Stub.java**

```

// This class was generated by 172 StubGenerator.
// Contents subject to change without notice.
// @generated

```

```

package comunicacao.grupo;

```

```

import javax.xml.rpc.JAXRPCException;
import javax.xml.namespace.QName;
import javax.microedition.xml.rpc.Operation;
import javax.microedition.xml.rpc.Type;
import javax.microedition.xml.rpc.ComplexType;
import javax.microedition.xml.rpc.Element;

```

```

public class Grupo_PortType_Stub implements comunicacao.grupo.Grupo_PortType,
    javax.xml.rpc.Stub {
    private String[] _propertyNames;
    private Object[] _propertyValues;
}

```

```

public Grupo_PortType_Stub() {
    _propertyNames = new String[] {ENDPOINT_ADDRESS_PROPERTY};
    _propertyValues = new Object[] {"http://www.e-
making.com/projetoweb/ws/grupo.php"};
}

public void _setProperty(String name, Object value) {
    int size = _propertyNames.length;
    for (int i = 0; i < size; ++i) {
        if (_propertyNames[i].equals(name)) {
            _propertyValues[i] = value;
            return;
        }
    }
    // Need to expand our array for a new property
    String[] newPropNames = new String[size + 1];
    System.arraycopy(_propertyNames, 0, newPropNames, 0, size);
    _propertyNames = newPropNames;
    Object[] newPropValues = new Object[size + 1];
    System.arraycopy(_propertyValues, 0, newPropValues, 0, size);
    _propertyValues = newPropValues;

    _propertyNames[size] = name;
    _propertyValues[size] = value;
}

public Object _getProperty(String name) {
    for (int i = 0; i < _propertyNames.length; ++i) {
        if (_propertyNames[i].equals(name)) {
            return _propertyValues[i];
        }
    }
    if (ENDPOINT_ADDRESS_PROPERTY.equals(name) ||
    USERNAME_PROPERTY.equals(name) || PASSWORD_PROPERTY.equals(name)) {
        return null;
    }
    if (SESSION_MAINTAIN_PROPERTY.equals(name)) {
        return new java.lang.Boolean(false);
    }
    throw new JAXRPCException("Stub does not recognize property: "+name);
}

protected void _prepOperation(Operation op) {
    for (int i = 0; i < _propertyNames.length; ++i) {
        op.setProperty(_propertyNames[i], _propertyValues[i].toString());
    }
}

//

```

```

// Begin user methods
//

public comunicacao.grupo.Resposta[] lista(java.lang.String usuario) throws
java.rmi.RemoteException {
    // Copy the incoming values into an Object array if needed.
    Object[] inputObject = new Object[1];
    inputObject[0] = usuario;

    Operation op = Operation.newInstance(_qname_lista, _type_lista,
_type_listaResponse);
    _prepOperation(op);
    op.setProperty(Operation.SOAPACTION_URI_PROPERTY, "http://www.e-
making.com/projetoweb/ws/lista");
    Object resultObj;
    try {
        resultObj = op.invoke(inputObject);
    } catch (JAXRPCException e) {
        Throwable cause = e.getLinkedCause();
        if (cause instanceof java.rmi.RemoteException) {
            throw (java.rmi.RemoteException) cause;
        }
        throw e;
    }
    comunicacao.grupo.Resposta[] result;
    // Convert the result into the right Java type.
    // Unwrapped return value
    Object[] saidaObj = (Object[]) ((Object[])resultObj)[0];
    if (saidaObj == null) {
        result = null;
    } else {
        int respostaArraySize = saidaObj.length;
        result = new comunicacao.grupo.Resposta[respostaArraySize];
        for (int respostaArrayIndex = 0;
            respostaArrayIndex < respostaArraySize;
            ++respostaArrayIndex) {
            if (saidaObj[respostaArrayIndex] == null) {
                result[respostaArrayIndex] = null;
            } else {
                result[respostaArrayIndex] = new
comunicacao.grupo.Resposta();
                java.lang.String string;
                Object usuarioObj =
((Object[])saidaObj[respostaArrayIndex])[0];
                string = (java.lang.String)usuarioObj;
                result[respostaArrayIndex].setUsuario(string);
                java.lang.String string2;
                Object distanciaObj =
((Object[])saidaObj[respostaArrayIndex])[1];
                string2 = (java.lang.String)distanciaObj;

