

Samuel Moreira Marques

Sistema para Localização de Dispositivos Móveis utilizando Android, Asp.Net, C# e Google Maps

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Itaúna, como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Itaúna

10 de junho de 2013



UNIVERSIDADE DE ITAÚNA
FACULDADE DE ENGENHARIA

FOLHA DE APROVAÇÃO

Sistema para Localização de Dispositivos Móveis utilizando
Android, Asp.Net, C# e Google Maps

Samuel Moreira Marques

Monografia defendida e aprovada pela banca examinadora.

Paulo de Tarso Gomide Castro Silva - Coordenador – Projeto Orientado II
Faculdade de Engenharia - UIT

Samuel Evangelista Lima de Oliveira – Orientador
Faculdade de Engenharia – UIT

Itaúna, 10 de junho de 2013.

Resumo

Apresentar à comunidade uma solução que auxilie na localização de veículos, pessoas e rastreamento em geral, possuindo um preço acessível de implementação e manutenção é necessário. O projeto proposto nesta monografia é parte integrante de uma solução que visa suprir estas necessidades, utilizando tecnologias atuais de maneira integrada.

A solução para localização de dispositivos móveis como celulares, smartphones ou tablets que possuem sistema operacional Android, encontra-se dividida em dois módulos, que por hora serão chamados de Global Tracker Location e Global Tracker Web.

O Global Tracker Location é responsável por obter a localização geográfica de um celular ou smartphone, utilizando um receptor GPS (*Global Position System*) integrado ao dispositivo móvel ou utilizando a rede de telefonia móvel juntamente da técnica de triangulação de antenas. Basicamente o software instalado no dispositivo móvel envia as coordenadas geográficas obtidas a um web service através de uma requisição SOAP (*Simple Object Access Protocol*).

O Global Tracker Web é responsável pela persistência das informações enviadas pelo dispositivo móvel ao servidor. O mesmo também fornece uma interface gráfica de alto nível para que um usuário administrador realize cadastros necessários e para que as informações enviadas pelo dispositivo móvel sejam persistidas. Outra funcionalidade deste módulo é a interface para que a localização do dispositivo móvel seja visualizada a partir de um mapa.

Outro fator observado nesta monografia é o desenvolvimento de uma aplicação ASP.Net utilizando tecnologias integradas como Web Service, dentre outras.

Palavras chave: Android, ASP.Net, Web Services, Google Maps, Subgurim.Net, Xamarin.Android, Desenvolvimento de Software.

Abstract

Show to community an application that assists to find vehicles, peoples and other things, that have a small price and few difficulty to development and maintenance is the objective of this academic work.

This project uses current technologies integrated, divided in two modules, which now will be called Global Location Tracker and Tracker Global Web.

The Global Location is responsible for obtaining geographic location of a mobile phone or smartphone using a GPS (Global Position System) integrated with a mobile device or using the telephone network along of technique Radio Direction Finder. Basically the software installed on the mobile device sends the geographic coordinates obtained from a web service through a SOAP request (Simple Object Access Protocol).

The Global Tracker Web is responsible for the persistence of informations that was sent by the mobile devices to the server. Also provides a front end for a user perform management activities like a visualization the location of the mobile device in a map.

Another thing that is view in this work is the develop of an application using integrated technologies like ASP.Net Web Service and others.

Tags: Android, ASP.Net, Web Services, Google Maps, Subgurim.Net, Xamarin.Android, Software Development.

Agradecimentos

Por mais esta etapa vencida, agradeço primeiramente a Deus, sempre presente em minha vida.

Aos meus pais, Clarice e Hélio pelo amor, carinho e apoio. À minha esposa Thaís e meu filho Henrique pelos momentos de silêncio, compreensão, apoio e amor incondicional. À minha irmã Maria Theresa pelas conversas descontraídas nos momentos de relaxamento e descanso.

À professora Milene Barbosa Carvalho e ao professor Samuel E. L. de Oliveira ensinamentos e orientações na condução e realização deste projeto.

Aos colegas de classe Diogo Damiani, Luciano Rodrigues, Monique Aparecida e Raphael de Castro pelos momentos que compartilhamos com alegria e companheirismo.

Aos grandes amigos, que tenho como irmãos Jhonatan Fernando, Diogo Damiani e Luciano Rodrigues, sempre com uma palavra de apoio nas horas certas.

Aos funcionários da Faculdade de Engenharia da UIT, em especial ao Sr. Roselé Fernando de Andrade, sempre dispostos a ajudar.

Enfim, muito obrigado a todos!

Sumário

Lista de Figuras	iv
1 Introdução.....	1
1.1 Global Tracker Location	2
1.2 Global Tracker Web	2
1.3 Integração entre Módulos.....	3
1.4 Vantagens.....	3
1.5 Organização	4
2 Tecnologias	6
2.1 .Net Framework.....	6
2.2 Linguagem C#.....	7
2.3 Linguagem ASP.NET	9
2.4 Visual Studio 2010.....	10
2.5 Xamarin	11
2.6 Subgurim.Net.....	11
2.7 Google Maps	11
3 <i>Global Tracker Location</i>.....	12
3.1 Soluções e Trabalhos Relacionados.....	12
3.2 Princípios de Funcionamento	13
3.2.1 <i>Configuração do Global Tracker Location</i>	13
3.2.2 <i>Utilização e funcionamento do Global Tracker Location</i>	16
3.3 Processo Unificado.....	16
3.4 Desenvolvimento	17
4 <i>Global Tracker Web</i>	21

4.1	Princípios de Funcionamento	22
4.1.1	Acesso ao <i>Global Tracker Web</i>	22
4.2	Processo Unificado.....	29
4.3	Desenvolvimento	30
4.3.1	<i>Global Tracker – Web Service</i>	30
4.3.2	<i>Global Tracker – Web Site</i>	31
5	Estudos de Aplicação.....	34
6	Análise Experimental.....	35
7	Conclusões.....	37
	Bibliografia	38
	Anexos	41
	Anexo A – <i>SRS (System Requirement Specification)</i>	41
	Anexo B – Especificação de Caso de Uso	57
	Anexo C – Código Fonte Comentado do <i>Global Tracker Mobile</i>	73
	Anexo D – Código Fonte Comentado do <i>Global Tracker Web Service</i>	82
	Anexo E – Código Fonte Comentado do <i>Global Tracker Web</i>	106
	Anexo F – Script para Criação do Banco de Dados <i>Global Tracker</i>	116

Lista de Figuras

Figura 1 - Diagrama de blocos da solução Global Tracker	3
Figura 2 - Diagrama de blocos da solução Global Tracker Location	12
Figura 3 – Ícone de Acesso ao aplicativo Android Global Tracker Mobile.....	14
Figura 4 – Tela inicial e de configuração do aplicativo Android Global Tracker Mobile.	15
Figura 5 – Arquitetura de compilação e execução de uma aplicação gerada com Xamarin.Android.	18
Figura 6 – Diagrama das principais classes do projeto GlobalTracker.Mobile.	20
Figura 7 - Diagrama de blocos da solução Global Tracker Web	21
Figura 8 – Página inicial do portal Global Tracker.....	22
Figura 9 – Página para informar login de acesso ao sistema Global Tracker Web. ...	23
Figura 10 – Página principal do Global Tracker Web.	24
Figura 11 – Listagem de dispositivos móveis que o usuário logado tem acesso	25
Figura 12 – Histórico de Navegação do Dispositivo utilizando a opção Online.	26
Figura 13 – Histórico de Navegação do Dispositivo utilizando a opção 5 Últimas. ...	26
Figura 14 – Definição da cerca eletrônica.	27
Figura 15 – Marcação de Cerca Eletrônica: 1º clique em Divinópolis, 2º clique em Cons. Lafaiete, 3º clique em Itabira, 4º clique em Sete Lagoas.	28
Figura 16 – Página para visualizar histórico de alertas.	29
Figura 17 – Diagrama das principais classes do projeto GlobalTracker.WebService.	31
Figura 18 – Diagrama das principais classes do projeto GlobalTracker.WebService.	33
Figura 19 – Comparação de coordenadas obtidas pela rede de telefonia celular e GPS.....	36

Capítulo 1

1 Introdução

A insegurança é um fator preocupante para a sociedade brasileira, indivíduos, órgãos públicos ou privados estão à mercê de atentados por demais conhecidos e que se repetem diariamente. Um caminhão de carga é assaltado em alguma rodovia do país, são levadas as mercadorias e muitas vezes o próprio veículo, que provavelmente será negociado com quadrilhas receptoras de veículos roubados ou entregue a oficinas de desmanches, para venda separada das peças.

Outro atentado conhecido se repete quando um assaltante rouba um veículo de passeio, a intenção pode ser utilizá-lo em ações criminosas e abandoná-lo posteriormente. O pior pode acontecer quando o roubo do veículo se transforma em um sequestro relâmpago, combinando violência física e psicológica.

Esta situação de insegurança é reflexo de problemas sociais graves, que insistem em não serem resolvidos pelos governantes. Felizmente a vida moderna oferece excelentes aliados para coibir e auxiliar na prevenção destas ações, uma delas é a Tecnologia.

O objetivo deste trabalho é desenvolver uma solução para rastreamento de dispositivos móveis, especificamente celulares, smartphones ou tablets. Esta é desenvolvida de maneira a se adaptar a qualquer necessidade de rastreamento, seja de veículos, pessoas ou cargas, minimizando os esforços de desenvolvimento, implantação e manutenção.

A solução de rastreamento de dispositivos móveis mencionada recebeu o nome de Global Tracker e encontra-se dividida em dois módulos: Global Tracker Location, Global Tracker Web.

O Global Tracker Location é o módulo responsável por obter o posicionamento geográfico do celular e enviá-lo a um servidor que irá persistir estas informações.

O Global Tracker Web é o módulo responsável por persistir as informações correspondentes a cada celular rastreado, permitindo que a mesma seja visualizada por meio de uma interface gráfica de acesso a estas informações de maneira

simples e funcional aos usuários através de um browser de acesso à internet. Além disso, este módulo, também é responsável por prover a interface de comunicação para que os dados necessários sejam enviados pelo Global Tracker Location.

1.1 Global Tracker Location

Com o advento da computação móvel, novos dispositivos e tecnologias surgem diariamente. O surgimento de dispositivos móveis que usam sistemas operacionais Android é um dos principais pilares que proporciona o desenvolvimento do Global Tracker Location.

Para determinar o posicionamento do celular será desenvolvido um software Android que se receberá informações de um receptor *GPS* ou utilizará a técnica de triangulação de antenas da rede de telefonia móvel. Este recebe informações referentes à localização, velocidade, altitude, dentre outras.

As informações obtidas são enviadas ao servidor através de um web service utilizando requisição *SOAP* através do acesso à internet provido pela operadora de telefonia móvel.

Para o pleno funcionamento deste módulo é necessário um dispositivo móvel que utilize o sistema operacional Android e possua um receptor *GPS* integrado. Caso o dispositivo não possua um receptor *GPS*, a solução também funcionará, porém a precisão da localização não será tão apurada por que utilizará triangulação de antenas.

Para realização dos testes utilizou-se os celulares Samsung Galaxy SII Lite e o software Global Tracker Location. Os componentes necessários, hardware e software, para este módulo são mostrados na Figura 1.

1.2 Global Tracker Web

O Global Tracker Web é um software desenvolvido em Asp.Net responsável por prover uma interface gráfica alto nível com um usuário que desejar monitorar um celular que está sendo rastreado.

Utilizando um browser o usuário faz uma autenticação no sistema e seleciona um dos dispositivos móveis que deseja monitorar e assim poderá consultar o status

atual do mesmo, visualizando seu deslocamento através de um mapa que será obtido em um servidor de mapas.

1.3 Integração entre Módulos

Os módulos Global Tracker Location e Web estão fracamente acoplados e oferecem uma estrutura concisa baseadas nas redes celulares e internet. O diagrama de blocos mostrado na Figura 1 retrata a integração entre os mesmos.

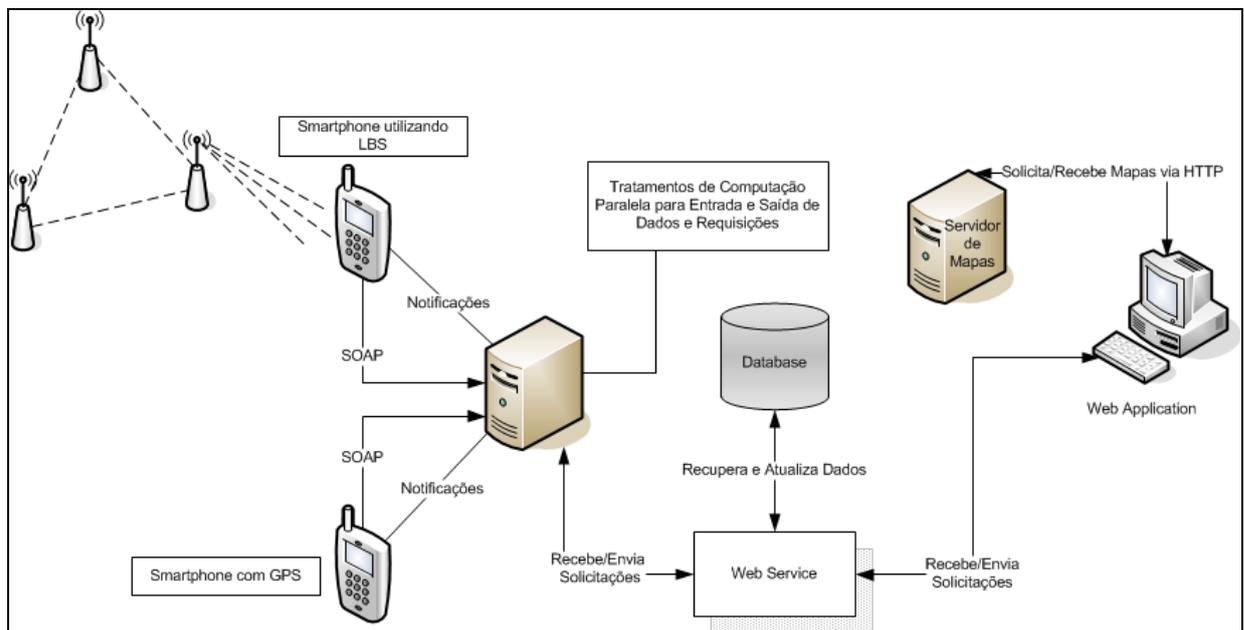


Figura 1 - Diagrama de blocos da solução Global Tracker

1.4 Vantagens

Existem diversas soluções atualmente no mercado que promovem o monitoramento e rastreamento em diversos seguimentos, porém a maioria delas não possui código fonte disponível para análises acadêmicas ou adaptações específicas de um determinado ramo de negócio. Diante desta situação, todo código fonte desta aplicação será disponibilizado para a comunidade desenvolvedora, sob os termos de licenciamento de software *Microsoft Public License (Ms-PL)* [1]. Para disponibilização do código fonte, foi utilizado o portal *CodePlex* e a URL <http://globaltracker.codeplex.com>.

Outra vantagem da Solução Global Tracker é promover soluções compatíveis com as existentes a um custo reduzido ou zero. Inicialmente seria necessário apenas um dispositivo móvel com as características citadas anteriormente. Um outro fator que favorece a redução de custos com a taxa de manutenção mensal é que o usuário pode escolher quando iniciar o monitoramento e quando interrompê-lo.

Além disso, um recurso como a configuração de um perímetro onde o dispositivo móvel poderia se deslocar, é um diferencial para aqueles que desejam monitorar rotas fixas de deslocamento. Esta funcionalidade será tratada nesta monografia como Cerca Eletrônica.

Em relação especificamente ao Global Tracker Location, uma outra vantagem que pode ser destacada, é que o Xamarin, IDE (*Integrated Development Enviroment*) utilizada para o desenvolvimento da mesma, permite o reaproveitamento de código, caso esta aplicação, precise ser portada para outras plataformas, como IOS ou Windows Phone. [2]

1.5 Organização

O restante desta monografia documenta os passos seguidos para o desenvolvimento do sistema e suas possíveis aplicações.

O capítulo 2 detalha e descreve as ferramentas que foram utilizadas para o desenvolvimento do sistema e algumas de suas características que serão aplicadas. O capítulo 3 descreve detalhadamente o sistema Global Tracker Location e sua implementação, passando pela documentação do mesmo e as questões referentes aos passos que são seguidos para seu desenvolvimento. O capítulo 4 descreve detalhadamente o sistema Global Tracker Web e sua implementação, passando pela documentação do mesmo e as questões referentes aos passos que são seguidos para seu desenvolvimento. O capítulo 5 apresenta uma análise dos resultados que poderão ser obtidos com a implantação da solução Global Tracker em relação a outras soluções do mercado. O capítulo 6 apresenta uma análise de experimentos realizados durante o processo de desenvolvimento e finalizando, o capítulo 7, apresenta um resumo conclusivo desse trabalho e duas propostas de projetos

futuros relacionados que abordam questões relacionadas à melhorias e mudanças de plataforma de desenvolvimento.

Capítulo 2

2 Tecnologias

A principal vantagem da utilização da tecnologia é aumentar a qualidade e diminuir custo. Esta definição também engloba as ferramentas para desenvolvimento de software que a cada dia tornam mais fáceis o trabalho dos desenvolvedores.

Ao se falar em algo que facilite a vida de desenvolvedores de software, pense-se em ferramentas que possuem alto custo de licenciamento, porém, nem sempre é verdade, existem muitas ferramentas gratuitas de qualidade, e são estas que serão utilizadas para o desenvolvimento da Solução Global Tracker.

2.1 .Net Framework

A Plataforma .NET é uma iniciativa da Microsoft que tinha como objetivo uma plataforma única para o desenvolvimento e execução de sistemas. Toda aplicação criada utilizando o código gerado para .Net poderá ser executado por qualquer dispositivo que possua o “.NET Framework”, bem semelhante ao que acontece na plataforma Java, onde o desenvolvedor escreve um código a ser interpretado por uma máquina virtual ao invés de escrever um código para uma plataforma específica.

Tecnicamente o “.NET Framework” é executado sobre uma *CLR (Common Language Runtime)*, que é um ambiente de execução independente da linguagem que interage com várias bibliotecas que fornecem funções do sistema operacional e outras funções implementadas pela Microsoft. O conjunto destas bibliotecas, unificadas, formam o *framework*, que possui uma *CLR* capaz de executar atualmente mais de vinte diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem [3].

Qualquer programa escrito em uma das linguagens de programação suportadas por esta plataforma tem seu código fonte compilado e gera um código intermediário em uma linguagem chamada *MSIL (Microsoft Intermediate Language)*,

Linguagem Intermediária Microsoft. Este novo código fonte gera um arquivo *assembly*, de acordo com o tipo de projeto selecionado pelo desenvolvedor previamente. O arquivo *assembly* é um arquivo escrito em linguagem de montagem e pode ser:

- Um “exe” de arquivos executáveis ou programas;
- Uma “dll” para biblioteca de funções;
- Um “aspx” para páginas web;
- Um “asmx” para web service;
- Um “apk” para Android.

O programa será compilado novamente no momento de sua execução através de outro compilador, o *JIT (Just In Time Compiler)*. No caso do web site, que gera um ou vários arquivos “aspx”, ao ser executado pela primeira vez o *JIT* irá compilar a página e a partir do segundo acesso é utilizada a versão compilada inicialmente [4].

Definindo de uma maneira simples e objetiva o “.NET Framework” é a camada de software na qual são invocadas todas as funções necessárias ao funcionamento dos programas, sob qualquer sistema operacional. Para se executar um programa desenvolvido sobre esta plataforma a única necessidade a ser atendida é que exista uma plataforma .NET para o dispositivo responsável pela execução do mesmo.

2.2 Linguagem C#

A linguagem de programação C# foi criada pela Microsoft em conjunto com o “.NET Framework” e é considerada a mais capaz entre outras linguagens suportadas pela Microsoft, a ponto de ser considerada como linguagem de referência .NET. Isso se deve a algumas razões primordiais e dentre elas pode se citar principalmente o fato desta linguagem ter sido desenvolvida do “zero” para funcionar na nova plataforma, ou seja, sem preocupações de compatibilidade com qualquer código já existente [5].

C# é a linguagem de programação que reflete a plataforma .NET sobre a qual todos os programas desenvolvidos utilizando-a executam. C# está tão acoplado à plataforma .NET que não existe o conceito de *unmanaged code*, código não-gerenciado, suas primitivas de dados são objetos em .NET. Outro recurso conhecido

de outra linguagem, como Java, incorporado é a desalocação automática de memória, mais conhecido como *garbage collector*. Outras das várias abstrações do C# são classes, interfaces, *delegates* e exceções.

C# é uma linguagem que, quando comparada a C e C++, é restrita e melhorada de várias formas:

- Objetos não são liberados explicitamente, mas utilizando um processo de *garbage collector*, que previne a ocorrência de referências inválidas.
- Destrutores não existem e seu equivalente mais próximo é a interface *Disposable*, que é invocado pelo método *Dispose()*, existente na maioria dos objetos e instancias de classes criadas.
- Como em Java, não é permitida herança múltipla, porém uma classe pode implementar várias interfaces abstratas. O objetivo principal é simplificar a implementação do ambiente de execução.
- C# é mais seguro com tipos que C++, por isso revela-se uma linguagem fortemente tipada. As únicas conversões implícitas por padrão são conversões seguras, tais como ampliação de inteiros e conversões de um tipo derivado para um tipo base.
- Qualquer conversão implícita definida pelo usuário deve ser marcada explicitamente, diferentemente dos construtores de cópia de C++.
- Algumas sintaxes são diferentes, e dentre elas podem se destacar a de declaração de vetores que possui alta redigibilidade. Usa-se `int[] a = new int[5]` ao invés de `int a[5]`.
- Propriedades permitem que métodos sejam chamados com a mesma sintaxe de acesso a membros de dados.

A linguagem C# é frequentemente comparada a Java e tida como sua similar, porém existe uma série de diferenças importantes entre ambas:

- Implementação de propriedades e sobrecarga de operadores permitido em C#;
- Java possui exceções checadas, enquanto exceções em C# são não checadas como em C++;
- Java não implementa o `goto` como estrutura de controle, enquanto C# sim;

- Java utiliza-se de comentários Javadoc para gerar documentação automática a partir de arquivos fonte. C# utiliza comentários baseados em XML.