```

```

        result[respostaArrayIndex].setDistancia(string2);
        java.lang.String string3;
        Object tempoObj =
((Object[])saidaObj[respostaArrayIndex])[2];
        string3 = (java.lang.String)tempoObj;
        result[respostaArrayIndex].setTempo(string3);
    }
}
return result;
}
//
// End user methods
//

protected static final QName _qname_distancia = new QName("", "distancia");
protected static final QName _qname_saida = new QName("", "saida");
protected static final QName _qname_tempo = new QName("", "tempo");
protected static final QName _qname_usuario = new QName("", "usuario");
protected static final QName _qname_lista = new QName("http://www.e-
making.com/projetoweb/ws/", "lista");
protected static final QName _qname_listaResponse = new QName("http://www.e-
making.com/projetoweb/ws/", "listaResponse");
protected static final Element _type_lista;
protected static final Element _type_listaResponse;
static {
    // Create all of the Type's that this stub uses, once.
    Element _type_usuario;
    _type_usuario = new Element(_qname_usuario, Type.STRING);
    ComplexType _complexType_entrada;
    _complexType_entrada = new ComplexType();
    _complexType_entrada.elements = new Element[1];
    _complexType_entrada.elements[0] = _type_usuario;
    _type_lista = new Element(_qname_lista, _complexType_entrada);
    Element _type_distancia;
    _type_distancia = new Element(_qname_distancia, Type.STRING);
    Element _type_tempo;
    _type_tempo = new Element(_qname_tempo, Type.STRING);
    ComplexType _complexType_resposta;
    _complexType_resposta = new ComplexType();
    _complexType_resposta.elements = new Element[3];
    _complexType_resposta.elements[0] = _type_usuario;
    _complexType_resposta.elements[1] = _type_distancia;
    _complexType_resposta.elements[2] = _type_tempo;
    Element _type_saida;
    _type_saida = new Element(_qname_saida, _complexType_resposta, 1, -1,
false);
    ComplexType _complexType_listaResponse;
    _complexType_listaResponse = new ComplexType();
    _complexType_listaResponse.elements = new Element[1];

```

```

        _complexType_listaResponse.elements[0] = _type_saida;
        _type_listaResponse = new Element(_qname_listaResponse,
        _complexType_listaResponse);
    }
}

```

### **ListaResponse.java**

```

// This class was generated by the JAXRPC SI, do not edit.
// Contents subject to change without notice.
// JSR-172 Reference Implementation wscompile 1.0, using: JAX-RPC Standard
Implementation (1.1, build R59)

```

```
package comunicacao.grupo;
```

```

public class ListaResponse {
    protected comunicacao.grupo.Resposta[] saida;

    public ListaResponse() {
    }

    public ListaResponse(comunicacao.grupo.Resposta[] saida) {
        this.saida = saida;
    }

    public comunicacao.grupo.Resposta[] getSaida() {
        return saida;
    }

    public void setSaida(comunicacao.grupo.Resposta[] saida) {
        this.saida = saida;
    }
}

```

### **Resposta.java**

```

// This class was generated by the JAXRPC SI, do not edit.
// Contents subject to change without notice.
// JSR-172 Reference Implementation wscompile 1.0, using: JAX-RPC Standard
Implementation (1.1, build R59)

```

```
package comunicacao.grupo;
```

```

public class Resposta {
    protected java.lang.String usuario;
    protected java.lang.String distancia;
    protected java.lang.String tempo;

    public Resposta() {
    }

    public Resposta(java.lang.String usuario, java.lang.String distancia, java.lang.String tempo)
    {

```

```

        this.usuario = usuario;
        this.distancia = distancia;
        this.tempo = tempo;
    }

    public java.lang.String getUsuario() {
        return usuario;
    }

    public void setUsuario(java.lang.String usuario) {
        this.usuario = usuario;
    }

    public java.lang.String getDistancia() {
        return distancia;
    }

    public void setDistancia(java.lang.String distancia) {
        this.distancia = distancia;
    }

    public java.lang.String getTempo() {
        return tempo;
    }

    public void setTempo(java.lang.String tempo) {
        this.tempo = tempo;
    }
}

```

Entrada.posicao

// This class was generated by the JAXRPC SI, do not edit.