Em C# nenhuma implementação de novas funções e recursos inclui qualquer tipo de bibliotecas de classe. Todas as novas funcionalidades estão diretamente ligadas ao .NET Framework do qual C# obtém suas classes ou funções de execução. O código é organizado em um conjunto de *namespaces*, semelhantes aos pacotes em Java, que agrupam as classes com funções similares. Um exemplo seriam as classe e funções para se trabalhar com gráficos e imagens que se encontram no *namespace System.Drawing*.

Abaixo se encontra um código escrito em C# do famoso “*Hello World!*”, o que destaca mais ainda sua semelhança com Java.

```
public class ClasseExemplo
{
    public static void Main()
    {
        System.Console.WriteLine("Hello World!");
    }
}
```

O ambiente de desenvolvimento recomendado para a programação da linguagem C# é o Visual Studio, desenvolvido pela Microsoft e que atualmente possui uma versão *freeware* conhecida como Visual C# 2005 Express Edition.

2.3 Linguagem ASP.NET

ASP.NET é a plataforma para o desenvolvimento de aplicações web criada pela Microsoft com o objetivo de ser o sucessor da tecnologia ASP e permitir a criação de páginas dinâmicas. Baseado no .NET Framework e herdando todas as suas características, as aplicações desenvolvidas sobre esta plataforma podem ser escritas em algumas linguagens como C# e VB .NET.

Uma aplicação web desenvolvida em ASP.NET pode fazer referência a uma DLL, mesmo que tenha sido escrita em linguagem VB.NET ou consumir um *web services* escritos em C++. Ao contrário da tecnologia ASP, as aplicações ASP.NET são compiladas antes da execução, trazendo sensível ganho de desempenho.

As aplicações web desenvolvidas em ASP.NET necessitam do .NET Framework em conjunto com o *IIS (Internet Information Service)* para executar em um sistema operacional Windows. Para permitir o funcionamento de aplicações ASP.NET e .NET em sistemas operacionais Linux vem sendo desenvolvido um projeto chamado MonoDevelop¹.

O ambiente de desenvolvimento recomendado para a programação do ASP.NET é o Visual Studio, desenvolvido pela Microsoft e que atualmente possui uma versão *freeware* conhecida como Visual Web Developer 2005 Express Edition [6].

2.4 Visual Studio 2010

O Visual Studio 2010 é um ambiente de desenvolvimento leve, simples e integrado, projetado para desenvolvedores experientes ou iniciantes que programam na linguagem C# e desejam criar aplicações desktop utilizando o *template* de projeto windows forms, bem como bibliotecas de classes e aplicativos consoles.

O Visual Studio também oferece versões *freeware*, criada pela Microsoft e oferece muitos recursos de produtividade encontrados na versão completa do Visual Studio 2010. Esta IDE (*Integrated Development Environment*) vem se popularizando devido ao suporte de alto nível fornecido pela Microsoft aos desenvolvedores que o utilizam. Outro fator a ser mencionado são algumas de suas funcionalidades que permitem a criação de *windows forms*, bibliotecas de classe e aplicativos baseados em console usando o *.NET Framework*. A o desenvolvimento de aplicações ASP.NET e Web Services também é contemplada pela ferramenta sem nenhuma dificuldade.

Os recursos de produtividade não foram esquecidos e são facilmente destacados quando se observa alguns pontos como o uso do arrastar e soltar para a criação de formulários e o editor de códigos de alto nível [6].

¹ **MonoDevelop** é uma IDE criada para programação utilizando C# e outras linguagens .NET languages. Mais informações pode ser encontradas em <http://monodevelop.com/>.

2.5 Xamarin

O Xamarin é uma ferramenta de desenvolvimento para criação de aplicativos para dispositivos móveis. Ela é composta por uma API que interage diretamente com a linguagem nativa do dispositivo móvel. Atualmente permite a criação de aplicações para os sistemas operacionais Android e IOS da empresa Apple.

As aplicações criadas utilizando o Xamarin permite o reaproveitamento de código, sendo necessário recriar apenas a interface gráfica.

O Xamarin também oferece um IDE de alto nível, onde o desenvolvedor pode realizar ações como criar uma tela utilizando uma interface gráfica para arrastar e soltar componentes, escrever o código fonte utilizando a linguagem C#, compilar, depurar e fazer o *deploy* da aplicação. [2]

2.6 Subgurim.Net

O Subgurim.Net é uma API disponível para uso em aplicações web ASP.Net, ele interage com o Google Maps sem a utilização de Java Script, bastando apenas arrastar e soltar o controle GMap para página e começar a utilizar o controle. Google Maps

2.7 Google Maps

O Google Maps é um serviço de pesquisa e visualização de mapas gerados a partir de fotos tiradas por satélite. Este serviço gratuito na web, é fornecido pela empresa Google.

Atualmente, o serviço disponibiliza mapas e rotas dos principais países do globo terrestre inclusive do Brasil. As principais cidades do planeta oferecem a possibilidade de um zoom bem definido.

Com uma conta Google é possível destacar rotas, pontos e áreas, além de gerar comentários e compartilhar os links de acesso ao mapa criado [7].

Capítulo 3

3 *Global Tracker Location*

O Global Tracker Location tem como objetivo principal disponibilizar uma interface gráfica na qual o usuário possa determinar as configurações que o dispositivo móvel a ser monitorado utilizará para se comunicar com o Web Service. Além da interface gráfica, este também executa um serviço que é responsável por enviar os dados de coordenadas geográficas para um Web Service que posteriormente irá prover estas informações ao Global Tracker Web. A Figura 2 determina o fluxo de informações entre o Global Tracker Location e o Web Service, que recebe as informações necessárias para o funcionamento pleno da solução Global Tracker.

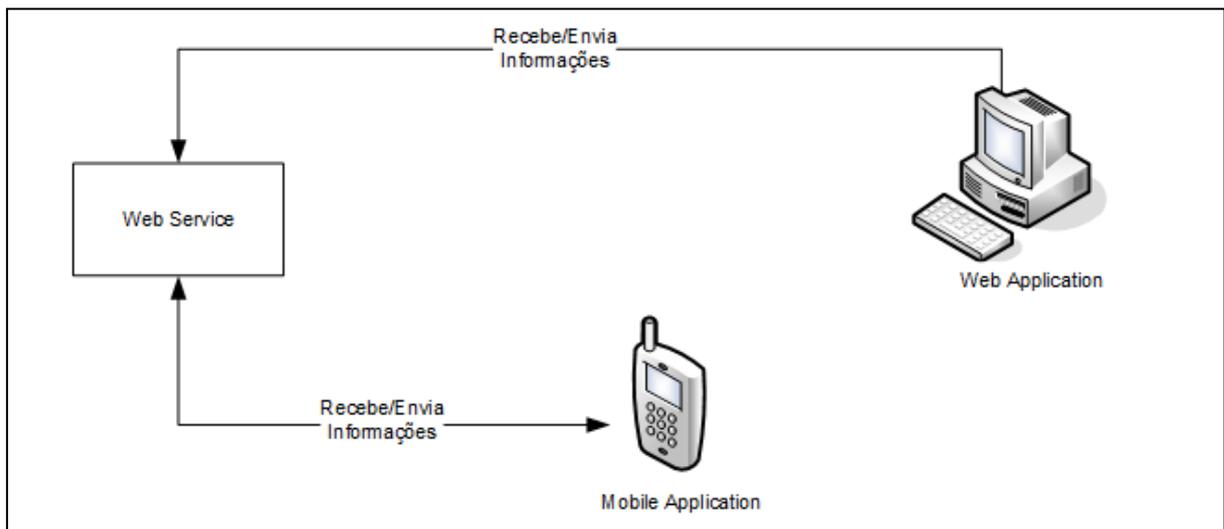


Figura 2 - Diagrama de blocos da solução Global Tracker Location

3.1 Soluções e Trabalhos Relacionados

Desde o início da concepção do Global Tracker Location foram analisados outros softwares para dispositivos móveis que oferecem algum tipo de rastreamento entre as suas funcionalidades. Alguns destes software são:

- Android Lost: É um aplicativo para o sistema operacional Android que permite o proprietário do dispositivo móvel, acessá-lo remotamente e fazer ações como disparar um alarme sonoro, enviar mensagens, apagar dados e descobrir a localização do mesmo. [8]
- GPS Tracker Pro: É um aplicativo para o sistema operacional Android que permite monitorar um dispositivo móvel com o consentimento do proprietário através de um outro dispositivo móvel. Ele também alerta a pessoa monitorada sobre situações adversas que pode estar ocorrendo na região em que ela está, como a ocorrência de uma greve de transportes públicos. [9]
- J2ME e GPS: Artigo técnico da revista Web Mobile [10] que aborda a interface entre J2ME e GPS, bem como fornece um exemplo de implementação.
- Sistema de Alarme e Monitoramento Automotivo com Tecnologia Android: É a monografia apresentada como exigência para obtenção do grau de Bacharel em Ciência da Computação, que apresenta todo o projeto de hardware e software para desenvolvimento do sistema de alarme e monitoramento. [11]

Baseados nestas soluções e trabalhos foram definidas as funcionalidades que fazem parte do Global Tracker Location.

3.2 Princípios de Funcionamento

O usuário que deseja utilizar o Global Tracker Location deverá possuir um conhecimento básico para instalação de aplicações em um dispositivo móvel que utilize o sistema operacional Android 2.3 ou superior, bem como o uso da internet no mesmo.

3.2.1 Configuração do Global Tracker Location

Para utilizar o Global Tracker Location o usuário deverá acessar o endereço www.innovaresw.com.br/globaltracker/website/globaltrackerlocation.zip, para efetuar o download do aplicativo a ser instalado no dispositivo móvel a ser monitorado. Após efetuar o download o usuário deverá descompactar o arquivo, gerando assim um

APK (*Android Package*), que será transferido para o dispositivo através da utilização de um cabo de dados.

Após a instalação da aplicação no dispositivo, o usuário deverá acessar a mesma através do ícone mostrado na figura 3.



Figura 3 – Ícone de Acesso ao aplicativo Android Global Tracker Mobile.

Ao acessar o aplicativo Android Global Tracker Location, o usuário poderá realizar parametrizações para definir o endereço do Web Service, caso o mesmo não deseje utilizar o padrão, que pode ser acessado através da [URL www.innovaresw.com.br/GlobalTracker/WebService/GeneralService.aspx](http://www.innovaresw.com.br/GlobalTracker/WebService/GeneralService.aspx).

Também será definido:

- Nome do Proprietário e Senha de Acesso para que posteriormente, no portal disponibilizado pelo Global Tracker Web, possa ser efetuado o acesso para acompanhar o deslocamento do dispositivo.

- Intervalo de Monitoramento, que corresponde ao tempo em minutos que os dados serão enviados para o banco de dados através do Web Service.

Além destas parametrizações o usuário visualiza o IMEI (International Mobile Equipment Identity) que corresponde a um identificador único universal do dispositivo que está sendo monitorado. Este identificador será utilizado para efetuar o acesso e visualizar o histórico de deslocamento do dispositivo através do Global Tracker Web.

Conforme podemos visualizar na Figura 4, o usuário também inicia e interrompe o serviço de monitoramento através desta tela.



Figura 4 – Tela inicial e de configuração do aplicativo Android Global Tracker Mobile.

3.2.2 Utilização e funcionamento do Global Tracker Location

Após executar todas as parametrizações, a aplicação estará pronta para ser utilizada, neste momento o usuário deverá clicar no botão de **Iniciar Monitoramento** e também ativar o **modo de rede de dados**, que permite a conexão do dispositivo com a internet, para que os dados possam ser enviados para o Web Service. Outra recomendação é que se ative o GPS (Global Positioning System) do dispositivo móvel, para que a precisão das informações de latitude e longitude fiquem mais apuradas.

Após iniciar o monitoramento, conforme o intervalo de tempo determinado, um serviço inicia um *timer* (objeto de evento cíclico que é disparado em um intervalo de tempo) que irá tentar obter os dados correspondente as coordenadas geográficas do dispositivo (especificamente a latitude e longitude), de duas maneiras: utilizando triangulação de antenas [12] ou as coordenadas do GPS. Caso o retorno das coordenadas seja positivo o serviço irá invocar o método `InserirCoordenadasDispositivoMovel` do Web Service.

Caso o usuário deseje interromper o monitoramento, o mesmo deverá abrir a aplicação, caso ela tenha sido fechada e clicar no botão **Parar Monitoramento**.

3.3 Processo Unificado

Para documentar o projeto de desenvolvimento do Global Tracker Location foram criados alguns artefatos do processo unificado, que é um arcabouço de processo que utiliza orientação a objeto na confecção de modelos realizados por ciclos iterativos e incrementais para a construção de sistemas.

Os artefatos criados para o desenvolvimento do Global Tracker Location são Especificação de Requisitos do Sistema, Especificação de Caso de Uso e Especificação do Diagrama de Classe.

O documento de Especificação de Requisitos do Sistema fornece uma descrição detalhada sobre os requisitos funcionais e não funcionais, bem como suas características principais. O Anexo A desta monografia apresenta o SRS (System Requirements Specification) criado para o desenvolvimento do Global Tracker Location.

O documento de Especificação de Caso de Uso representa e descreve em alto nível as funcionalidades e ações executadas por cada usuário ou sistema (atores). O Anexo B desta monografia apresenta a especificação de caso de uso e os que são referentes ao projeto Global Tracker Location encontram-se iniciados pela letra “M”.

O Diagramas de Classe representa termos do domínio do sistema como ideias, objetos, ações e conceitos do mundo real, este é encontrado no fim deste capítulo representado pela figura 6.

3.4 Desenvolvimento

A aplicação Global Tracker Mobile foi desenvolvida na linguagem c#, utilizando o Visual Studio integrado a API Xamarin.Android. A estrutura de arquivos, que referentes a esta aplicação podem ser encontrados na solução Global Tracker, no projeto GlobalTracker.Mobile, que possui o código fonte disponível em <http://globaltracker.codeplex.com/SourceControl/latest>.

Basicamente, um projeto escrito utilizando o Xamarin.Android e linguagem C# utiliza bibliotecas que fazem chamadas diretamente à plataforma nativa do sistema operacional Android e quando se compila o código fonte obtém-se um executável para plataforma nativa que no caso do Android é um arquivo de extensão APK (*Application Package*).

Neste arquivo encontram-se embutidos outras duas aplicações que serão necessárias para o funcionamento aplicativo mobile: o *Mono Shared Runtime* e *Mono Android-10 Support*. Ambos servirão para que as aplicações compiladas tenham acesso a plataforma nativa e também manipule automaticamente alocação de memória e garbage collector. A figura 5 representa a arquitetura envolvida na construção, compilação e execução de qualquer aplicação que utilize o Xamarin. [12]

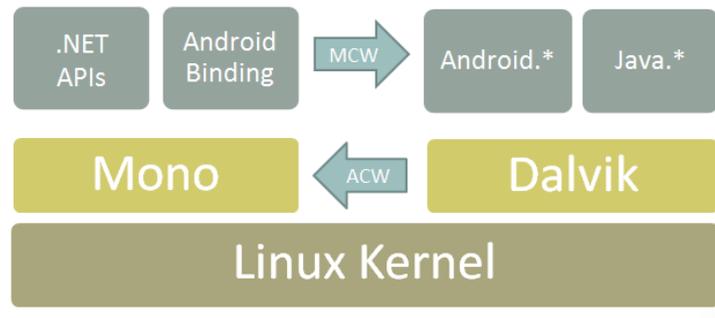


Figura 5 – Arquitetura de compilação e execução de uma aplicação gerada com Xamarin.Android.

Explicando resumidamente a arquitetura apresentada na figura 5 temos: a .Net APIs, que fornece métodos na linguagem C#, por exemplo, acessar o GPS do dispositivo móvel. Este mesmo método, que se encontra acessível via código C#, quando compilado, consegue acesso as bibliotecas do Android ou Java por intermédio do MCW (*Managed Callable Wrappers*), que é a ponte acessada todo o tempo para que o código gerenciado invoque as funções nativas das bibliotecas que estão sobre a máquina virtual de Dalvik [13]. Já o processo inverso, ou seja, a resposta da invocação feita anteriormente, se dá através do ACW (*Android Callable Wrappers*), que estabelece a ponte para a plataforma Mono ou como citada anteriormente Xamarin.Android. Tudo isso rodando sobre o Linux Kernel.

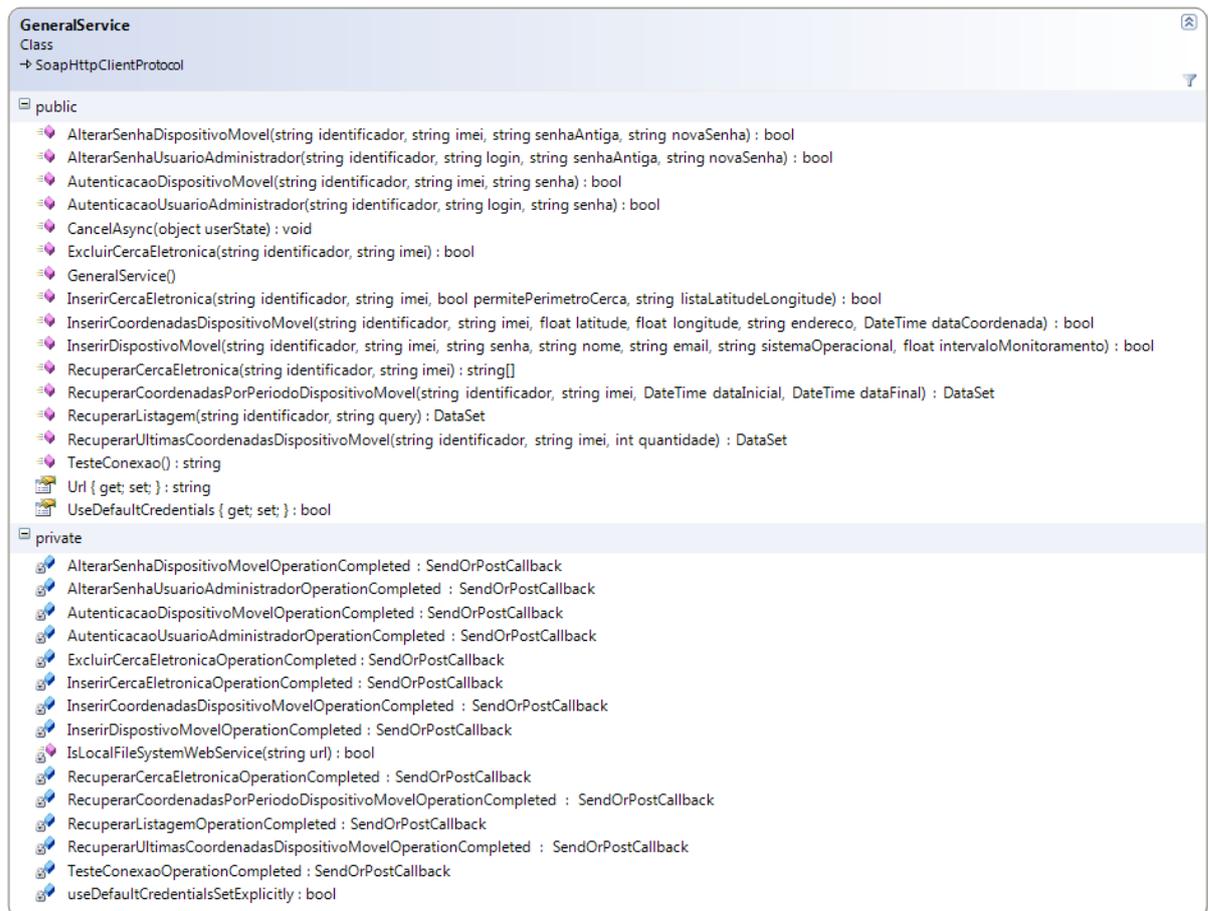
A implementação do Global Tracker Mobile está baseada na classe *Activity*. Dentro do conceito de projetos do tipo Android Application, pertencente ao grupo Mono for Android, no Visual Studio, as *Activities* possuem um ciclo iniciado no evento *OnCreate*, de forma que, este evento é responsável por inicializar e atribuir eventos aos controles que foram incluídos em uma tela. Após a execução do *OnCreate* a tela é carregada no dispositivo móvel e visualizada pelo usuário do sistema.

No projeto Global Tracker Location, destacaremos basicamente duas *Activities*, a *ActivitySplashScreen* que é responsável pela tela inicial da aplicação mobile e a *ActivityFormMain* que contém toda regra de negócio necessária para funcionamento da aplicação e envio das coordenadas para o banco de dados.

Outro componente de código importante neste projeto, é a adição de referência ao Web Service, que é gerado a partir do projeto GlobalTracker.WebService da

solução Global Tracker. Esta referência da origem a classe GeneralService, que é imprescindível para que a aplicação mobile tenha comunicação com o banco de dados em que irá armazenar os dados das coordenadas.

No diagrama abaixo podemos visualizar as classes de destaque do projeto GlobalTracker.Mobile.



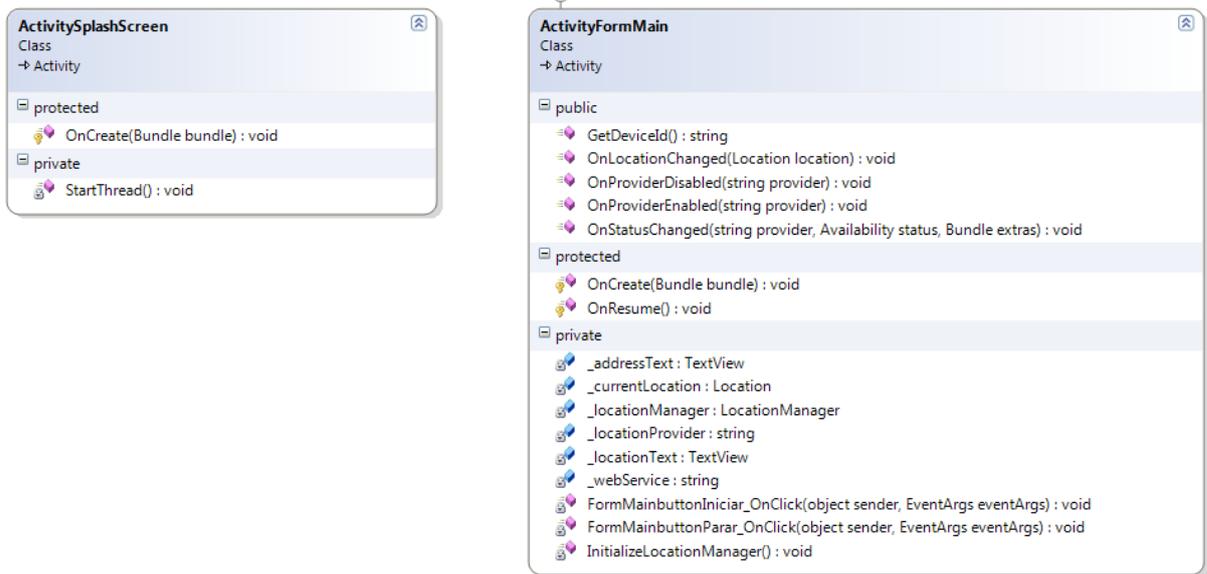


Figura 6 – Diagrama das principais classes do projeto *GlobalTracker.Mobile*.

As informações técnicas sobre a implementação destas classes encontram-se no Anexo C, Código Fonte Comentado do Global Tracker Mobile.

Capítulo 4

4 Global Tracker Web

O Global Tracker Web tem como principal funcionalidade disponibilizar uma interface gráfica para o usuário que deseja rastrear ou monitorar um dispositivo móvel, interagindo diretamente com o web service que também é utilizado pelo Global Tracker Location para persistir as informações em um banco de dados, referentes ao deslocamento realizado pelo dispositivo móvel. Além do serviço de rastreamento o Global Tracker Web oferece outros recursos como, cerca eletrônica e visualização de alertas. A Figura 7 determina o fluxo de informações entre o Global Tracker Web e Web Service, que provê as informações necessárias para todas as funcionalidade da aplicação. Ainda nesta figura é possível observar a comunicação da aplicação web com o Google Maps.

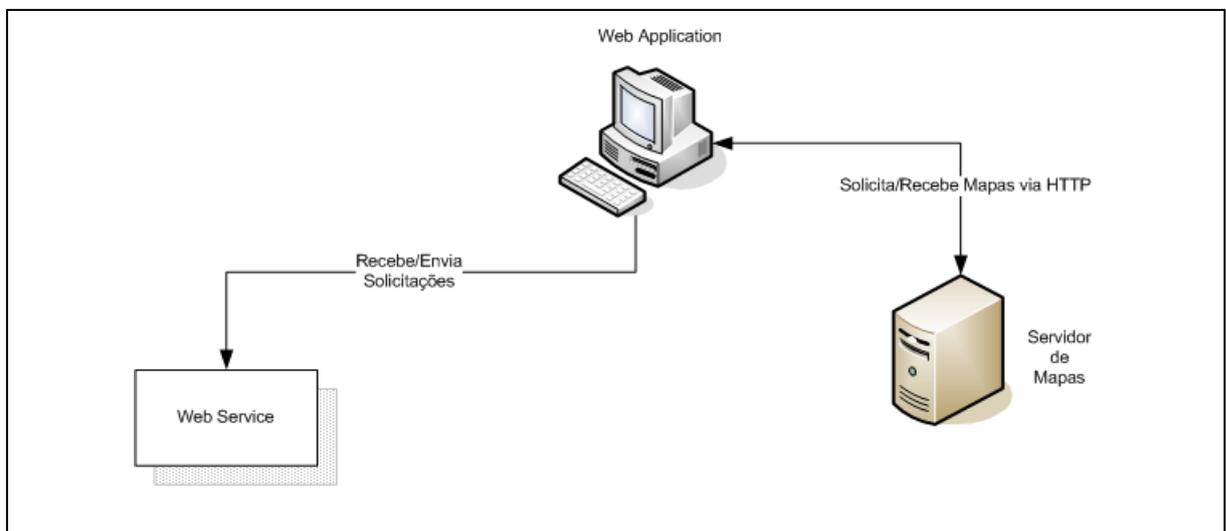


Figura 7 - Diagrama de blocos da solução Global Tracker Web

4.1 Princípios de Funcionamento

O usuário que deseja utilizar o Global Tracker Web deverá possuir um conhecimento básico para o uso da internet, bem como possuir um sistema operacional com um navegador de internet.

4.1.1 Acesso ao Global Tracker Web

Para utilizar o Global Tracker Web o usuário acessa o endereço <http://www.innovaresw.com.br/GlobalTracker/WebSite/index.aspx>, utilizando um navegador de internet. O portal acessado contém informações sobre os demais módulos que compõem a solução Global Tracker e fornece através do menu principal acesso a Área Restrita.



Figura 8 – Página inicial do portal Global Tracker.

Ao acessar a Área Restrita o usuário visualiza uma interface para informar o IMEI ou *login* de Administrador, bem como a senha de acesso. Caso informe o IMEI correspondente a um dispositivo móvel e a senha que foi definida para o mesmo durante a configuração do Global Tracker Mobile o usuário terá acesso apenas a

informações deste dispositivo. Ao informar um *login* de Administrador, bem como sua senha, o usuário terá acesso a todos dispositivos cadastrados na base de dados e poderá escolher qual deseja consultar.

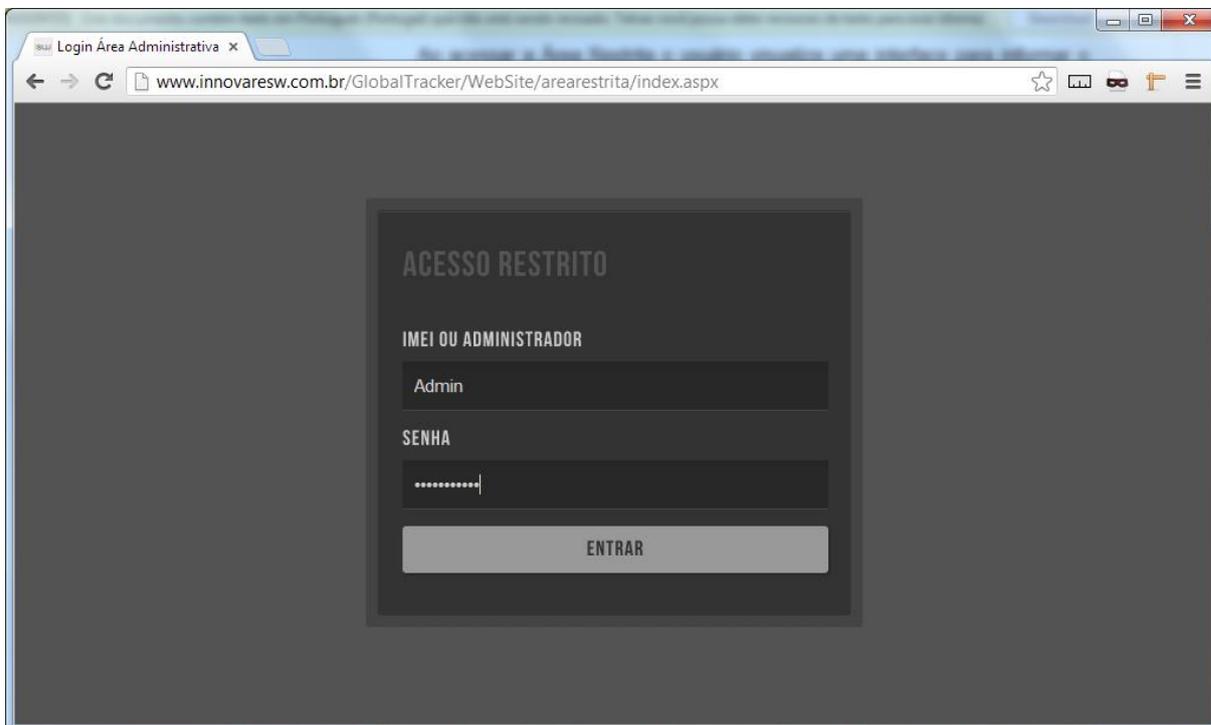


Figura 9 – Página para informar login de acesso ao sistema Global Tracker Web.

Após acessar a Área Restrita o usuário é direcionado a página principal do Global Tracker Web, onde ele pode decidir qual ação irá realizar através de um menu que se encontra à esquerda. Dentre as ações possíveis estão:

- Listar os dispositivos móveis;
- Acompanhar o histórico de navegação;
- Cadastrar uma cerca eletrônica para um dispositivo;
- Listar históricos de alertas para um dispositivo.

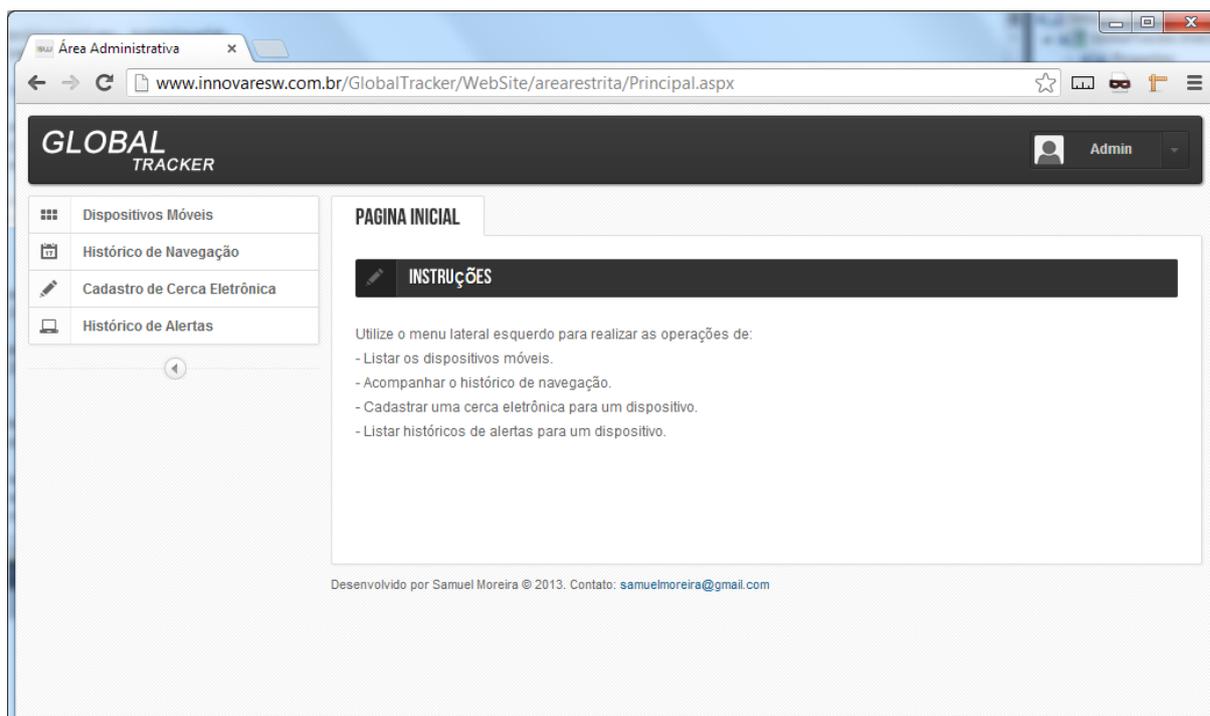


Figura 10 – Página principal do Global Tracker Web.

Caso o usuário selecione a opção de Dispositivos Móveis, ele irá visualizar a lista dos dispositivos cadastrados conforme as permissões do usuário que acessou o Global Tracker Web utilizado para acessar o sistema. Esta lista será exibida em formato de tabela/grid que permite a filtragem dos dados conforme as colunas:

- Código: código interno de auto numeração do banco de dados;
- IMEI: código universal do dispositivo móvel cadastrado;
- Senha: senha de acesso a Área Restrita;
- Nome: nome do proprietário do dispositivo móvel;
- E-Mail: e-mail de contato do proprietário do dispositivo móvel;
- Sistema Operacional: versão do sistema operacional Android do dispositivo móvel;
- Intervalo: tempo de envio das informações de coordenadas

The screenshot shows a web browser window with the URL www.innovaresw.com.br/GlobalTracker/WebSite/arearestrita/dispositivosmouveis.aspx. The page title is "GLOBAL TRACKER" and the user is logged in as "Admin". On the left, there is a navigation menu with options: "Dispositivos Móveis", "Histórico de Navegação", "Cadastro de Cerca Eletrônica", and "Histórico de Alertas". The main content area is titled "DISPOSITIVOS MÓVEIS" and contains a "LISTAGEM" table with the following data:

Código	IMEI	Senha	Nome	E-Mail	Sistema Operacional	Intervalo
1	123	S123	Samuel Moreira	samuelpmoreira@gmail.com	Android	5
2	12345	S12345	Teste Moreira	testemoreira@gmail.com	Android	5
3	352184051306187	123	Samuel Moreira	samuelpmoreira@gmail.com	Android	5

At the bottom of the page, it says "Desenvolvido por Samuel Moreira © 2013. Contato: samuelpmoreira@gmail.com".

Figura 11 – Listagem de dispositivos móveis que o usuário corrente tem acesso

Ao selecionar a opção Histórico de Navegação o usuário que efetuou acesso ao Global Tracker Web é direcionado para a página que exibe em um *combo box* os dispositivos que podem ser monitorados (Selecione o Dispositivo Móvel) e em outro *combo box* as informações referentes às coletas que podem ser consultadas (Selecione Coletas Que Deseja Visualizar).

Esta página é sempre carregada com o primeiro dispositivo da lista e com coletas a visualizar definido como Online, isso faz com que seja sempre mostrado a última coleta do dispositivo com data e hora, conforme pode ser visto na figura 12.

A visualização de coletas como Online, também faz com que a página seja carregada automaticamente a cada minuto. Além da opção de exibição Online o usuário pode escolher 5 Últimas Coletas, 10 Últimas Coletas, 25 Últimas Coletas, 50 Últimas Coletas, 100 Últimas Coletas e 500 Últimas Coletas.

Caso seja selecionada, por exemplo a opção corresponde às 5 Últimas Coletas, será exibida a rota feita correspondente às mesmas, conforme pode ser visto na figura 13.

The screenshot shows the 'GLOBAL TRACKER' web interface. The browser address bar displays 'www.innovaresw.com.br/GlobalTracker/WebSite/arearestrita/historiconavegacao.aspx'. The page title is 'GLOBAL TRACKER' and the user is logged in as 'Admin'. On the left sidebar, there are menu items: 'Dispositivos Móveis', 'Histórico de Navegação', 'Cadastro de Cerca Eletrônica', and 'Histórico de Alertas'. The main content area is titled 'HISTÓRICO DE NAVEGAÇÃO' and contains a 'MAPA' section. Below the map title, there are two dropdown menus: 'Selecione o Dispositivo Móvel' (set to '352184051306187 - Samuel Moreira') and 'Selecione Coletas Que Deseja Visualizar' (set to 'Online'). The map shows a street view of a city area with a red pin and a pop-up window displaying the date and time '6/14/2013 10:49:33 AM'. The map includes labels for streets like 'R. Calambau', 'R. do Ouro', 'R. Divinópolis', and 'Av. Domínio Luna'. The Google logo and 'Dados cartográficos ©2013 Google, MapLink - Termos de Uso' are visible at the bottom of the map.

Figura 12 – Histórico de Navegação do Dispositivo utilizando a opção Online.

The screenshot shows the same 'GLOBAL TRACKER' web interface. The browser address bar and page title are identical to Figure 12. The user is still logged in as 'Admin'. The 'HISTÓRICO DE NAVEGAÇÃO' section now shows the 'Selecione Coletas Que Deseja Visualizar' dropdown menu set to '5 Últimas'. The map displays a wider area of the city, including labels for 'Av. Miguel Augusto Gonçalves', 'Praça Juvenal Pereira', 'Praça Luiz Ribeiro', 'Av. São João', 'Praça Donina R Alves', 'Estádio Alcides Pio', 'Câmara Municipal de Itaúna', 'Praça de Esportes JK', 'Prefeitura Municipal de Itaúna', 'Praça Monsenhor Hilton G Souza', 'Paróquia Ns. Aparecida', 'Av. Domínio Luna', and 'R. Divinópolis'. The Google logo and 'Dados cartográficos ©2013 Google, MapLink - Termos de Uso' are visible at the bottom of the map.

Figura 13 – Histórico de Navegação do Dispositivo utilizando a opção 5 Últimas.

Ao selecionar a opção Cadastro de Cerca Eletrônica o usuário que acessou o Global Tracker Web é direcionado para a página que exibe em um *combo box* os dispositivos que podem ter suas cercas eletrônicas definidas.

Após o definir o dispositivo que terá sua cerca eletrônica definida, o usuário deverá se deslocar no mapa até a área que deseja delimitar. Ao localizar a área o usuário deverá ativar a marcação da área de monitoramento através do *check box* abaixo do mapa.

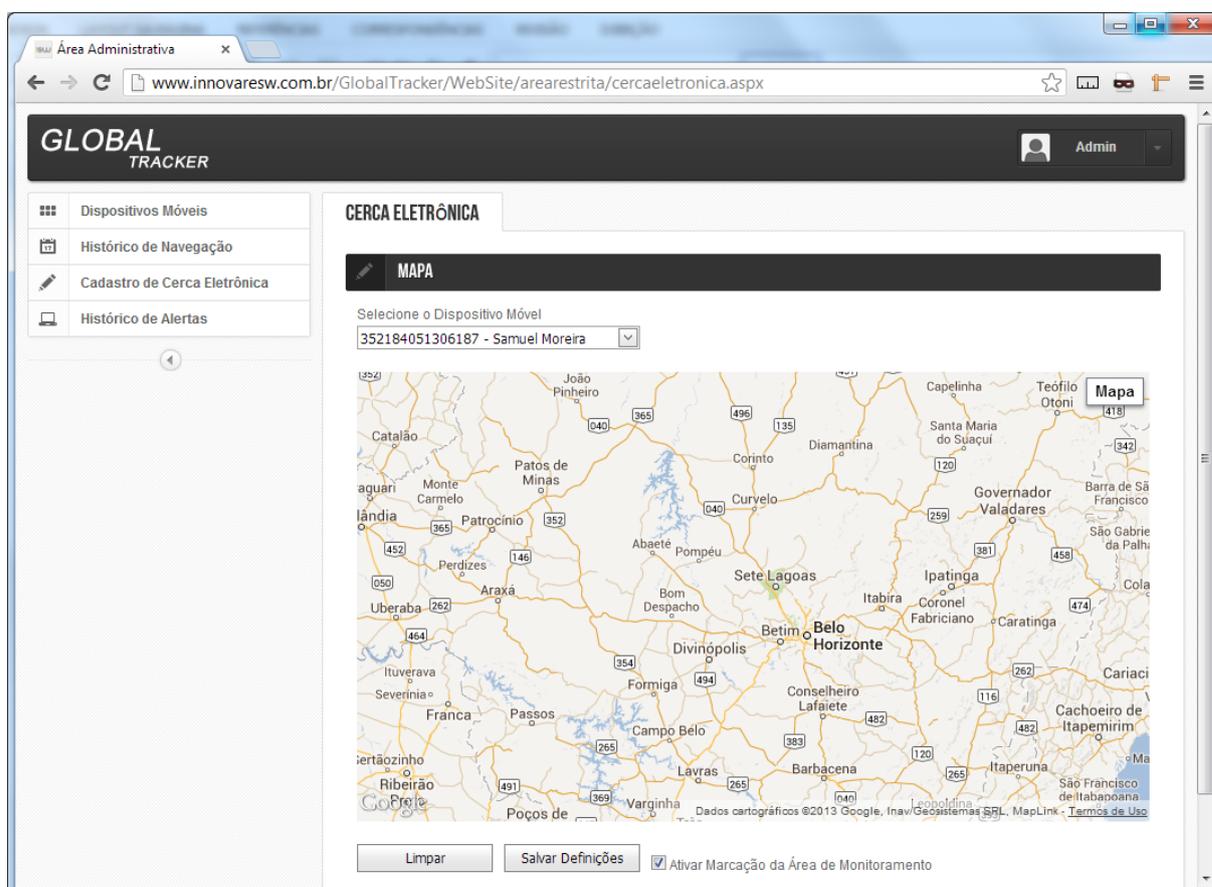


Figura 14 – Definição da cerca eletrônica.

Ao acionar a ativação o usuário deverá clicar nos pontos seguindo uma ordem lógica para marcação dos mesmos conforme é mostrado na figura 15. O perímetro da área de monitoramento começa ser plotado no mapa somente após a definição do terceiro ponto.