// Contents subject to change without notice.

// JSR-172 Reference Implementation wscompile 1.0, using: JAX-RPC Standard Implementation (1.1, build R59)

package posicao;

```

public class Entrada {
    protected java.lang.String usuario;
    protected java.lang.String latitude;
    protected java.lang.String longitude;

    public Entrada() {
    }

    public Entrada(java.lang.String usuario, java.lang.String latitude, java.lang.String longitude) {
        this.usuario = usuario;
    }
}

```

```

        this.latitude = latitude;
        this.longitude = longitude;
    }

    public java.lang.String getUsuario() {
        return usuario;
    }

    public void setUsuario(java.lang.String usuario) {
        this.usuario = usuario;
    }

    public java.lang.String getLatitude() {
        return latitude;
    }

    public void setLatitude(java.lang.String latitude) {
        this.latitude = latitude;
    }

    public java.lang.String getLongitude() {
        return longitude;
    }

    public void setLongitude(java.lang.String longitude) {
        this.longitude = longitude;
    }
}
Posicao_PortType.java
// This class was generated by 172 StubGenerator.
// Contents subject to change without notice.
// @generated

package posicao;

public interface Posicao_PortType extends java.rmi.Remote {
    public java.lang.String envia(java.lang.String usuario, java.lang.String latitude,
java.lang.String longitude) throws java.rmi.RemoteException;

}
Posicao_PortType_Stub.java
// This class was generated by 172 StubGenerator.
// Contents subject to change without notice.
// @generated

package posicao;

import javax.xml.rpc.JAXRPCException;
import javax.xml.namespace.QName;
import javax.xml.rpc.Operation;

```

```

import javax.microedition.xml.rpc.Type;
import javax.microedition.xml.rpc.ComplexType;
import javax.microedition.xml.rpc.Element;

public class Posicao_PortType_Stub implements posicao.Posicao_PortType,
javax.xml.rpc.Stub {
    private String[] _propertyNames;
    private Object[] _propertyValues;

    public Posicao_PortType_Stub() {
        _propertyNames = new String[] {ENDPOINT_ADDRESS_PROPERTY};
        _propertyValues = new Object[] {"http://www.e-
making.com/projetoweb/ws/posicao.php"};
    }

    public void _setProperty(String name, Object value) {
        int size = _propertyNames.length;
        for (int i = 0; i < size; ++i) {
            if (_propertyNames[i].equals(name)) {
                _propertyValues[i] = value;
                return;
            }
        }
        // Need to expand our array for a new property
        String[] newPropNames = new String[size + 1];
        System.arraycopy(_propertyNames, 0, newPropNames, 0, size);
        _propertyNames = newPropNames;
        Object[] newPropValues = new Object[size + 1];
        System.arraycopy(_propertyValues, 0, newPropValues, 0, size);
        _propertyValues = newPropValues;

        _propertyNames[size] = name;
        _propertyValues[size] = value;
    }

    public Object _getProperty(String name) {
        for (int i = 0; i < _propertyNames.length; ++i) {
            if (_propertyNames[i].equals(name)) {
                return _propertyValues[i];
            }
        }
        if (ENDPOINT_ADDRESS_PROPERTY.equals(name) ||
USERNAME_PROPERTY.equals(name) || PASSWORD_PROPERTY.equals(name)) {
            return null;
        }
        if (SESSION_MAINTAIN_PROPERTY.equals(name)) {
            return new java.lang.Boolean(false);
        }
        throw new JAXRPCException("Stub does not recognize property: "+name);
    }
}