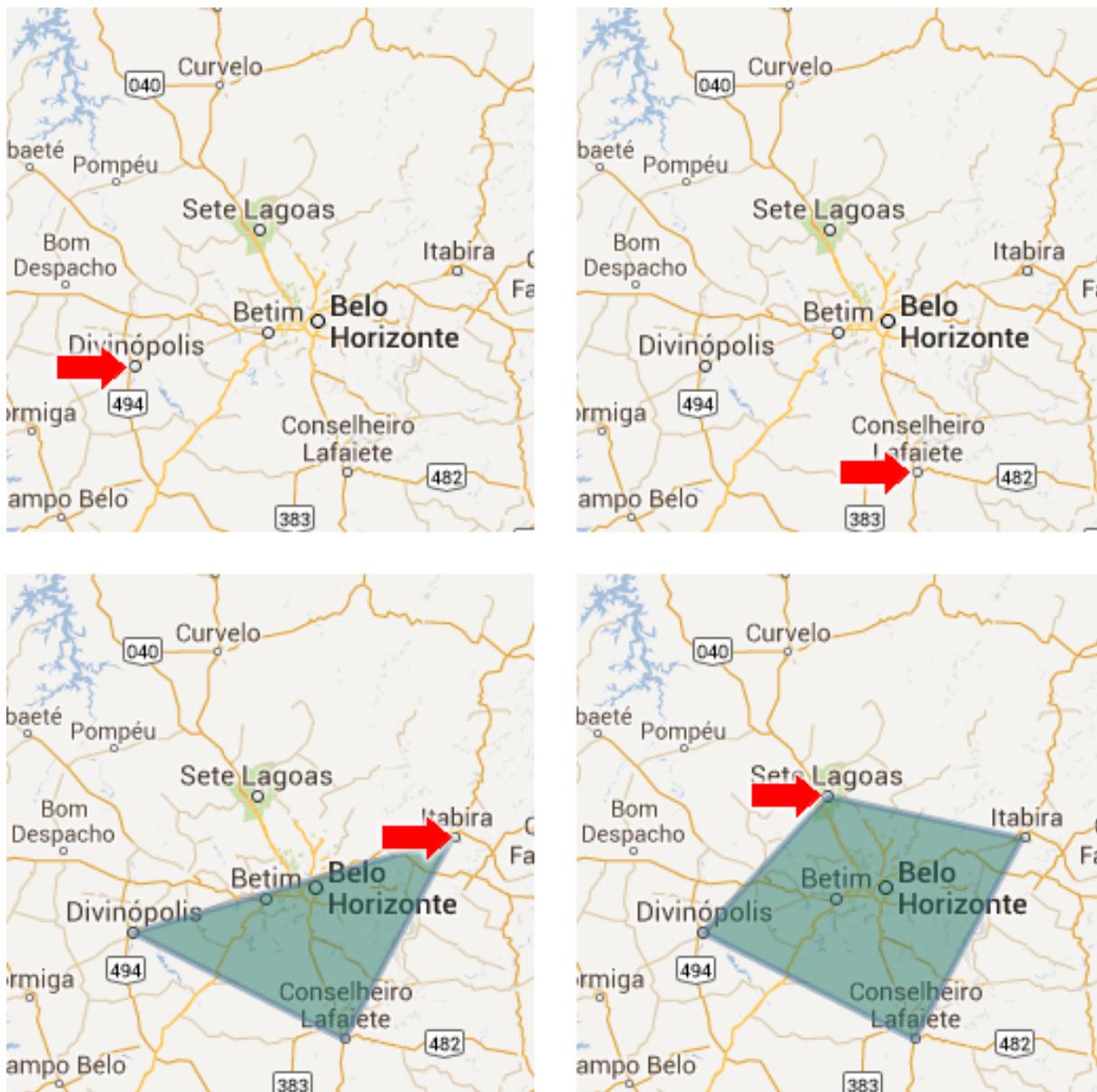


Figura 15 – Marcação de Cerca Eletrônica: 1º clique em Divinópolis, 2º clique em Cons. Lafaiete, 3º clique em Itabira, 4º clique em Sete Lagoas.

Após definir os pontos para salvar basta clicar no botão Salvar Definições. Para apagar a definição de uma cerca eletrônica, basta clicar no botão Limpar e em seguida Salvar as definições novamente.

Caso o usuário queira visualizar o histórico de alertas exibidos quando um dispositivo móvel ultrapassar uma cerca eletrônica determinada, pode-se selecionar esta opção no menu lateral e acessá-la através de uma tabela/grid e filtrar as informações utilizando os campos:

- IMEI: código universal do dispositivo móvel cadastrado;
- Data: data e hora da ocorrência;
- Latitude: coordenada geográfica;
- Longitude: coordenada geográfica.

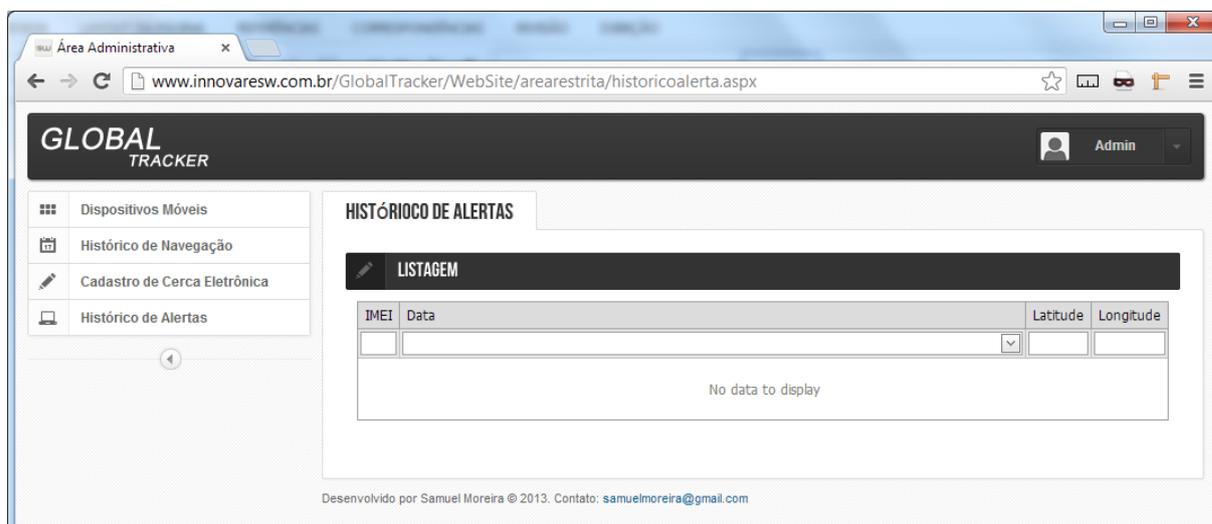


Figura 16 – Página para visualizar histórico de alertas.

4.2 Processo Unificado

Para documentar o projeto de desenvolvimento do Global Tracker Web também foram criados alguns artefatos do processo unificado que se encontram referenciados no Anexo A.

O documento de Especificação de Caso de Uso o Anexo B desta monografia apresenta a especificação de caso de uso e os que são referentes ao projeto Global Tracker Web e encontram-se iniciados pela letra “W”.

O Diagrama de Classe representa termos do domínio do sistema como ideias, objetos, ações e conceitos do mundo real, este é encontrado no fim deste capítulo representado pela figura 17.

4.3 Desenvolvimento

A aplicação Global Tracker Web foi desenvolvida na linguagem C#, utilizando o Visual Studio e é composta por um projeto do tipo Web Service e por outro do tipo Web Application utilizando ASP.Net. A estrutura de arquivos referentes a esta aplicação podem ser encontrados na solução Global Tracker, nos projetos GlobalTracker.WebSite e GlobalTracker.WebService, que possui o código fonte disponível em <http://globaltracker.codeplex.com/SourceControl/latest>.

4.3.1 Global Tracker – Web Service

O projeto GlobalTracker.WebService é responsável por prover acesso a um banco de dados Microsoft SQL Server, que é utilizado como repositório de todas as informações necessárias para o pleno funcionamento da solução Global Tracker. Os scripts para criação banco de dados encontram-se disponíveis no Anexo F.

A implementação do Web Service tem como objetivo disponibilizar a troca de informações ou execução de procedimentos por inúmeras aplicações ou serviços de plataforma web ou não, através do protocolo SOAP [14]. Neste projeto foram criadas basicamente três classes:

- DatabaseAccess: contém os métodos necessários para estabelecer a conexão com um banco de dados Microsoft SQL Server. Inserir, atualizar, excluir e selecionar informações neste mesmo banco.
- GeneralService: contém a interface que é consumida pelos sistemas que utilizam este Web Service como referência.
- EMail: utilizada para enviar os e-mails de alerta caso um dispositivo móvel ultrapasse o limite da cerca eletrônica.

No diagrama da figura 17, podemos visualizar as classes de destaque do projeto GlobalTracker.WebService.

As informações técnicas sobre a implementação destas classes encontram-se no Anexo D, Código Fonte Comentado do Global Tracker Web Service.

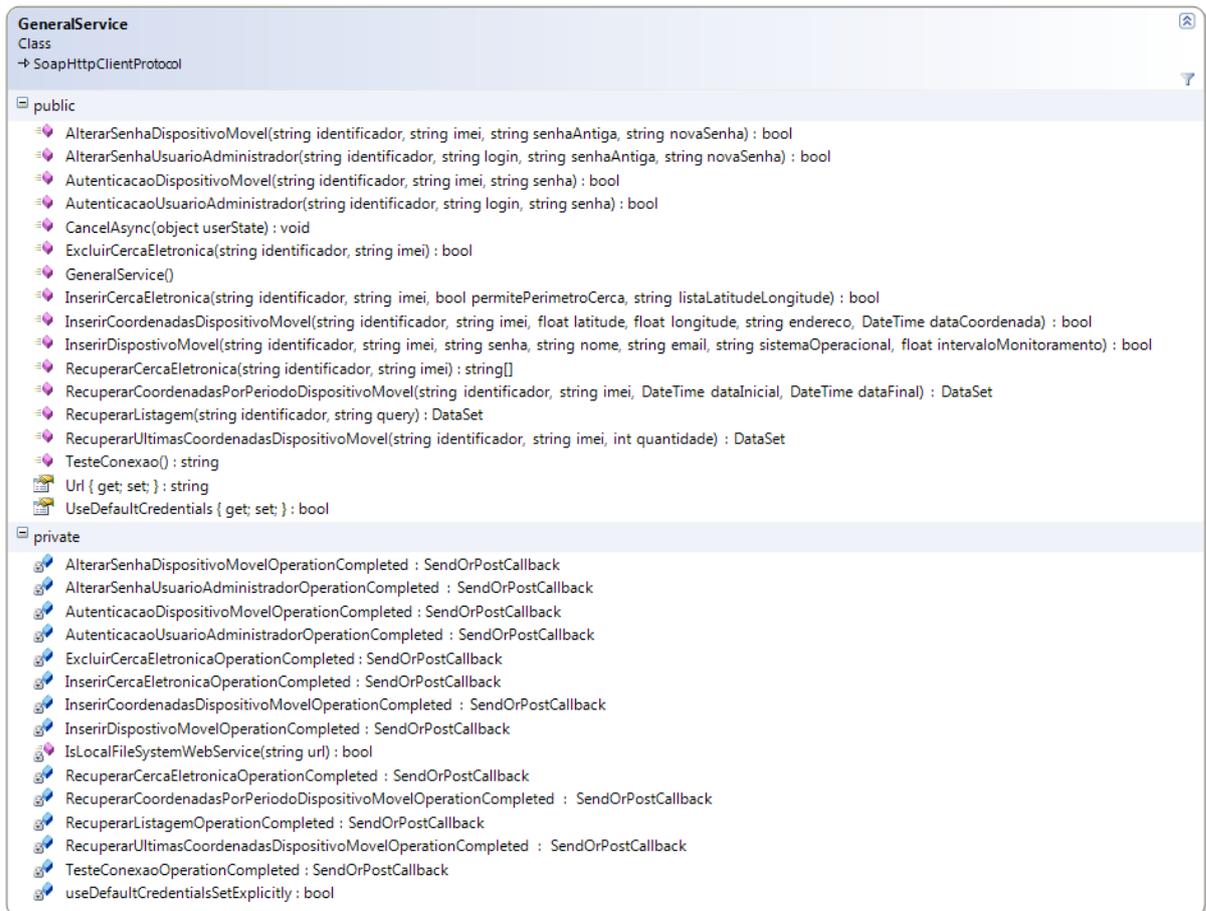


Figura 17 – Diagrama das principais classes do projeto GlobalTracker.WebService.

4.3.2 Global Tracker – Web Site

A aplicação web escrita em linguagem C# utilizando ASP.Net é responsável por prover a interface de acesso de quem deseja fazer o monitoramento de um dispositivo móvel.

A implementação deste projeto consiste na criação de páginas web que executam ações assíncronas ou do lado do servidor. As regras de negócio encontram-se fortemente acopladas as páginas conforme sua funcionalidade, sendo assim cada página possui uma classe com suas implementações.

Um componente de grande importância neste projeto é o Subgurim.Net, que é responsável por fornecer o objeto GMap, utilizado em todas as situações que é necessário utilização de um mapa.

Neste projeto podemos destacar 7 classes:

- DispositivosMoveis: listar os dispositivos móveis cadastrados na base de dados provida pelo Web Service.
- HistoricoAlerta: listar os alertas de dispositivos móveis que ultrapassaram a cerca eletrônica.
- Index: classe utilizada para autenticação na área restrita.
- Security: classe para segurança de acesso da aplicação.
- CercaEletronica: permite a criação de perímetro sobre o mapa para o dispositivo selecionado.
- HistoricoNavegacao: exibe o histórico de navegação do dispositivo conforme o filtro determinado.
- GeneralService: classe de utilização do Web Service.

No diagrama abaixo podemos visualizar as classes de destaque do projeto GlobalTracker.WebSite.

As informações técnicas sobre a implementação destas classes encontram-se no Anexo E, Código Fonte Comentado do Global Tracker Web Site.

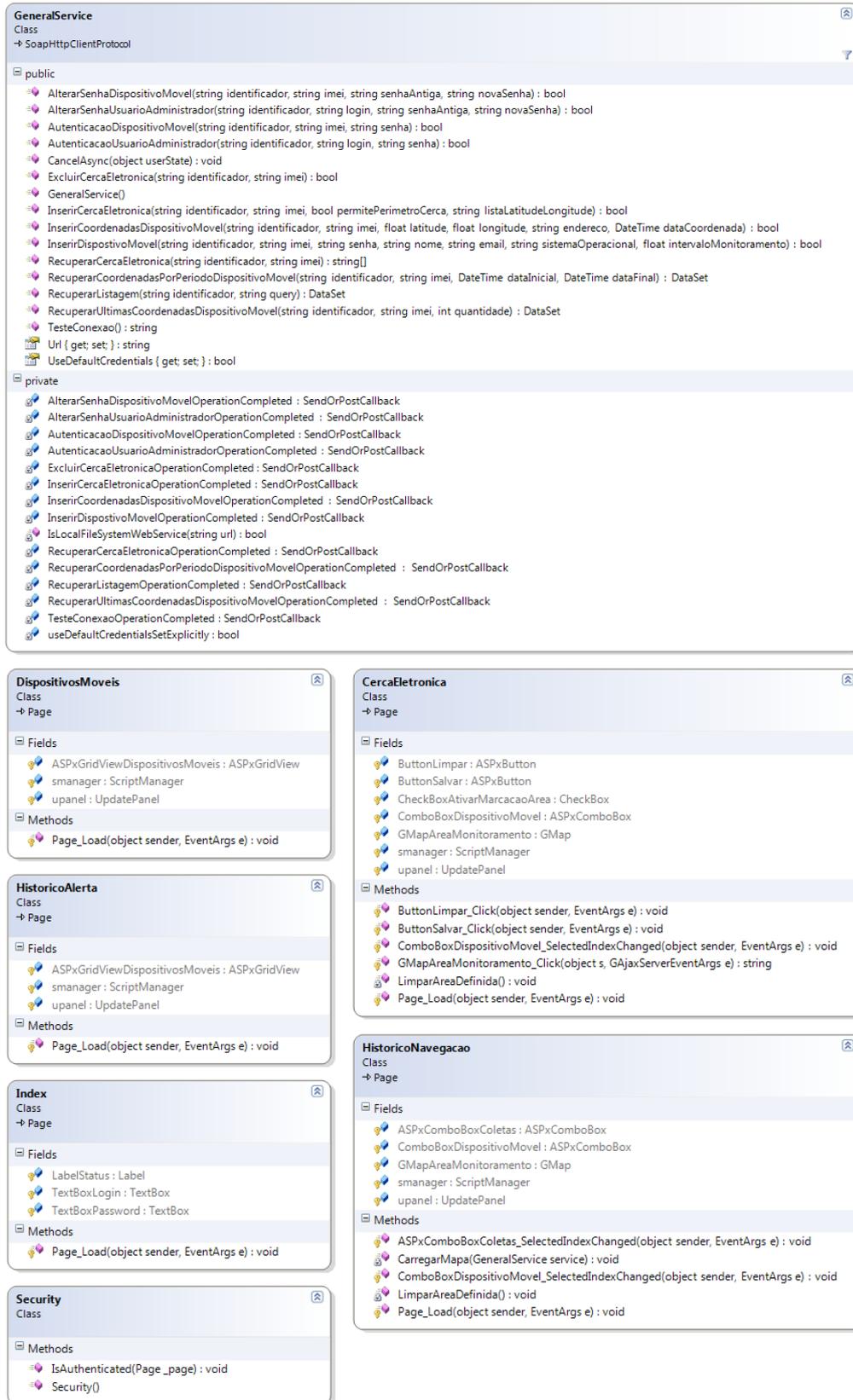


Figura 18 – Diagrama das principais classes do projeto *GlobalTracker.WebService*.

Capítulo 5

5 Estudos de Aplicação

Para realizar a implantação da solução Global Tracker, inicialmente é necessário uma infraestrutura que a maioria das pequenas e médias empresas possui, que é um servidor de internet.

Este servidor deverá utilizar uma versão do sistema operacional Windows que permita a instalação dos recursos referentes ao IIS (Internet Information Service), que é um serviço e aplicativo utilizado para hospedagem de aplicações web em geral que utilizam linguagem de programação ASP, ASP.NET, dentre outras.

Outro requisito de infraestrutura necessário para implantação da solução, é gerenciador de banco de dados relacional Microsoft SQL Server, que neste caso pode ser utilizado na versão Express 2008 ou superior. A versão Express desta aplicação é totalmente gratuita, limitada apenas a utilização de uma base de dados de tamanho máximo de 4 gigabytes.

Caso o usuário ou empresa não possua a infraestrutura necessária, ainda assim a Solução poderá ser implantada em um servidor de hospedagem de web sites que forneçam estes recursos por um valor que pode variar de U\$10,00 a U\$15,00 mensais.

Quanto aos ramos de atividade em que a solução pode ser aplicada, é bastante variado, passando pelo rastreamento de veículos, cargas e pessoas. Em situações onde um smartphone ou tablet for considerado um equipamento inadequado ou frágil, também poderia se considerar a criação de um hardware blindado utilizando a arquitetura do Arduíno [15], que atualmente permite a programação do hardware utilizando Android através de um kit de desenvolvimento físico de aparelhos Android, baseados em placas Arduíno lançado pela Google.

Capítulo 6

6 Análise Experimental

Durante o processo de implementação da solução Global Tracker foram realizados alguns experimentos práticos até se obter uma versão funcional.

Na fase de desenvolvimento do Web Service, utilizado para fazer interface entre Global Tracker Mobile e Global Tracker Web, não houve dificuldades significativas, pois a base de conhecimento sobre este assunto é ampla na internet.

Já na fase de desenvolvimento do Web Site, que é utilizado para exibição das informações referente ao posicionamento do dispositivo móvel através de um mapa, uma questão em que houve uma demanda de tempo maior foi a definição de qual componente seria utilizado para exibição do mapa. Havia como opções utilizar diretamente a API do Google Maps ou pesquisar a existência de algum componente ASP.Net que também oferece os recursos de delimitar um perímetro, ligar pontos de um coordenadas em um mapa, dentre outras necessidades. Considerando que utilizar a API do Google Maps implicaria em criar um subprojeto para realizar todas estas interações utilizando *Java Script*, foi adotada a segunda opção e utilizado o componente Subgurim.Net para Google Maps que facilitou bastante a implementação.

Quanto a análise experimental realizada sobre o Global Tracker Mobile, após definir que o desenvolvimento seria realizado utilizando o Xamarin.Android, pode-se observar questões mais práticas relacionadas à aplicação, como por exemplo a precisão que era obtida ao se utilizar o GPS para obter as coordenadas geográficas ao invés da rede de telefonia celular. Em casos onde se utiliza o GPS e se tem condições climáticas ideais a precisão das coordenadas obtidas chega a cerca de 2 a 3 metros, enquanto nas mesmas condições, utilizando a rede de telefonia celular, a precisão variava de 200 a 1000 metros.

Na figura 19 existe uma comparação entre duas imagens, a primeira demonstra uma coordenada utilizando a rede de telefonia celular e a segunda uma coordenada

obtida com o GPS e, em ambas o dispositivo encontrava-se fisicamente no mesmo local, exatamente no ponto indicado pelo GPS.

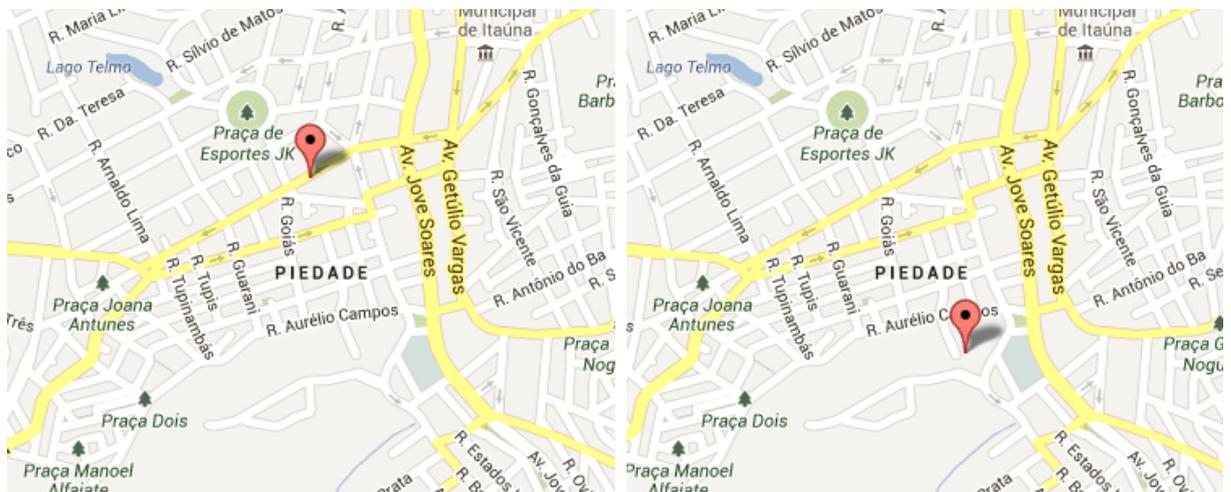


Figura 19 – Comparação de coordenadas obtidas pela rede de telefonia celular e GPS.

Capítulo 7

7 Conclusões

Durante a fase de estudos, determinar quais tecnologias seriam utilizadas para o desenvolvimento do Global Tracker Web foi crítico. Porém, na fase de levantamento de requisitos, observou-se que a utilização de várias tecnologias em conjunto facilitaria a implementação da solução Global Tracker.

Os objetivos iniciais propostos por este trabalho, como exibir o ponto onde um dispositivo móvel se encontra, acompanhamento do histórico, criação de cercas eletrônicas e exibição de alertas foram alcançados, conforme pode ser observado em simulações realizadas utilizando alguns protótipos. Porém algumas funcionalidades que seriam propostas como, traçar rotas foram consideradas difíceis de se implementar e apresentam como uma sugestão para um trabalho futuro, além disso, foram observados alguns pontos que podem melhorar a experiência do usuário e enriquecer os recursos oferecidos pelas aplicações.

No Global Tracker Mobile pode ser implementado uma funcionalidade para que nos momentos de indisponibilidade de conexão com a internet, as coletas de coordenadas sejam realizada e armazenas em um banco de dados local no dispositivo, para que posteriormente elas possam ser enviadas até o banco de dados principal através do web service. Pode-se implementar também uma funcionalidade para que os alertas de violação da cerca eletrônica sejam emitidos no dispositivo móvel, através da área de notificação do sistema operacional.

Em relação ao Global Tracker Web a criação de relatórios analíticos e sintéticos pode ser algo considerável para monitoramento de uma grande quantidade de dispositivos móveis.

Bibliografia

- [1] Open Source Initiative.
Título da Página: Microsoft Public License (MS-PL) | Open Source Initiative, Junho 2013.
Disponível em <http://opensource.org/licenses/MS-PL/>.

- [2] Xamarin.Studio
Título da Página: Tour of Creating Cross-platform iOS, Android, and Windows Mobile Apps in C# - Xamarin, Junho 2013.
Disponível em <http://xamarin.com/tour/>.

- [3] Autores: Edwin Lima, Eugênio Reis.
Livro: C# e .Net – Guia do Desenvolvedor, página 8.
ISBN 85-352-0954-9, Editora Campus Ltda, 2002.

- [4] Autores: Edwin Lima, Eugênio Reis.
Livro: C# e .Net – Guia do Desenvolvedor, página 13.
ISBN 85-352-0954-9, Editora Campus Ltda, 2002.

- [5] Autores: Edwin Lima, Eugênio Reis.
Livro: C# e .Net – Guia do Desenvolvedor, página 3.
ISBN 85-352-0954-9, Editora Campus Ltda, 2002

- [6] Autores: Mickey Gousset, Brian Keller, Ajoy Krishnamoorthy e Martin Woodward
Livro: Professional Application Lifecycle Management with Visual Studio 2010, chapter 6.
ISBN 978-0-470-48426-5, Wiley Publishung, Inc.

- [7] Google Inc.
Título da Página: API do Google Maps - Google Developers, Junho 2013.
Disponível em <https://developers.google.com/maps/?hl=pt-br>.

- [8] Android Lost
Título da Página: Android Lost, Junho 2013.
Disponível em <http://www.androidlost.com/#getstarted>.
- [9] Google Play
Título da Página: GPS Tracker Pro, Março 2012.
Disponível em
<https://market.android.com/details?id=com.maplogix.app.gpstrackerpro>.
- [10] Autores: Elias F. Lopes, Jaime G. Nogueira, Thiego G. Nacif, and Thienne M. Johnson.
Artigo: Sistemas baseados em localização em um mundo sem-fio com J2ME e API location. 3o SBSI - Simpósio Brasileiro de Sistemas de Informação, 2006.
- [11] Autores: Gabriel C. Santos, Rodrigo R. Rocha, Wesley C. Lima
Orientador: Ivair Teixeira
Monografia: Sistema de Alarme e Monitoramento Automotivo com Tecnologia Android - Ciência da Computação - Faculdade Anhanguera de Valinhos
- [12] Xamarin Studio
Título da Página: Architecture | Xamarin, Junho 2013.
Disponível em
http://docs.xamarin.com/guides/android/advanced_topics/architecture/.
- [13] Xamarin Studio
Título da Página: Android Callable Wrappers | Xamarin, Junho 2013.
Disponível em
http://docs.xamarin.com/guides/android/advanced_topics/java_integration_over_view/android_callable_wrappers/.
- [14] Microsoft Corporation
Título da Página: Web Services, Junho 2013.
Disponível em <http://msdn.microsoft.com/pt-br/library/cc564893.aspx>.

[15] O'Reilly Media, Inc.

Título da Página: Google I/O 2011: 5 things to keep watching - Reilly Radar

Disponível em <http://radar.oreilly.com/2011/05/google-io-2011-5-things-to-watch.html>.

Anexos

Anexo A – SRS (*System Requirement Specification*)

Abaixo encontra-se a especificação de requisitos da solução Global Tracker preparada por Samuel Moreira Marques, revisada junto ao orientador Samuel E. L. de Oliveira.

Introdução

1.1. Propósito

Este documento tem como propósito definir os requisitos específicos para a solução Global Tracker, detalhando as especificações, capacidades, atributos críticos, e características principais dos módulos Web e Mobile para localização de dispositivos móveis utilizando Web Services.

A avaliação deste documento é importante pois a partir do mesmo, poderão ser conhecidos os benefícios e aplicabilidade do Sistema Web e Mobile para Localização de Dispositivos Móveis utilizando Web Services, bem como prover uma base para as estimativas de tempo e esforços necessários para desenvolver, testar, implantar, e mantê-lo.

Definições referentes ao cronograma de desenvolvimento, responsáveis pela realização das atividades, tecnologias utilizadas não serão descritas neste documento.

1.2. Escopo

A solução Global Tracker será utilizado para permitir o monitoramento de dispositivos móveis para diversos fins, com as coordenadas geográficas obtidas

como latitude e longitude que serão enviadas a um servidor, responsável por armazená-las em um banco de dados.

O servidor em questão deverá prover o acesso e tratamento destas informações utilizando um web service. Um sistema composto por dois módulos, Web e Mobile, irá consumir as informações fornecidas pelo web service para que seja estabelecida uma interface com o usuário, que poderá visualizar a localização do dispositivo móvel em mapas, bem como manipular outras funcionalidades.

1.3. Contexto do Sistema

A solução Global Tracker encontra-se dividida em três módulos:

- Módulo 01: responsável pelo posicionamento e comunicação do hardware a ser monitorado.
- Módulo 02: responsável pela persistência das informações enviadas pelos dispositivos móveis e disponibiliza-las utilizando web services para demais módulos que necessitem consumi-las ou manipulá-las.
- Módulo 03: é o Sistema de Localização dos Dispositivos Móveis Web, que baseia se nas informações obtidas utilizando os web services e Google Maps.

Cada Módulo citado neste breve Contexto estará contemplado neste projeto de conclusão de curso. Integrados estes módulos, irão constituir a Solução de Localização de Dispositivos Móveis conforme é mostrado na Figura 1.

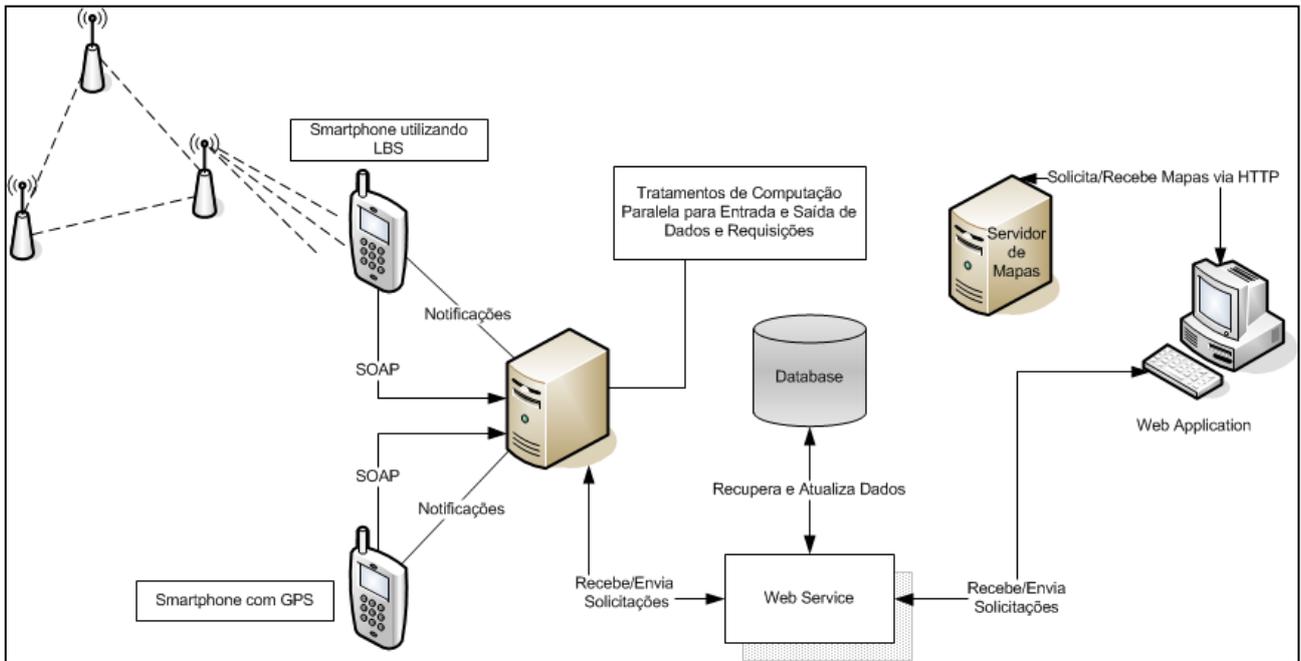


Figura 1 - Diagrama de blocos da solução Localização de Dispositivos Móveis

1.3.1. Módulo 01 - Sistemas Baseados em Localização e GPS: Pesquisa e Desenvolvimento de uma Solução

Utilizando um serviço conhecido como LBS, Sistemas Baseados em Localização, que é capaz de obter informações que possam determinar a localização de um dispositivo móvel, será desenvolvido um software para celular que identificará as três ERBs mais próximas do dispositivo fazendo assim uma triangulação entre estas. A partir dessas informações será possível determinar a localização do dispositivo.

Esta solução também será compatível com dispositivos moveis que possuïrem em GPS e receberá as informações de localização, velocidade, dentre outras.

Os dados recebidos serão processados, caso necessário, e enviados utilizando o protocolo SOAP, através da internet disponibilizada pela rede de telefonia celular, para um servidor web.

1.3.2. Módulo 02 - Gerenciamento de um Servidor para Manipulação de Dados

O provedor de informações e serviços consiste na implementação e configuração de um Servidor que é responsável por receber as informações vindas de um dispositivo móvel, armazená-las em um banco de dados e disponibilizá-las a qualquer aplicação que deseje utilizá-las.

Para realizar esta implementação será desenvolvido um Web Service para que aplicações Desktop, Mobile e Web possam utilizar de maneira simples e rápida.

1.3.3. Módulo 03 - Sistema Web para Localização de Dispositivos Móveis utilizando Web Services

O Sistema de Localização de Dispositivos Móveis é responsável por prover a interface do usuário com o servidor e o dispositivo móvel a ser monitorado. O mesmo é constituído por uma aplicação Web, que utilizando um navegador de internet, permitirá ao usuário fazer uma autenticação, selecionar o dispositivo que pode monitorar e assim fazer consultas do status atual do mesmo, visualizando seu deslocamento através de um mapa que será obtido em um servidor de mapas.

1.4. Stakeholders Principais

A seguir, uma lista de stakeholders envolvidos no desenvolvimento do Sistema de Localização de Dispositivos Móveis. Cada responsabilidade pode ter vários stakeholders de referência que devem ser consultados para coleta de requisitos. Cada responsabilidade também terá um stakeholder principal que, resolverá desacordos e terá aprovação final para requisitos neste papel.

Função/Responsabilidade	Stakeholder Principal	Stakeholders Secundários
Desenvolvedor do projeto Sistemas Baseados em Localização e GPS: Pesquisa e Desenvolvimento de uma Solução	Samuel Moreira	-
Apoio Técnico – Hardware e Software	Samuel Moreira	-
Orientador	Samuel E. L. de Oliveira	-
Desenvolvedor do projeto do web Service	Samuel Moreira	-

Desenvolvedor do Sistema Web para Localização de Dispositivos Móveis utilizando Web Services	Samuel Moreira	-
--	----------------	---

1.5. Abreviações e Acrônimos

<i>Acrônimo/Abreviatura</i>	<i>Termo Expandido</i>
API	Application Programmers Interface (Interface do Programador da Aplicação)
GUI	Graphical User Interface (Interface Gráfica com o Usuário)
ERB	Estação Rádio-Base
GPS	Global Position System (Sistema de Posicionamento Global)
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
LBS	Location-Based Services (Serviços Baseados em Localização)
WAP	Wireless Application Protocol (Protocolo para Aplicações Sem Fio)

1.6. Como este Documento é Organizado

As seguintes seções fornecem todos os requisitos conhecidos do sistema, incluindo ambos, requisitos funcionais e não funcionais. Este documento é completo exceto onde notado com referência a uma fonte externa. É recomendado, mas não necessário, ler as seções em ordem sequencial.

A Seção 2 descreve o as restrições e suposições do projeto. A Seção 3 descreve os riscos do projeto e como estes serão minimizados. A Seção 4 descreve os requisitos funcionais (FR's) do Sistema. A maior parte dos requisitos funcionais existe para dar suporte direto ao processo de negócios; alguns existem para auxiliar operações corretas do próprio sistema. Todos os requisitos funcionais são descritos em termos de casos de uso. A Seção 5 descreve requisitos não funcionais (NFR's) do Sistema. A Seção 6 fornece um Glossário do Projeto que inclui termos

relacionados ao projeto assim como termos relacionados ao desenvolvimento de software.

1.7. Ordens de Mudança de Engenharia

As mudanças necessárias durante o desenvolvimento do Sistema Web e Mobile para Localização de Dispositivos Móveis utilizando Web Services, serão expostas ao Orientador do projeto e aos demais responsáveis pelo desenvolvimento dos módulos relacionados ao mesmo. Uma decisão referente à Mudança de Engenharia será tomada após esta exposição e dependendo do nível da mudança a mesma será adicionada como um novo requisito do Sistema.

1.8. Referências

- “Google Maps API”. Google, 2007.
Disponível em: <http://www.google.com/apis/maps/documentation/>
- Elias F. Lopes, Jaime G. Nogueira, Thiego G. Nacif, Thienne M. Johnson. “Sistemas Baseados em Localização em um Mundo Sem-fio com J2ME e API Location”. Simpósio Brasileiro de Sistemas de Informação, Nov 2006.
Disponível em:
<http://www.cci.unama.br/margalho/portaltcc/tcc2006/pdf/tcc006.pdf>
- “Revista WebMobile”. Edição 11. Artigo J2ME e GPS. Editora: Devmedia
Disponível em: <http://www.devimedia.com.br>
- Templates para documentação do Projeto (SRS, Caso de Uso, etc)
Disponível em:
<http://www.pagliares.com.br/disciplinas/engSoftwareI/index.jsp>

1.9. Restrições e Suposições

É desejável que os Módulos 01 e 02 do Sistema de Localização de Dispositivos Móveis estejam concluídos e integrados ao Módulo 03 para que exista um funcionamento pleno da Solução proposta, porém isso não é imprescindível pois uma simulação pode ser realizada para que seja demonstrado o funcionamento do Sistema Web para Localização de Dispositivos Móveis utilizando Web Services.

1.9.1. Processo de Desenvolvimento e Restrições de Equipe

O processo de desenvolvimento do Sistema Web e Mobile para Localização de Dispositivos Móveis utilizando Web Services será realizado pelo autor deste documento, Samuel Moreira Marques, aluno do curso de Ciência da Computação da Universidade de Itaúna portador do CIU 36520.

1.9.2. Restrições Ambientais e Tecnológicas

O Sistema Web para Localização de Dispositivos Móveis utilizando Web Services pode ser utilizado em sistemas operacionais Windows e Linux, pois dependerá apenas da existência de um navegador de internet. Porém a versão mobile, deverá ser utilizada inicialmente em sistema operacional Android. Visando diminuir os custos de implementação do Sistema, poderão ser utilizadas versões gratuitas dos softwares citados no capítulo 2 da monografia.

Para a exibição dos mapas, será utilizado um componente gratuito para aplicações ASP.Net chamado Subgurim.Net, que utiliza com base de mapas o Google Maps.

As restrições citadas acima não limitam os meios e ferramentas que serão utilizadas para o desenvolvimento do Sistema, se necessária as mesmas poderão ser redefinidas na fase de desenvolvimento.

1.9.3. Restrições de Entrega e Implantação

Restrições referentes à entrega serão limitas à conclusão do cronograma abaixo:

<i>Código da Atividade</i>	<i>Meses</i>
----------------------------	--------------

Código da Atividade	Meses									
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Legenda										
1. Pesquisa de hardware, software e tecnologia a ser utilizada. 2. Documento de Especificação de Requisitos (SRS) 3. Diagrama e especificação de Casos de Uso 4. Diagrama de Classes 5. Implementação do Sistema Web para Localização de Dispositivos Móveis 6. Implementação do Sistema Mobile para Localização de Dispositivos Móveis 7. Escrita da Monografia. 8. Testes e Finalização do Software. 9. Finalização da Monografia										

2. Minimização de Risco

O principal risco para conclusão do projeto é o curto tempo disponível, que será cerca de 5 meses a partir do início da implementação.

2.1. Riscos Tecnológicos

A implementação do projeto não oferece riscos tecnológicos, pois utilizará tecnologias consagradas no mercado e que possuem uma boa base de informações e suporte para esclarecimento de dúvidas.

2.2. Risco de Recursos e Habilidades

Não há riscos de recurso e habilidades devido a conhecimento das ferramentas que serão utilizadas para o desenvolvimento.

2.3. Risco de Requisitos

Inicialmente não existem riscos de requisitos, pois todas as funcionalidades e necessidades dos Sistemas poderão ser plenamente atendidas.

2.4. Riscos Políticos

Não se aplica.

3. Requisitos Funcionais

Este documento engloba os requisitos funcionais do Sistema para Localização de Dispositivos Móveis utilizando Web Services, desenvolvido para a plataforma Web e Mobile.

Como citado anteriormente este sistema será capaz permitir ao usuário localizar um dispositivo móvel através de suas coordenadas geográficas em um mapa. Os procedimentos descritos demonstrarão como será o fluxo de funcionamento do sistema.

Para utilizar o sistema o usuário deverá inicialmente informar um usuário e senha e em seguida selecionar um dos dispositivos moveis que tem acesso a monitorar. Após estas operações iniciais o usuário deverá selecionar qual ação deseja realizar e dentre elas estarão disponíveis Visualizar Histórico, Visualizar Alertas e Criar Cerca Eletrônica.

Um dispositivo poderá ou não estar sendo monitorado em determinado momento, ou seja o dispositivo está enviando ou não suas coordenadas geográficas para o servidor, para isso o usuário possui funcionalidades que iniciam ou interrompem o monitoramento.

3.1. Características Principais

Nesta seção, serão classificadas as características principais para Sistema Web e Mobile para Localização de Dispositivos Móveis utilizando Web Services através de três categorias.

Características essenciais, ou seja, que não podem deixar de ser implementadas. Características de alto valor que podem deixar de ser implementadas, embora isto não seja desejável. Características Não Essenciais que são aquelas que podem ser dispensáveis ou implementadas posteriormente.

3.1.1. Características Essenciais

- Realizar comunicação do dispositivo móvel com o web service.
- Obter informações pertinentes à localização do dispositivo mobile para que sejam enviadas para a base de dados através do web service.
- Realizar a autenticação do usuário para acesso ao Sistema Web.
- Prover toda a segurança necessária e privacidade para o tráfego de informações.
- Obter e exibir mapas que irão conter as coordenadas do dispositivo móvel monitorado.
- Permitir a configuração de cercas eletrônicas.

3.1.2. Características de Alto Valor

- Traçar rotas entre pontos.
- Emitir alertas quando um dispositivo monitorado ultrapassar a área determinada pela cerca elétrica.
- Gerar históricos das rotas realizadas pelo dispositivo móvel.

3.1.3. Características Não Essenciais

- Implementar todas as funcionalidades da interface Web para a Mobile.