```

```

protected void _prepOperation(Operation op) {
    for (int i = 0; i < _propertyNames.length; ++i) {
        op.setProperty(_propertyNames[i], _propertyValues[i].toString());
    }
}

//
// Begin user methods
//

public java.lang.String envia(java.lang.String usuario, java.lang.String latitude,
java.lang.String longitude) throws java.rmi.RemoteException {
    // Copy the incoming values into an Object array if needed.
    Object[] inputObject = new Object[3];
    inputObject[0] = usuario;
    inputObject[1] = latitude;
    inputObject[2] = longitude;

    Operation op = Operation.newInstance(_qname_envia, _type_envia,
_type_enviaResponse);
    _prepOperation(op);
    op.setProperty(Operation.SOAPACTION_URI_PROPERTY, "http://www.e-
making.com/projetoweb/ws/envia");
    Object resultObj;
    try {
        resultObj = op.invoke(inputObject);
    } catch (JAXRPCException e) {
        Throwable cause = e.getLinkedCause();
        if (cause instanceof java.rmi.RemoteException) {
            throw (java.rmi.RemoteException) cause;
        }
        throw e;
    }
    java.lang.String result;
    // Convert the result into the right Java type.
    // Unwrapped return value
    Object msgRespostaObj = ((Object[])resultObj)[0];
    result = (java.lang.String)msgRespostaObj;
    return result;
}

//
// End user methods
//

protected static final QName _qname_latitude = new QName("", "latitude");
protected static final QName _qname_longitude = new QName("", "longitude");
protected static final QName _qname_msgResposta = new QName("",
"msgResposta");
protected static final QName _qname_usuario = new QName("", "usuario");

```

```

    protected static final QName _qname_envia = new QName("http://www.e-
making.com/projetoweb/ws/", "envia");
    protected static final QName _qname_enviaResponse = new QName("http://www.e-
making.com/projetoweb/ws/", "enviaResponse");
    protected static final Element _type_envia;
    protected static final Element _type_enviaResponse;
    static {
        // Create all of the Type's that this stub uses, once.
        Element _type_usuario;
        _type_usuario = new Element(_qname_usuario, Type.STRING);
        Element _type_latitude;
        _type_latitude = new Element(_qname_latitude, Type.STRING);
        Element _type_longitude;
        _type_longitude = new Element(_qname_longitude, Type.STRING);
        ComplexType _complexType_entrada;
        _complexType_entrada = new ComplexType();
        _complexType_entrada.elements = new Element[3];
        _complexType_entrada.elements[0] = _type_usuario;
        _complexType_entrada.elements[1] = _type_latitude;
        _complexType_entrada.elements[2] = _type_longitude;
        _type_envia = new Element(_qname_envia, _complexType_entrada);
        Element _type_msgResposta;
        _type_msgResposta = new Element(_qname_msgResposta, Type.STRING);
        ComplexType _complexType_resposta;
        _complexType_resposta = new ComplexType();
        _complexType_resposta.elements = new Element[1];
        _complexType_resposta.elements[0] = _type_msgResposta;
        _type_enviaResponse = new Element(_qname_enviaResponse,
_complexType_resposta);
    }
}

```

```

}

```

### **Resposta.java**

```

// This class was generated by the JAXRPC SI, do not edit.

```

```

// Contents subject to change without notice.

```

```

// JSR-172 Reference Implementation wscompile 1.0, using: JAX-RPC Standard
Implementation (1.1, build R59)

```

```

package posicao;

```

```

public class Resposta {
    protected java.lang.String msgResposta;

    public Resposta() {
    }

    public Resposta(java.lang.String msgResposta) {
        this.msgResposta = msgResposta;
    }
}

```

```
public java.lang.String getMsgResposta() {  
    return msgResposta;  
}  
  
public void setMsgResposta(java.lang.String msgResposta) {  
    this.msgResposta = msgResposta;  
}  
}
```