4. Diagrama de Caso de Uso

Para demonstrar em alto nível as ações e funcionalidades que estarão disponíveis para os usuários, foram criados os diagramas abaixo:

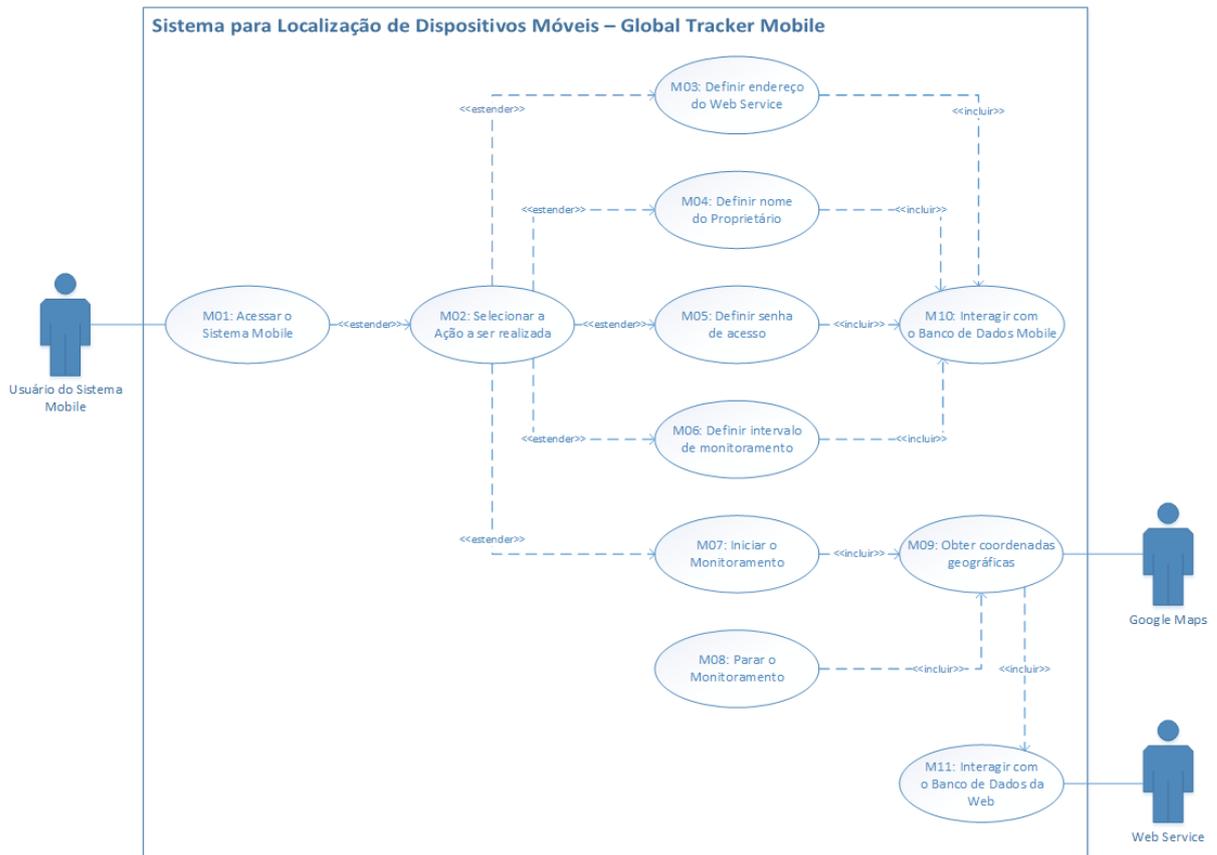


Figura 2 – Diagrama de Caso de Uso – Usuário do Global Tracker Mobile

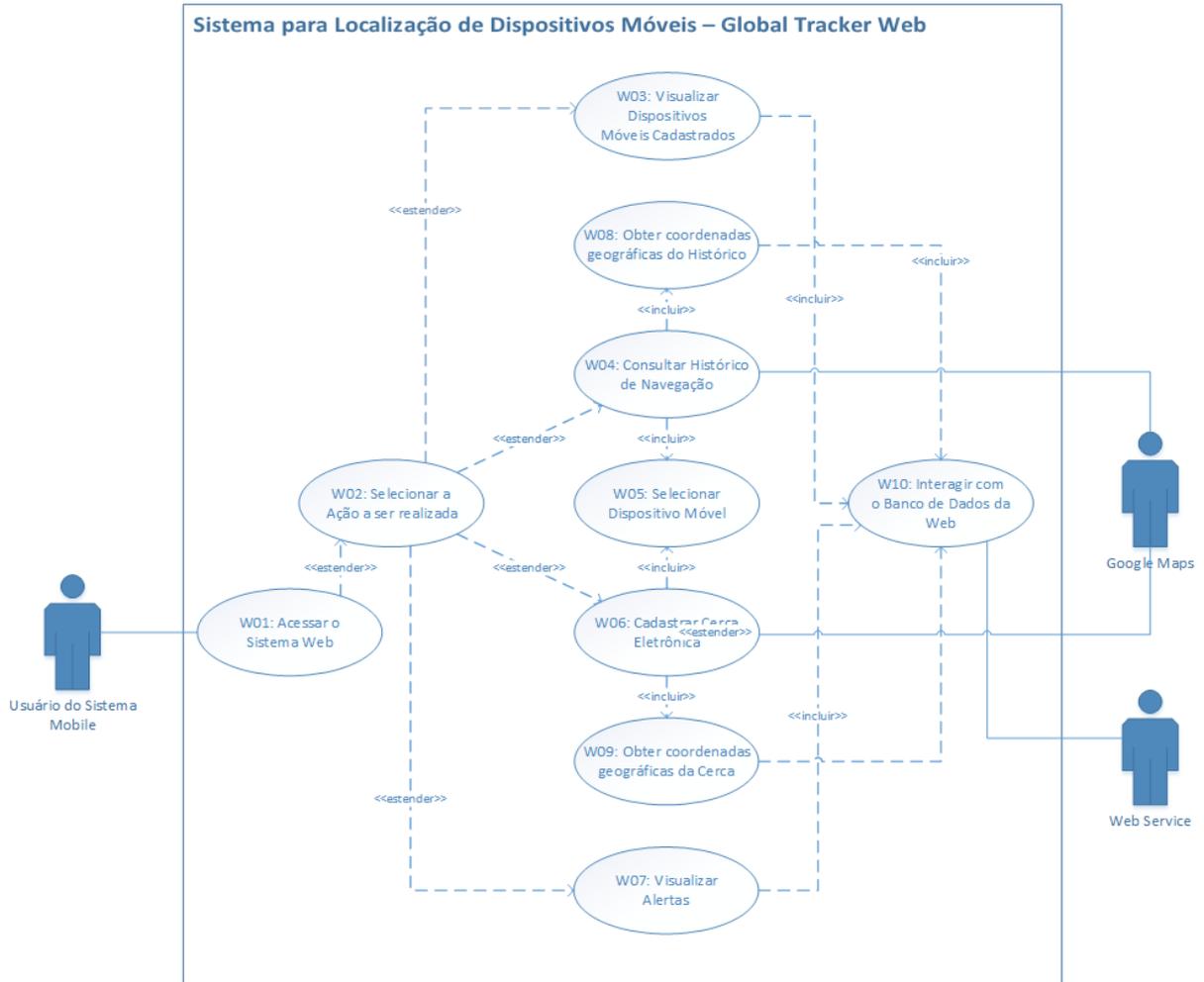


Figura 3 – Diagrama de Caso de Uso – Usuário do Global Tracker Web

4.1. Atores

Estes são as funções exercidas por pessoas e sistemas que interagem com o sistema.

Nome do Ator	Descrição
Usuário do Sistema Web	Usuário que irá acessar o sistema utilizando um PC.
Usuário do Sistema Mobile	Usuário que irá acessar o sistema utilizando um Smartphone ou Tablet.

<i>Nome do Ator</i>	<i>Descrição</i>
Web Service	Serviço que irá prover o acesso ao banco de dados e a comunicação com o dispositivo móvel.
Google Maps	Serviço que irá prove os mapas necessários para que ao Sistema Web e Mobile.

4.1.1. Ator: Usuário do Sistema Web

Uso principal da aplicação: este ator irá utilizar a aplicação para monitorar determinado dispositivo móvel, criar cercar eletrônicas, consultar históricos e visualizar alertas.

Tempo médio de uso por vez: *05 a 60 minutos.*

Frequência média de uso: diária.

Educação: formação básica e treinamento se possível.

Experiência: com computadores no geral e acesso à internet.

4.1.2. Ator: Usuário do Sistema Mobile

Uso principal da aplicação: este ator irá utilizar a aplicação para configurar o dispositivo móvel, iniciar e interromper um monitoramento.

Tempo médio de uso por vez: *03 a 10 minutos.*

Frequência média de uso: diária.

Educação: formação básica e treinamento se possível.

Experiência: com smartphones ou tablets e acesso à internet através destes dispositivos.

4.1.3. Ator: Web Service

Uso principal da aplicação: este ator é um serviço que será implementado no projeto de conclusão de curso Gerenciamento de um Servidor para Manipulação de Dados Geográficos de Dispositivos Móveis Celulares e

referenciado para que o Global Tracker Web e Mobile possuam comunicação com o banco de dados.

Tempo médio de uso por vez: 24 horas.

Frequência média de uso: constante.

4.1.4. Ator: GoogleMaps

Uso principal da aplicação: este ator é um serviço público implementado pela Google que disponibiliza Mapas através de fotos tiradas de satélite e definições das ruas das principais cidades do país. Este será utilizado para prover os mapas necessários ao Global Tracker Web e Mobile.

Tempo médio de uso por vez: 24 horas.

Frequência média de uso: constante.

4.2. Casos de Uso

Global Tracker Mobile			
<i>Nome do Caso de Uso</i>	<i>Prioridade</i>	<i>Código</i>	<i>Descrição</i>
Acessar o Sistema Mobile	1	M01	Abrir o sistema mobile.
Selecionar a Ação a ser realizada	1	M02	Definir qual ação será realizada.
Definir o endereço do Web Service	1	M03	Informar o endereço do web service.
Definir o nome do Proprietário	1	M04	Informar o nome do proprietário do dispositivo móvel.
Definir senha de acesso	1	M05	Informar o senha de acesso para visualizar dados do dispositivo móvel na Web.
Definir intervalo de monitoramento	1	M06	Informar intervalo de tempo que o dispositivo móvel enviará os dados.
Iniciar o Monitoramento	1	M07	Iniciar o processo de monitoramento do dispositivo.
Parar o Monitoramento	1	M08	Interromper o processo de monitoramento do dispositivo.
Obter coordenadas geográficas	1	M09	Obter coordenadas geográficas.

Global Tracker Mobile			
Nome do Caso de Uso	Prioridade	Código	Descrição
Interagir com o Banco de Dados Mobile	1	M10	Enviar e receber informações do banco de dados local.
Interagir com o Banco de Dados Mobile	1	M11	Enviar e receber informações do banco de dados através do web service.

Global Tracker Web			
Nome do Caso de Uso	Prioridade	Código	Descrição
Acessar o Sistema Web	1	W01	Faz login no sistema.
Selecionar a ação a ser realizada	1	W02	Usuário seleciona a ação a ser realizada.
Visualizar dispositivos móveis cadastrados	1	W03	Visualizar lista de dispositivos móveis cadastrados.
Consultar Histórico de Navegação	1	W04	Exibe os pontos pelo qual o dispositivo móvel passou através do mapa.
Selecionar Dispositivo Móvel	1	W05	Selecionar o dispositivo móvel que deseja informações.
Cadastrar Cerca Eletrônica	1	W06	Definir perímetro de em que o dispositivo pode circular.
Visualizar Alertas	1	W07	Visualizar alertas de quando o dispositivo ultrapassa o perímetro determinado.
Obter coordenadas geográficas do Histórico	1	W08	Visualizar locais pelos quais o dispositivo passou.
Obter coordenadas geográficas da Cerca	1	W09	Visualizar perímetro determinado para cerca eletrônica.
Interagir com Banco de Dados Web	1	W10	Enviar e receber informações do Banco de Dados através do web service.

4.2.1. Aplicações

Nome da Aplicação	Descrição / Caso de Uso
Global Tracker Mobile	Interface mobile para o usuário inicia o monitoramento de dispositivos. Suporta os Casos de Uso: M01, M02, M03, M04, M05, M06, M07, M08, M09, M10 e M11.

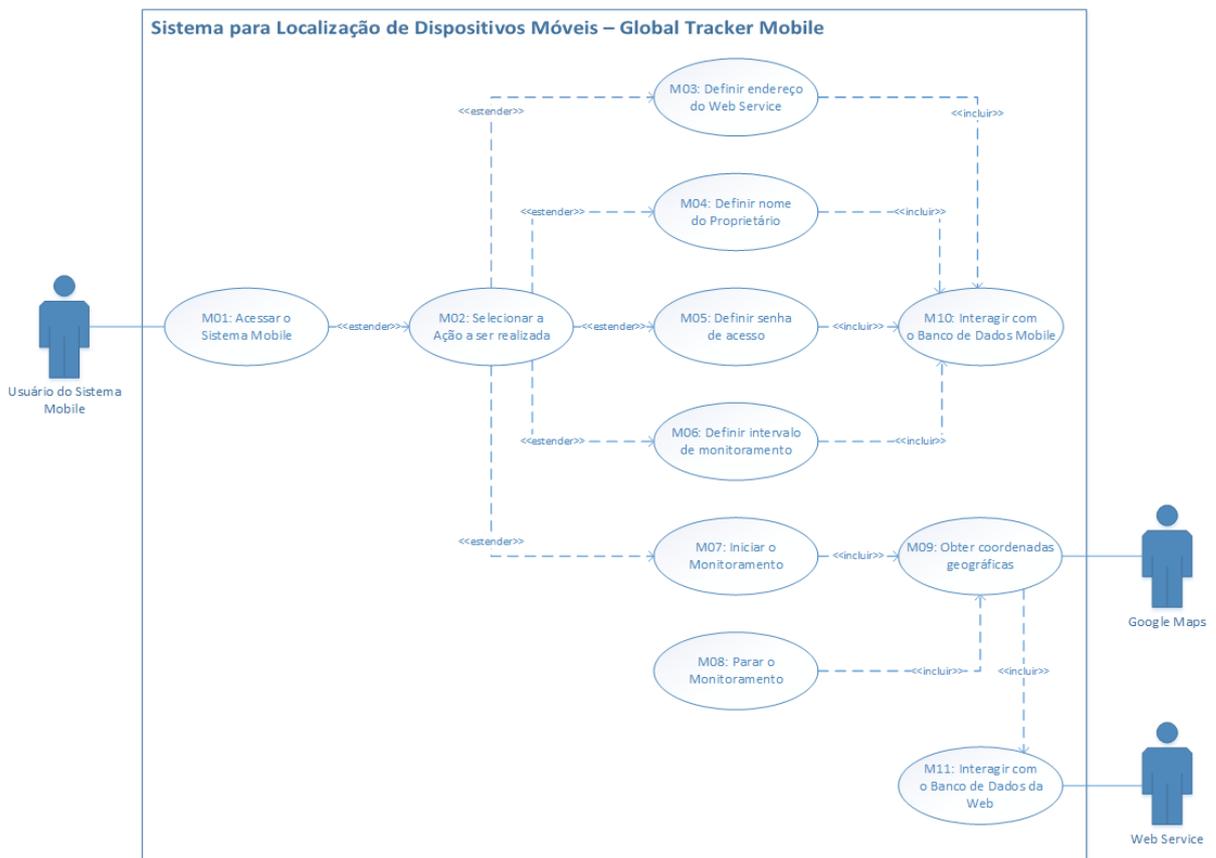
Nome da Aplicação	Descrição / Caso de Uso
Global Tracker Web	Interface web para o usuário realizar o monitoramento de dispositivos. Suporta os Casos de Uso: W01, W02, W03, W04, W05, W06, W07, W08, W09 e W10.

Anexo B – Especificação de Caso de Uso

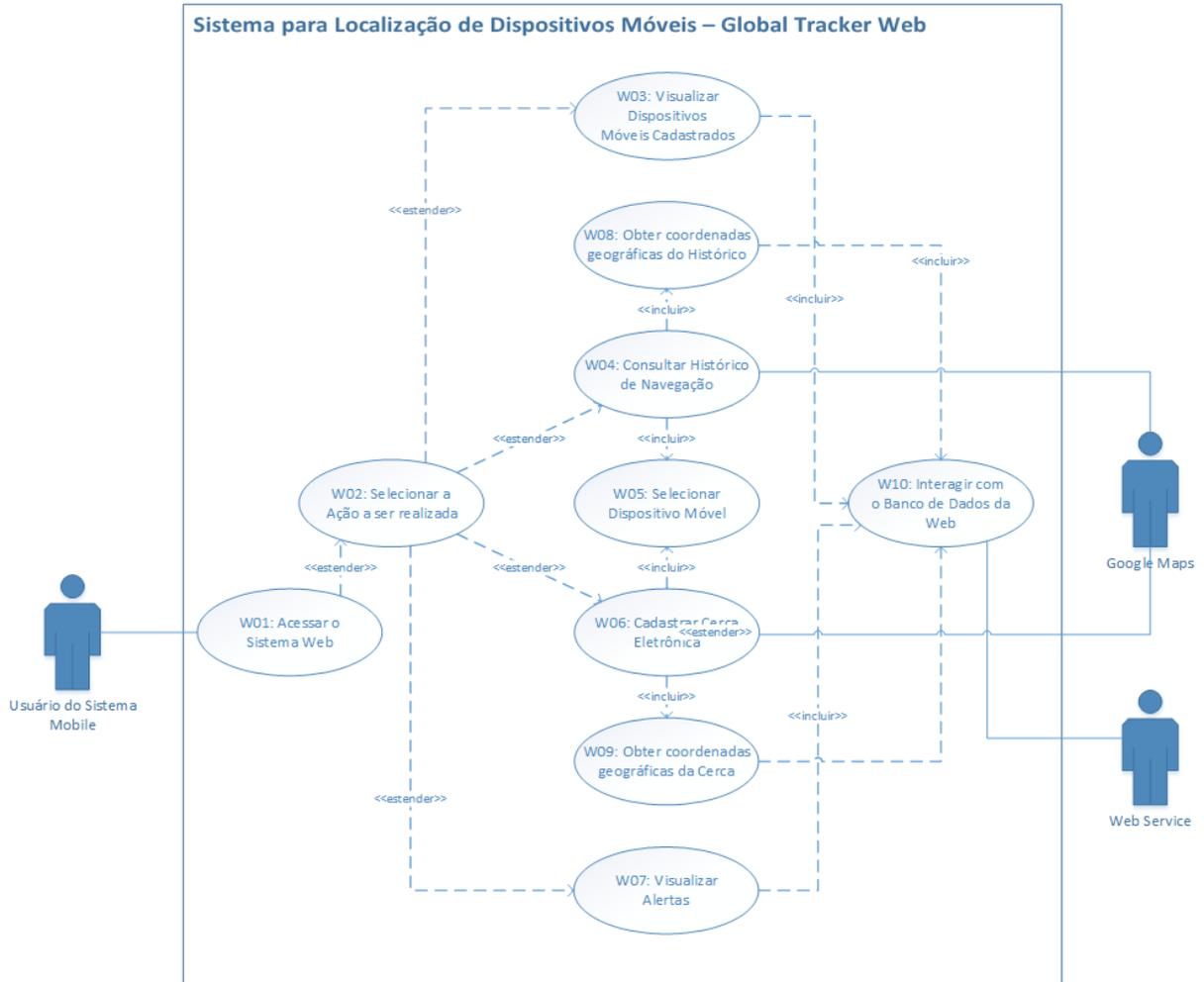
Abaixo encontra-se a especificação de casos de uso da solução Global Tracker preparada por Samuel Moreira Marques, revisada junto ao orientador Samuel E. L. de Oliveira.

1. Diagramas de Caso de Uso

1.1. Sistema para Localização de Dispositivos Móveis – Global Tracker Mobile



1.2. Sistema para Localização de Dispositivos Móveis – Global Tracker Web



2. Formulário Complexo de Caso de Uso

2.1. Sistema para Localização de Dispositivos Móveis – Global Tracker Mobile

*#/ID do Caso de Uso e nome	M01: Acessar o Sistema Mobile
Descrição	Caso de uso para fazer localizar a aplicação móvel no dispositivo e acessá-la.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo

*#/ID do Caso de Uso e nome	M01: Acessar o Sistema Mobile
* Pré-condições / Suposições	Nenhuma
Gatilho	W01: Acessar o Sistema Mobile
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário localiza o aplicativo. 2. O usuário clica no aplicativo. 3. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M02: Selecionar Ação a Ser Realizada
Descrição	Caso de uso para que o usuário escolha qual ação será realizada.
*Ator(es)	Primário: Usuário Secundário: Sistema (implícito)
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário escolhe a ação a ser realizada. 2. O usuário dispara a ação que deseja. 3. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	<i>M03: Definir o endereço do Web Service</i>
Descrição	Caso de uso em que o usuário informa o endereço do web service ou permite que seja mantido o endereço padrão.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa o endereço do web service. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	<i>M04: Definir o nome do proprietário</i>
Descrição	O usuário informa o nome do proprietário.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa o nome do proprietário. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	

*#/ID do Caso de Uso e nome	M04: Definir o nome do proprietário
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M05: Definir senha de acesso
Descrição	Caso de uso em que o usuário informa a senha de acesso que será utilizada no Global Tracker Web para autenticação.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa a senha de acesso. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M06: Definir intervalo de monitoramento
Descrição	Caso de uso em que o Usuário define o intervalo de tempo em que as informações serão enviadas para o banco de dados.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	

*#/ID do Caso de Uso e nome	M06: Definir intervalo de monitoramento
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa o intervalo de monitoramento em minutos. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M07: Iniciar o Monitoramento
Descrição	Caso de uso que o usuário utiliza para solicitar ao dispositivo móvel que inicie o envio de suas coordenadas geográficas conforme configuração pré-definida.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema e possuir uma conexão com a internet.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O sistema envia uma requisição para que o dispositivo móvel inicie o envio de suas coordenadas geográficas para o servidor. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M08: Parar o Monitoramento
-----------------------------	-----------------------------------

*#/ID do Caso de Uso e nome	<i>M08: Parar o Monitoramento</i>
Descrição	Caso de uso que o usuário utiliza para solicitar ao dispositivo móvel que pare de enviar suas coordenadas geográficas para o Servidor.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O sistema envia uma requisição para que o dispositivo móvel pare de enviar de suas coordenadas geográficas para o servidor. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	<i>M09: Obter coordenadas geográficas</i>
Descrição	Caso de uso que o sistema obtenha as coordenadas geográficas nas quais o dispositivo se encontra, bem como o endereço correspondente às coordenadas.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema, possuir uma conexão com a internet e se possível <i>GPS</i> ativado.
Gatilho	M7: Iniciar o Monitoramento

*#/ID do Caso de Uso e nome	M09: Obter coordenadas geográficas
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O sistema inicia o monitoramento. 2. O dispositivo obtém as coordenadas geográficas. 3. O sistema tenta obter o endereço correspondente à coordenada geográfica. 4. O sistema verifica se existe uma cerca eletrônica cadastrada para o dispositivo corrente. 5. Caso exista o sistema envia um alerta e insere a coordenada geográfica. Se não existir insere apenas o alerta. 6. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M10: Interagir com o Banco de Dados Mobile
Descrição	Caso de uso que o sistema utiliza para gravar e consultar informações no banco de dados local do dispositivo móvel.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	M03: Definir endereço do Web Service, M04: Definir nome do Proprietário, M05: Definir Senha de Acesso, M06: Definir Intervalo de Monitoramento
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O sistema grava ou recupera informações do banco de dados local. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	

*#/ID do Caso de Uso e nome	M10: Interagir com o Banco de Dados Mobile
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	M11: Interagir com o Banco de Dados da Web
Descrição	Caso de uso que o sistema utiliza para gravar e recuperar dados através do web service.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O monitoramento deve estar inicializado.
Gatilho	M09: Obter coordenadas geográficas.
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O sistema grava ou recupera informações do banco de dados utilizando o web service. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

2.2. Sistema para Localização de Dispositivos Móveis – Global Tracker Web

*#/ID do Caso de Uso e nome	W01: Acessar o Sistema Web
Descrição	Caso de uso para fazer a autenticação do usuário.
*Ator(es)	Primário: Usuário
Prioridade	Essencial

*#/ID do Caso de Uso e nome	W01: Acessar o Sistema Web
Risco	Baixo
* Pré-condições / Suposições	O usuário deve possuir acesso à internet.
Gatilho	W01: Acessar o Sistema Web
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa seu login e senha. 2. O sistema verifica os dados informados e permite ou bloqueia o acesso do usuário. 3. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W02: Selecionar a Ação a ser realizada
Descrição	Caso de uso para que o usuário decida qual ação irá realizar.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário define a ação que irá realizar. 2. O usuário executa a ação. 3. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W03: Visualizar Dispositivos Móveis Cadastrados
Descrição	Caso de uso que o usuário visualiza a lista de dispositivos móveis cadastrados.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário visualiza informações referentes aos dispositivos móveis que tem acesso a visualizar. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W04: Consultar histórico de navegação
Descrição	Caso de uso que o usuário visualiza a rota pela qual o dispositivo móvel passou em um determinado período.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema e ter um dispositivo móvel selecionado.
Gatilho	

*#/ID do Caso de Uso e nome	W04: Consultar histórico de navegação
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa o período que deseja consultar. 2. O sistema requisita um mapa. 3. O sistema requisita o histórico das coordenadas geográficas conforme o período informado pelo usuário. 4. O sistema exibe o mapa requisitado. 5. O sistema traça as coordenadas recebidas no mapa que está sendo exibido ao usuário. 6. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W05: Selecionar Dispositivo Móvel
Descrição	Caso de uso que o usuário utiliza para selecionar um dispositivo móvel que quer visualizar histórico ou cadastrar cerca eletrônica.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.
Gatilho	W04: Consultar Histórico de Navegação, W06: Cadastrar Cerca Eletrônica
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário seleciona o dispositivo móvel. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W06: Cadastrar Cerca Eletrônica
Descrição	Caso de uso que o usuário utiliza para realizar uma marcação eletrônica em determinada área.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema e ter um dispositivo móvel selecionado.
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário deve ativar a marcação de cerca eletrônica através da aplicação web. 2. O usuário deve iniciar a marcação do perímetro seguindo uma lógica para determinar o perímetro. 3. Após determinar 3 pontos a área começa a ser exibida. 4. A área demarcada poderá possuir inúmeros pontos demarcados. 5. Após determinar a área o usuário salva a configuração da mesma. 6. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W07: Visualizar Alertas
Descrição	Caso de uso que o usuário utiliza para visualizar alerta de quando o dispositivo são da área delimitada pela cerca eletrônica.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema.

*#/ID do Caso de Uso e nome	W07: Visualizar Alertas
Gatilho	
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário visualiza a lista de dispositivos que possuem alertas. 2. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W08: Obter Coordenadas Geográficas do Histórico
Descrição	Caso de uso que o sistema utiliza para obter coordenadas do histórico de monitoramento de um dispositivo móvel.
*Ator(es)	Primário: Sistema Secundário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema e ter um dispositivo móvel selecionado.
Gatilho	W04: Selecionar Ação a ser Realizada
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário informa o período que deseja consultar. 2. O sistema requisita um mapa. 3. O sistema requisita o histórico das coordenadas geográficas conforme o período informado pelo usuário. 4. O sistema exibe o mapa requisitado. 5. O sistema traça as coordenadas recebidas no mapa que está sendo exibido ao usuário. 6. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	

*#/ID do Caso de Uso e nome	W08: Obter Coordenadas Geográficas do Histórico
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W09: Obter Coordenadas Geográficas da Cerca
Descrição	Caso de uso que o sistema utiliza para obter coordenadas do perímetro determinado para um dispositivo móvel.
*Ator(es)	Primário: Usuário
Prioridade	Essencial
Risco	Baixo
* Pré-condições / Suposições	O usuário deve estar acessando o sistema e possuir um dispositivo móvel selecionado.
Gatilho	W06: Cadastrar Cerca Eletrônica
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none"> 1. O sistema requisita um mapa. 2. O sistema requisita os pontos correspondentes ao perímetro definido pelo usuário anteriormente. 3. O sistema exibe o mapa requisitado. 4. O sistema plota a área que delimita o perímetro. 5. O caso de uso termina.
*Pós-condições	
*Cenário alternativo	
Requisitos não funcionais que devem ser satisfeitos pelo caso de uso	Vide SRS.

*#/ID do Caso de Uso e nome	W10: Interagir com Banco de Dados Web
Descrição	Caso de uso que o sistema utiliza para realizar inserções, atualizações e seleções no Banco de Dados que irá conter informações relevantes sobre o posicionamento do dispositivo que se quer monitorar.
*Ator(es)	Primário: Global Tracker Web
Prioridade	Essencial

*#/ID do Caso de Uso e nome	W10: Interagir com Banco de Dados Web
Risco	Baixo
* Pré-condições / Suposições	A requisição feita pelo Sistema Global Tracker Web deve ter sido processada.
Gatilho	W08: Obter Coordenadas Geográficas do Histórico, W09: Obter Coordenadas Geográficas da Cerca.
* Cenário Principal Fluxo de Eventos	<ol style="list-style-type: none">1. Executar determinado método do Web Service.2. Processar a resposta retornada pelo Web Service, quando necessário.3. O caso de uso termina.
Observação	

Anexo C – Código Fonte Comentado do Global Tracker Mobile

Abaixo encontra-se o código fonte comentado das principais classes do projeto GlobalTracker.Mobile.

1. ActivitySplashScreen.cs

```
using System;
using Android.App;
using Android.Content;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.OS;
using Android.Content.PM;
using Java.Lang;

namespace GlobalTracker.Mobile
{
    /// <summary>
    /// Esta classe foi criada para exibir a tela inicial da aplicação.
    /// </summary>
    [Activity(Label = "Global Tracker Mobile", MainLauncher = true, Theme =
"@style/Theme.Splash", NoHistory = true, Icon = "@drawable/icon")]
    public class ActivitySplashScreen : Activity
    {
        /// <summary>
        /// Evento disparado ao acessar a tela inicial da aplicação.
        /// </summary>
        /// <param name="bundle">Container de componentes.</param>
        protected override void OnCreate(Bundle bundle)
        {
            RequestedOrientation = ScreenOrientation.Portrait;
            base.OnCreate(bundle);
            StartThread();
        }

        /// <summary>
        /// Thread para força que a tela inicial seja exibida por pelo menos 2 segundos.
        /// </summary>
        private void StartThread()
        {
            new Thread(new Runnable(
                delegate
                {
                    Thread.Sleep(2000);
                    var activityMain = new Intent(this, typeof(ActivityFormMain));
                    Finish();
                    StartActivity(activityMain);
                }
            )).Start();
        }
    }
}
```

```
}
}
```

2. ActivityFormMain.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using R = Android;
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.Content.PM;
using Android.Locations;
using Android.Telephony;
using System.Data;
using Java.Lang;

namespace GlobalTracker.Mobile
{
    /// <summary>
    /// Esta é a classe principal do projeto Mobile e é utilizada para:
    /// - Interagir com o Web Service;
    /// - Inserir o dispositivo no banco de dados;
    /// - Obter as coordenadas geográficas;
    /// - E Inserir as coordenadas geográficas.
    ///
    /// Ela foi criada para satisfazer diretamente os casos de uso:
    /// - M01, M02, M03, M04, M05, M06, M07, M08 e M09.
    /// </summary>
    ///
    [Activity(Label = "Global Tracker Mobile", Theme = "@style/Theme.Padrao")]
    public class ActivityFormMain : Activity, ILocationListener // Herda de Activity e
    implementa ILocationListener
    {
        private Location _localizacaoCorrente; // Objeto utilizado para armazenar as
        coordenadas corrente.
        private LocationManager _locationManager; // Objeto utilizado para interagir com
        todos mecanismo de coordenadas geográficas e componentes envolvidos.
        private TextView _coordenadasText; // Controle de texto para exibição de
        informações referentes às coordenadas.
        private TextView _enderecoText; // Controle de texto para exibição de informações
        referentes ao endereço.

        private TextView _nomeText; // Controle de texto para exibição do nome.
        private TextView _senhaText; // Controle de texto para exibição da senha.
        private TextView _intervaloText; // Controle de texto para exibição do intervalo.
        private TextView _emailText; // Controle de texto para exibição de-mail.
        private TextView _webServiceText; // Controle de texto para exibição WebService.
        private TextView _imeiText; // Controle de texto para exibição do IMEI.
    }
}
```

```

    private string _locationProvider = string.Empty; // Nome do meio que será
    utilizado para obter a localização (gps ou ethernet)
    private NotificationManager notificationManager;
    WebServiceReference.GeneralService _service = new
    WebServiceReference.GeneralService();

    /// <summary>
    /// Método implementado para satisfazer ILocationListener.
    /// </summary>
    public void OnProviderDisabled(string provider)
    {
    }

    /// <summary>
    /// Método implementado para satisfazer ILocationListener.
    /// </summary>
    public void OnProviderEnabled(string provider)
    {
    }

    /// <summary>
    /// Método implementado para satisfazer ILocationListener.
    /// </summary>
    public void OnStatusChanged(string provider, Availability status, Bundle extras)
    {
    }

    /// <summary>
    /// Método implementado para alterar as informações exibidas no controle de texto
    que exibe as coordenadas geográficas.
    /// </summary>
    /// <param name="location">Localização Corrente</param>
    public void OnLocationChanged(Location location)
    {
        _localizacaoCorrente = location;
        if (_localizacaoCorrente == null)
        {
            _coordenadasText.Text = "Não é possível determinar a localização.";
        }
        else
        {
            _coordenadasText.Text = string.Format("{0},{1}",
            _localizacaoCorrente.Latitude, _localizacaoCorrente.Longitude);
        }
    }

    /// <summary>
    /// Evento disparado ao acessar a tela principal da aplicação.
    /// </summary>
    /// <param name="bundle">Container de componentes.</param>
    protected override void OnCreate(Bundle bundle)
    {
        // Seta a exibição da aplicação sempre para Portrait.
        RequestedOrientation = ScreenOrientation.Portrait;

        base.OnCreate(bundle);
        SetContentView(Resource.Layout.FormMain);

        // Recupera o controle para exibição do endereço e coordenadas.

```

```

_enderecoText = FindViewById<TextView>(Resource.Id.address_text);
_coordenadasText = FindViewById<TextView>(Resource.Id.location_text);

// Recupera controles que irão receber os parâmetros de configuração.
_nomeText = FindViewById<TextView>(Resource.Id.FormMaintxtNome);
_senhaText = FindViewById<TextView>(Resource.Id.FormMaintxtSenha);
_intervaloText = FindViewById<TextView>(Resource.Id.FormMaintxtIntervalo);
_emailText = FindViewById<TextView>(Resource.Id.FormMaintxtEmail);
_webServiceText = FindViewById<TextView>(Resource.Id.FormMaintxtWebService);

// Recupera valores definidos como parâmetros.
_nomeText.Text =
MobileDatabaseSigleton.MobileDatabase.RetornaNomeConfiguracoes();
_senhaText.Text =
MobileDatabaseSigleton.MobileDatabase.RetornaSenhaConfiguracoes();
_intervaloText.Text =
MobileDatabaseSigleton.MobileDatabase.RetornaIntervaloConfiguracoes();
_emailText.Text =
MobileDatabaseSigleton.MobileDatabase.RetornaEMailConfiguracoes();
_webServiceText.Text =
MobileDatabaseSigleton.MobileDatabase.RetornaWebServiceConfiguracoes();

// Atibui eventos que serão utilizados para salvar as definições de
parametrização.
_nomeText.AfterTextChanged += new
EventHandler<R.Text.AfterTextChangedEventArgs>(_nomeText_AfterTextChanged);
_senhaText.AfterTextChanged += new
EventHandler<R.Text.AfterTextChangedEventArgs>(_senhaText_AfterTextChanged);
_intervaloText.AfterTextChanged += new
EventHandler<R.Text.AfterTextChangedEventArgs>(_intervaloText_AfterTextChanged);
_emailText.AfterTextChanged += new
EventHandler<R.Text.AfterTextChangedEventArgs>(_emailText_AfterTextChanged);
_webServiceText.AfterTextChanged += new
EventHandler<R.Text.AfterTextChangedEventArgs>(_webServiceText_AfterTextChanged);

// Recupera e atribuiu o evento de clique no botão para iniciar o monitoramento.
FindViewById<TextView>(Resource.Id.FormMainbuttonIniciar).Click +=
FormMainbuttonIniciar_OnClick;
// Recupera e atribuiu o evento de clique no botão para para o monitoramento.
FindViewById<TextView>(Resource.Id.FormMainbuttonParar).Click +=
FormMainbuttonParar_OnClick;

// Inicializa o gerenciador que será utilizado para obter as coordenadas
geográficas.
InitializeLocationManager();

// Recupera o controle para exibição do IMEI.
_imeiText = FindViewById<EditText>(Resource.Id.FormMaintxtUniqueId);
// Recupera o IMEI do dispositivo móvel.
_imeiText.Text = RetornaImeiDispositivo();
}

/// <summary>
/// Salva alteração realizada no campo endereço do Web Service no banco de dados
local.
/// </summary>
/// <param name="sender">Controle que invocou a alteração</param>
/// <param name="e">Parâmetros passados.</param>

```

```

    void _webServiceText_AfterTextChanged(object sender,
R.Text.AfterTextChangedEventArgs e)
    {
        MobileDatabaseSingleton.MobileDatabase.Execute("update Configuracao set
webservice = '" + _webServiceText.Text + "'");
        //_service.Url = _webServiceText.Text;
    }

    /// <summary>
    /// Salva alteração realizada no campo e-mail no banco de dados local.
    /// </summary>
    /// <param name="sender">Controle que invocou a alteração</param>
    /// <param name="e">Parâmetros passados.</param>
    void _emailText_AfterTextChanged(object sender, R.Text.AfterTextChangedEventArgs
e)
    {
        MobileDatabaseSingleton.MobileDatabase.Execute("update Configuracao set email =
'" + _emailText.Text + "'");
    }

    /// <summary>
    /// Salva alteração realizada no campo intervalo no banco de dados local.
    /// </summary>
    /// <param name="sender">Controle que invocou a alteração</param>
    /// <param name="e">Parâmetros passados.</param>
    void _intervaloText_AfterTextChanged(object sender,
R.Text.AfterTextChangedEventArgs e)
    {
        MobileDatabaseSingleton.MobileDatabase.Execute("update Configuracao set intervalo
= '" + _intervaloText.Text + "'");
    }

    /// <summary>
    /// Salva alteração realizada no campo senha no banco de dados local.
    /// </summary>
    /// <param name="sender">Controle que invocou a alteração</param>
    /// <param name="e">Parâmetros passados.</param>
    void _senhaText_AfterTextChanged(object sender, R.Text.AfterTextChangedEventArgs
e)
    {
        MobileDatabaseSingleton.MobileDatabase.Execute("update Configuracao set senha =
'" + _senhaText.Text + "'");
    }

    /// <summary>
    /// Salva alteração realizada no campo nome no banco de dados local.
    /// </summary>
    /// <param name="sender">Controle que invocou a alteração</param>
    /// <param name="e">Parâmetros passados.</param>
    void _nomeText_AfterTextChanged(object sender, R.Text.AfterTextChangedEventArgs e)
    {
        MobileDatabaseSingleton.MobileDatabase.Execute("update Configuracao set nome = '"
+ _nomeText.Text + "'");
    }

    /// <summary>
    /// Método utilizado para definir qual método estará disponível para obtenção das
coordenadas.
    /// </summary>

```

```

private void InitializeLocationManager()
{
    _locationManager = (LocationManager)GetSystemService(LocationService);
    var criteriaForLocationService = new Criteria
    {
        Accuracy = Accuracy.Fine
    };
    var acceptableLocationProviders =
_locationManager.GetProviders(criteriaForLocationService, true);

    if (acceptableLocationProviders.Any()) // Caso exista algum provider ativo, como
por exemplo o GPS...
    {
        _locationProvider = acceptableLocationProviders.First(); //... o mesmo será
retornado aqui, ...
    }
    else
    {
        _locationProvider = "network"; //... caso contrário será utilizado a rede de
telefonía ou o roteador wi-fi para determinar a localização.
    }
}

/// <summary>
/// Evento executado sempre que a tela da aplicação é exibida.
/// </summary>
protected override void OnResume()
{
    base.OnResume();
    _locationManager.RequestLocationUpdates(_locationProvider, 0, 0, this);
}

/// <summary>
/// Evento disparado ao clicar no botão utilizado para interromper o
monitoramento.
/// </summary>
/// <param name="sender">Controle que invocou a ação</param>
/// <param name="eventArgs">Argumentos passados pelo controle</param>
private void FormMainbuttonParar_OnClick(object sender, EventArgs eventArgs)
{
    // Interrompe serviço de envio de coordenadas.
    StopService(new Intent(this, typeof(ServicoEnvioCoordenadas)));
}

public void AlertDialog(Context activity, string message, string button,
EventHandler<DialogClickEventArgs> methodPositive)
{
    AlertDialog alertDialog = new AlertDialog.Builder(activity).Create();
    alertDialog.SetMessage(message);
    alertDialog.SetButton((int)DialogButtonType.Positive, button, methodPositive);
    alertDialog.SetTitle("Atenção");
    alertDialog.Show();
}

/// <summary>
/// Evento disparado ao clicar no botão utilizado para iniciar o monitoramento.
/// </summary>
/// <param name="sender">Controle que invocou a ação</param>
/// <param name="eventArgs">Argumentos passados pelo controle</param>

```

```

private void FormMainbuttonIniciar_OnClick(object sender, EventArgs eventArgs)
{
    // Valida preenchimento dos campos de Configurações.
    if (_enderecoText.Text.Trim().Length == 0)
    {
        AlertDialog(this, "Por favor, preencha o campo Web Service.", "OK", null);
        return;
    }
    else if (_nomeText.Text.Trim().Length == 0)
    {
        AlertDialog(this, "Por favor, preencha o campo Nome do Proprietário.", "OK",
null);
        return;
    }
    else if (_emailText.Text.Trim().Length == 0)
    {
        AlertDialog(this, "Por favor, preencha o campo E-Mail.", "OK", null);
        return;
    }
    else if (_senhaText.Text.Trim().Length == 0)
    {
        AlertDialog(this, "Por favor, preencha o campo Senha.", "OK", null);
        return;
    }
    else if (_intervaloText.Text.Trim().Length == 0)
    {
        AlertDialog(this, "Por favor, preencha o campo Intervalo.", "OK", null);
        return;
    }
}

// Inicializa parâmetros que serão passados para o serviço de envio de
coordenadas.
Intent intent = new Intent(this, typeof(ServicoEnvioCoordenadas));
intent.PutExtra("intervalo", this._intervaloText.Text);
intent.PutExtra("webService", this._webServiceText.Text);
intent.PutExtra("imei", RetornaImeiDispositivo());

// Interrompe serviço de envio de coordenadas, para caso o mesmo esteja
inicializado não fique mais que uma instância.
StopService(new Intent(this, typeof(ServicoEnvioCoordenadas)));

// Inicia serviço de envio de coordenadas.
new Thread(new Runnable(
    delegate
    {
        StartService(intent);
    }
)).Start();

_locationManager = (LocationManager)GetSystemService(LocationService);
var criteriaForLocationService = new Criteria{Accuracy = Accuracy.Fine};
var acceptableLocationProviders =
_locationManager.GetProviders(criteriaForLocationService, true);

if (acceptableLocationProviders.Any()) // Caso exista algum provider ativo, como
por exemplo o GPS...
    _locationProvider = acceptableLocationProviders.First(); //... o mesmo será
retornado aqui, ...
else

```

```

        _locationProvider = "network"; //... caso contrário será utilizado a rede de
        telefonia ou o roteador wi-fi para determinar a localização.

        if (_locationProvider.Length > 0)
        {
            try
            {
                _service.Url = _webServiceText.Text;
                DataSet dsDispositivo = _service.RecuperarListagem(string.Empty, "SELECT
                IMEI FROM DISPOSITIVOMOVEL WHERE IMEI = '" + _imeiText.Text + "'");
                if ((dsDispositivo.Tables.Count > 0) && (dsDispositivo.Tables[0].Rows.Count
                == 0)) // Verifica se o dispositivo existe na base de dados...
                {
                    // se não existir insere o mesmo.
                    _service.Url = _webServiceText.Text;
                    _service.InserirDispositivoMovel(string.Empty, _imeiText.Text,
                    _senhaText.Text, _nomeText.Text, _emailText.Text, "Android",
                    float.Parse(_intervaloText.Text));
                }

                var locationCriteria = new Criteria();
                locationCriteria.Accuracy = Accuracy.NoRequirement;
                locationCriteria.PowerRequirement = Power.NoRequirement;

                var geocdr = new Geocoder(this);

                // Tenta obter a localização aqui sempre, porque o provider pode ser
                "network".
                string locationProvider = _locationManager.GetBestProvider(locationCriteria,
                true);
                string lat =
                _locationManager.GetLastKnownLocation(locationProvider).Latitude.ToString();
                string lon =
                _locationManager.GetLastKnownLocation(locationProvider).Longitude.ToString();

                if (_locationManager.GetLastKnownLocation("gps") != null) // Caso tenha GPS
                ativado obtem a localização mais apurada com o endereço.
                {
                    lat = _locationManager.GetLastKnownLocation("gps").Latitude.ToString();
                    lon = _locationManager.GetLastKnownLocation("gps").Longitude.ToString();
                    _service.Url = _webServiceText.Text;
                    try
                    {
                        {
                            var addresses = geocdr.GetFromLocation(float.Parse(lat),
                            float.Parse(lon), 1);
                            _service.InserirCoordenadasDispositivoMovel(string.Empty,
                            _imeiText.Text, lat, lon, "GPS: " + addresses[0].ToString().Substring(0, 249),
                            DateTime.Now);
                            _enderecoText.Text = "GPS: " + addresses[0].ToString().Substring(0,
                            249);
                            _coordenadasText.Text = "Coordenadas: " + lat + "; " + lon;
                        }
                    }
                    catch
                    {
                        _service.InserirCoordenadasDispositivoMovel(string.Empty,
                        _imeiText.Text, lat, lon, "GPS: Não foi possível recuperar o endereço.",
                        DateTime.Now);
                        _enderecoText.Text = "GPS: Não foi possível recuperar o endereço.";
                    }
                }
            }
        }
    }
}

```

```

        _coordenadasText.Text = "Coordenadas: " + lat + "; " + lon;
    }

    ExibirMensagemNaAreaDeNotificacao(_coordenadasText.Text);
}
else
{
    // Caso não tenha GPS, tenta obter o endereço do network mesmo.
    _service.Url = _webServiceText.Text;
    try
    {
        var addresses = geocdr.GetFromLocation(double.Parse(lat,
System.Globalization.CultureInfo.CurrentCulture), double.Parse(lon,
System.Globalization.CultureInfo.CurrentCulture), 1);
        _service.InserirCoordenadasDispositivoMovel(string.Empty,
_imeiText.Text, lat, lon, "Net: " + addresses[0].ToString().Substring(0, 249),
DateTime.Now);
        _enderecoText.Text = "Net: " + addresses[0].ToString().Substring(0,
249);
        _coordenadasText.Text = "Coordenadas: " + lat + "; " + lon;
    }
    catch
    {
        _service.InserirCoordenadasDispositivoMovel(string.Empty,
_imeiText.Text, lat, lon, "Net: Não foi possível recuperar o endereço.",
DateTime.Now);
        _enderecoText.Text = "Net: Não foi possível recuperar o endereço.";
        _coordenadasText.Text = "Coordenadas: " + lat + "; " + lon;
    }

    ExibirMensagemNaAreaDeNotificacao(_coordenadasText.Text);
}
}
catch (System.Exception exp)
{
    ExibirMensagemNaAreaDeNotificacao(exp.Message.ToString());
}
}

/// <summary>
/// Este método é utilizado para exibir mensagens na área de notificações do
Dispositivo Móvel.
/// </summary>
/// <param name="mensagem">Mensagem a ser exibida.</param>
private void ExibirMensagemNaAreaDeNotificacao(string mensagem)
{
    notificationManager =
(NotificationManager)GetSystemService(Context.NotificationService);
    var sinceEpoch = DateTime.UtcNow - new DateTime(1970, 1, 1);
    var msSinceEpoch = (long)sinceEpoch.TotalMilliseconds;
    var notification = new Notification(Resource.Drawable.Icon, mensagem,
msSinceEpoch);
    notification.Defaults = NotificationDefaults.All;

    notification.ContentView = new RemoteViews(ApplicationContext.PackageName,
Resource.Layout.Notificacao);
    notification.ContentView.SetTextViewText(Resource.Id.textViewStatus, mensagem);
}

```

```

        notification.ContentView.SetTextColor(Resource.Id.textViewStatus,
R.Graphics.Color.White);

        PendingIntent contentIntent = PendingIntent.GetActivity(this, 0, null, 0);
        notification.ContentIntent = contentIntent;
        notificationManager.Notify(0, notification);
    }

    /// <summary>
    /// Método criado para recupera o IMEI do dispositivo móvel.
    /// </summary>
    /// <returns>Retorna o IMEI</returns>
    public static string RetornaImeiDispositivo()
    {
        TelephonyManager tm =
(TelephonyManager)Application.Context.GetService(Context.TelephonyService);
        string deviceIdTablet = tm.DeviceId;

        if (deviceIdTablet == null)
        {
            return
R.Provider.Settings.Secure.GetString(Application.Context.ContentResolver,
R.Provider.Settings.Secure.AndroidId);
        }
        else
        {
            return tm.DeviceId;
        }
    }
}
}
}

```

Anexo D – Código Fonte Comentado do Global Tracker Web Service

Abaixo encontra-se o código fonte comentado das principais classes do projeto GlobalTracker.WebService.

1. GeneralService.asmx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Data;
using System.Collections;

namespace GlobalTracker.WebService
{
    /// <summary>

```

```

    /// Web service utilizado para prover comunicação entre as aplicações Global Tracker
    Mobile e Global Tracker Web.
    ///
    /// Ela foi criada para satisfazer diretamente os casos de uso:
    /// - W10: Interagir com Banco de Dados da Web
    /// - M11: Interagir com Banco de Dados da Web
    ///
    /// Todos os demais casos de uso também necessitam deste Web Service indiretamente.
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment
    the following line.
    // [System.Web.Script.Services.ScriptService]
    public class GeneralService : System.Web.Services.WebService
    {

        /// <summary>
        /// Este método é utilizado para verificar se as coordenadas enviadas pelo
        dispositivo móvel ultrapassou o perímetro delimitado pela cerca eletrônica.
        /// </summary>
        /// <param name="identificador">Código ou expressão a ser validada.</param>
        /// <param name="latitude">Latitude a ser avaliada.</param>
        /// <param name="longitude">Longitude a ser avaliada</param>
        /// <param name="imei">Código universal do dispositivo</param>
        /// <returns>Retorna se o dispositivo ultrapassou o limite.</returns>
        [WebMethod]
        public bool UltrapassouCercaEletronica(string identificador, string latitude,
        string longitude, string imei)
        {
            Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess();
            try
            {
                // Estabelece conexão com o banco de dados.
                databaseAccess.Connect();
                string cercaEletronica = databaseAccess.GetDataToString("SELECT
LISTALATITUDELONGITUDE FROM CERCAELETRONICA WHERE IMEI = '" + imei + "'"); // Recupera
limites da cerca.
                string[] retornoArray = cercaEletronica.Split('#');
                float[] arrayLatitude = new float[retornoArray.Length - 1]; //Cria array para
coordenadas de Latitude. (Eixo Y)
                float[] arrayLongitude = new float[retornoArray.Length - 1]; //Cria array para
coordenadas de Longitude. (Eixo X)
                float fLatitude = System.Convert.ToSingle(latitude); // Latitude a ser
avaliada com o tipo float.
                float fLongitude = System.Convert.ToSingle(longitude); // Longitude a ser
avaliada com o tipo float.
                bool dentroDaCerca = false; // Variável de controle para determinar se esta
dentro da cerca.

                // Preenche os arrays de latitude e longitude.
                for (int i = 0; i < retornoArray.Length; i++)
                {
                    if ((retornoArray[i].ToString().Trim().Length - 1) > 0)
                    {
                        arrayLatitude[i] =
System.Convert.ToSingle(retornoArray[i].ToString().Substring(0,
retornoArray[i].ToString().IndexOf(";")).Replace(".", ","));

```

```

        arrayLongitude[i] =
System.Convert.ToSingle(retornoArray[i].ToString().Substring(retornoArray[i].ToString(
).IndexOf(";") + 1).Replace(".", ","));
    }
}

// Variavel j, utilizada para avaliar o ponto anterior do período.
int j = (arrayLatitude.Length - 1);

// Avalia se o ponto em questão esta dentro do polígono, baseado na ideia de
que a latitude do mesmo deve possuir quantidade ímpar de lados do polígono à sua
direita e esquerda.
for (int i = 0; i < arrayLatitude.Length; i++)
{
    if (arrayLatitude[i] < fLatitude && arrayLatitude[j] >= fLatitude ||
        arrayLatitude[j] < fLatitude && arrayLatitude[i] >= fLatitude)
    {
        if (arrayLongitude[i] + (fLatitude - arrayLatitude[i]) /
(arrayLatitude[j]-arrayLatitude[i]) * (arrayLongitude[j]-arrayLongitude[i]) <
fLongitude)
        {
            dentroDaCerca = !dentroDaCerca;
        }
    }
    j = i;
}
return !dentroDaCerca;
}
catch
{
    databaseAccess.Disconnect();
    databaseAccess = null;
    GC.Collect();
    return false;
}
finally
{
    databaseAccess.Disconnect();
    databaseAccess = null;
    GC.Collect();
}
}

/// <summary>
/// APENAS PARA TESTES - Este método é utilizado para verificar se as coordenadas
enviadas pelo dispositivo móvel ultrapassou o perímetro delimitado pela cerca
eletrônica.
/// </summary>
/// <param name="identificador">Código ou expressão a ser validada.</param>
/// <param name="latitude">Latitude a ser avaliada.</param>
/// <param name="longitude">Longitude a ser avaliada</param>
/// <param name="imei">Código universal do dispositivo</param>
/// <returns>Retorna string dos testes realizados.</returns>
[WebMethod]
public string UltrapassouCercaEletronicaStr(string identificador, string latitude,
string longitude, string imei)
{
    string s = string.Empty;
    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess();

```

```

try
{
    databaseAccess.Connect();
    string cercaEletronica = databaseAccess.GetDataToString("SELECT
LISTALATITUDELONGITUDE FROM CERCAELETRONICA WHERE IMEI = '" + imei + "'");
    string[] retornoArray = cercaEletronica.Split('#');
    float[] arrayLatitude = new float[retornoArray.Length - 1]; //Y
    float[] arrayLongitude = new float[retornoArray.Length - 1]; //X
    float fLatitude = System.Convert.ToSingle(latitude);
    float fLongitude = System.Convert.ToSingle(longitude);
    bool dentroDaCerca = false;

    s = "Latitude/Longitude<br>";
    s += latitude + "/" + longitude + "<br>";
    s += "Array Latitude e Longitude<br>";

    for (int i = 0; i < retornoArray.Length; i++)
    {
        if ((retornoArray[i].ToString().Trim().Length - 1) > 0)
        {
            arrayLatitude[i] =
System.Convert.ToSingle(retornoArray[i].ToString().Substring(0,
retornoArray[i].ToString().IndexOf(";")).Replace(".", ","));
            arrayLongitude[i] =
System.Convert.ToSingle(retornoArray[i].ToString().Substring(retornoArray[i].ToString(
).IndexOf(";") + 1).Replace(".", ","));

            s += arrayLatitude[i].ToString() + "/" + arrayLongitude[i].ToString() +
"<br>";
        }
    }

    int j = (arrayLatitude.Length - 1);

    s += "Testes<br>";

    for (int i = 0; i < arrayLatitude.Length; i++)
    {
        s += "if (" + arrayLatitude[i].ToString() + " < " + latitude + " && " +
arrayLatitude[j].ToString() + " >= " + latitude + " || " + arrayLatitude[j].ToString()
+ " < " + latitude + " && " + arrayLatitude[i].ToString() + " >= " + latitude +
")<br>";

        if (arrayLatitude[i] < fLatitude && arrayLatitude[j] >= fLatitude ||
arrayLatitude[j] < fLatitude && arrayLatitude[i] >= fLatitude)
        {
            s += "if (" + arrayLongitude[i].ToString() + " + (" + latitude + " - " +
arrayLatitude[i].ToString() + ") / (" + arrayLatitude[j].ToString() + " - " +
arrayLatitude[i].ToString() + ") * (" + arrayLongitude[j].ToString() + " - " +
arrayLongitude[i].ToString() + ") < " + longitude + ")<br>";
            if (arrayLongitude[i] + (fLatitude - arrayLatitude[i]) / (arrayLatitude[j]
- arrayLatitude[i]) * (arrayLongitude[j] - arrayLongitude[i]) < fLongitude)
            {
                s += "OK;<br>";
                dentroDaCerca = !dentroDaCerca;
            }
        }
    }
    j = i;
}

```

```

    }
    return s;
}
catch
{
    databaseAccess.Disconnect();
    databaseAccess = null;
    GC.Collect();
    return string.Empty;
}
finally
{
    databaseAccess.Disconnect();
    databaseAccess = null;
    GC.Collect();
}
}

/// <summary>
/// Este método será implementado para funcionar como um validador de chave
pública e chave privada,
/// para que nenhuma aplicação não autorizada consiga utilizar este Web Service.
/// </summary>
/// <param name="identificador">Código ou expressão a ser validada.</param>
/// <returns>Retorna true se correto e false se incorreto.</returns>
private bool ValidarIdentificador(string identificador)
{
    return true; // Validação não implementada, sempre retorna true.
}

/// <summary>
/// Método utilizado para realizar teste de conexão.
/// </summary>
/// <returns>Retorna true se conectado, caso contrário false.</returns>
[WebMethod]
public string TesteConexao()
{
    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;
    return databaseAccess.ConnectTest();
}

/// <summary>
/// Método utilizado para realizar autenticação de um usuário administrador.
/// </summary>
/// <param name="identificador">Chave de validação.</param>
/// <param name="login">Usuário utilizado para autenticação.</param>
/// <param name="senha">Senha utilizada para autenticação.</param>
/// <returns>Retorna true caso a autenticação seja positiva e false caso
negativa.</returns>
[WebMethod]
public bool AutenticacaoUsuarioAdministrador(string identificador, string login,
string senha)
{
    if (!ValidarIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess();

    try

```

```

    {
        databaseAccess.Connect();
        int count = 0;
        // Realiza consulta na tabela de usuário e retorna 1 caso exista usuário e 0
        caso não exista.
        int.TryParse(databaseAccess.GetDataToString("SELECT COUNT(*) FROM USUARIO
        WHERE LOGIN = '" + login + "' AND SENHA = '" + senha + "'"), out count);

        if (count == 0)
            return false;
        else
            return true;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para alterar senha do usuário Administrador.
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="login">Usuário utilizado para autenticação.</param>
/// <param name="senhaAntiga">Senha utilizada para autenticação.</param>
/// <param name="novaSenha">Nova senha que será utilizada para
autenticação.</param>
/// <returns>Retorna true caso a operação tenha obtido sucesso e false caso tenha
falhado.</returns>
[WebMethod]
public bool AlterarSenhaUsuarioAdministrador(string identificador, string login,
string senhaAntiga, string novaSenha)
{
    if (!ValidarIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

    try
    {
        databaseAccess.Connect();
        int count = 0;
        // Realiza alteração da senha e retorna 1 caso exista a alteração tenha sido
        realizada e 0 caso contrário.
        int.TryParse(databaseAccess.ExecuteCommand("UPDATE USUARIO SET SENHA = '" +
        novaSenha + "' WHERE LOGIN = '" + login + "' AND SENHA = '" + senhaAntiga + "'"), out
        count);

        if (count == 0)

```

```

        return false;
    else
        return true;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para inserir um dispositivo móvel no sistema.
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo</param>
/// <param name="senha">Senha que será utilizada para autenticação</param>
/// <param name="nome">Nome do Proprietário</param>
/// <param name="email">E-Mail do proprietário</param>
/// <param name="sistemaOperacional">Versão do sistema operacional</param>
/// <param name="intervaloMonitoramento">Intervalo para envio das
coordenadas</param>
/// <returns></returns>
[WebMethod]
public bool InserirDispostivoMovel(string identificador, string imei, string
senha, string nome, string email, string sistemaOperacional, float
intervaloMonitoramento)
{
    if (!ValidarIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

    try
    {
        databaseAccess.Connect();
        int count = 0;
        // Realiza inserção do dispositivo móvel e retorna 1 caso tenha obtido sucesso
e 0 caso contrário.
        int.TryParse(databaseAccess.ExecuteCommand("INSERT INTO DISPOSITIVOMOVEL
(IMEI, SENHA, NOME, EMAIL, SISTEMAOPERACIONAL, INTERVALOMONITORAMENTO)\r\n"+
"VALUES ('" + imei + "', '" + senha + "', '" + nome + "', '" + email + "', '"
+ sistemaOperacional + "', " + intervaloMonitoramento.ToString().Replace(",", ".") +
"')"), out count);

        if (count == 0)
            return false;
        else
            return true;
    }
    catch

```

```

    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para realizar autenticação de um dispositivo móvel.
/// </summary>
/// <param name="identificador">Chave de validação.</param>
/// <param name="login">IMEI utilizado para autenticação.</param>
/// <param name="senha">Senha utilizada para autenticação.</param>
/// <returns>Retorna true caso a autenticação seja positiva e false caso
negativa.</returns>
[WebMethod]
public bool AutenticacaoDispositivoMovel(string identificador, string imei, string
senha)
{
    if (!ValidarIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

    try
    {
        databaseAccess.Connect();
        int count = 0;
        int.TryParse(databaseAccess.GetDataToString("SELECT COUNT(*) FROM
DISPOSITIVOMOVEL WHERE IMEI = '" + imei + "' AND SENHA = '" + senha + "'"), out
count);

        if (count == 0)
            return false;
        else
            return true;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }

    return true;
}

```

```

}

/// <summary>
/// Método utilizado para alterar senha de um dispositivo móvel.
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo móvel.</param>
/// <param name="senhaAntiga">Senha utilizada para autenticação.</param>
/// <param name="novaSenha">Nova senha que será utilizada para
autenticação.</param>
/// <returns>Retorna true caso a operação tenha obtido sucesso e false caso tenha
falhado.</returns>
[WebMethod]
public bool AlterarSenhaDispositivoMovel(string identificador, string imei, string
senhaAntiga, string novaSenha)
{
    if (!ValidarIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

    try
    {
        databaseAccess.Connect();
        int count = 0;
        int.TryParse(databaseAccess.ExecuteCommand("UPDATE DISPOSITIVOMOVEL SET SENHA
= '' + novaSenha + '' WHERE IMEI = '' + imei + '' AND SENHA = '' + senhaAntiga + ''"),
out count);

        if (count == 0)
            return false;
        else
            return true;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para enviar as coordenadas geográficas e endereço para o
banco de dados.
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo móvel.</param>
/// <param name="latitude">Latitude obtida</param>
/// <param name="longitude">Longitude obtida</param>
/// <param name="endereco">Endereço corresponde à coordenada</param>
/// <param name="dataCoordenada">Data da coleta</param>

```

```

    /// <returns>Retorna true caso a operação tenha obtido sucesso e false caso tenha
    falhado.</returns>
    [WebMethod]
    public bool InserirCoordenadasDispositivoMovel(string identificador, string imei,
    string latitude, string longitude, string endereco, DateTime dataCoordenada)
    {
        if (!ValidarIdentificador(identificador))
            return false;

        Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

        try
        {
            databaseAccess.Connect();
            int count = 0;
            int.TryParse(databaseAccess.ExecuteCommand("INSERT INTO
COORDENADADISPOSITIVOMOVEL (IMEI, LATITUDE, LONGITUDE, ENDERECO, DATA)\r\n" +
                "VALUES ('" + imei + "', " + latitude.ToString().Replace(",", ".") + ", " +
                longitude.ToString().Replace(",", ".") + ", '" + endereco + "', GETDATE())"), out
            count);

            if(UltrapassouCercaEletronica(identificador, latitude, longitude, imei))
            {
                string emailOrigem =
                System.Configuration.ConfigurationManager.AppSettings["EmailOrigem"].ToString();
                string loginEmail =
                System.Configuration.ConfigurationManager.AppSettings["UsuarioEmail"].ToString();
                string senhaEmail =
                System.Configuration.ConfigurationManager.AppSettings["SenhaEmail"].ToString();
                string servidorEmail =
                System.Configuration.ConfigurationManager.AppSettings["ServidorSMTP"].ToString();
                string portaServidorEmail =
                System.Configuration.ConfigurationManager.AppSettings["PortaSMTP"].ToString();
                string emailDestinatario = databaseAccess.GetDataToString("SELECT EMAIL FROM
DISPOSITIVOMOVEL WHERE IMEI = '" + imei + "'");

                Core.Email mail = new Core.Email();
                mail.HasAuthentication = true;
                mail.EnableSSL = true;
                mail.SendMail("Alerta de violação de cerca eletrônica", "O dispositivo móvel
                identificado pelo IMEI " + imei + " ultrapassou o limite determinado para cerca
                eletrônica.<br>- Latitude: " + latitude + "<br>- Longitude: " + longitude,emailOrigem,
                emailDestinatario, servidorEmail, portaServidorEmail, loginEmail, senhaEmail);

                databaseAccess.ExecuteCommand("INSERT INTO HISTORICOALERTA (IMEI, DATA,
                LATITUDE, LONGITUDE, ENDERECO) VALUES ('" + imei + "', GETDATE(), " +
                latitude.ToString().Replace(",", ".") + ", " + longitude.ToString().Replace(",", ".") + ", '" + endereco +
                "'");
            }

            if (count == 0)
                return false;
            else
                return true;
        }
        catch
        {
            databaseAccess.Disconnect();
        }
    }
}

```

```

        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para persistir as configurações de cerca eletrônica no banco
de dados
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo móvel</param>
/// <param name="permitePerimetroCerca">Consistir a permanência do dispositivo
móvel dentro do perímetro</param>
/// <param name="listaLatitudeLongitude">Lista de coordenadas geográficas que
determinam o perímetro</param>
/// <returns>Retorna true caso a operação tenha obtido sucesso e false caso tenha
falhado.</returns>
[WebMethod]
public bool InserirCercaEletronica(string identificador, string imei, bool
permitePerimetroCerca, string listaLatitudeLongitude)
{
    if (!ValidarIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;
    string perimetro = "Sim";

    if (!permitePerimetroCerca)
        perimetro = "Não";

    try
    {
        databaseAccess.Connect();
        int count = 0;

        databaseAccess.ExecuteNonQuery("DELETE FROM CERCAELETRONICA WHERE IMEI = '"+
imei + "'");
        int.TryParse(databaseAccess.ExecuteNonQuery("INSERT INTO CERCAELETRONICA (IMEI,
PERMITIRPERIMETROCERCA, LISTALATITUDELONGITUDE)\r\n" +
"VALUES ('" + imei + "', '" + perimetro + "', '" + listaLatitudeLongitude +
"'"), out count);

        if (count == 0)
            return false;
        else
            return true;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

```

```

        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para recuperar as coordenadas correspondentes a uma cerca
eletrônica.
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo móvel</param>
/// <returns>Retorna um Dataset contendo as coordenadas geográficas
correspondentes à cerca eletrônica</returns>
[WebMethod]
public string[] RecuperarCercaEletronica(string identificador, string imei)
{
    if (!ValidarIdentificador(identificador))
        return null;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess();

    try
    {
        databaseAccess.Connect();
        string retornoString = databaseAccess.GetDataToString("SELECT
LISTALATITUDELONGITUDE FROM CERCAELETRONICA WHERE IMEI = '" + imei + "'");
        string[] retornoArray = retornoString.Split('#');

        return retornoArray;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return null;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para excluir uma cerca eletrônica
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo móvel</param>
/// <returns>Retorna true caso a operação tenha obtido sucesso e false caso tenha
falhado.</returns>
[WebMethod]
public bool ExcluirCercaEletronica(string identificador, string imei)

```

```

{
    if (!ValidadorIdentificador(identificador))
        return false;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

    try
    {
        databaseAccess.Connect();
        int count = 0;

        int.TryParse(databaseAccess.ExecuteCommand("DELETE INTO CERCAELETRONICA WHERE
IMEI = '" + imei + '""), out count);

        if (count == 0)
            return false;
        else
            return true;
    }
    catch
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
        return false;
    }
    finally
    {
        databaseAccess.Disconnect();
        databaseAccess = null;
        GC.Collect();
    }
}

/// <summary>
/// Método utilizado para retornar as coordenadas geográficas conforme os
parâmetros passados.
/// </summary>
/// <param name="identificador">Chave de Validação</param>
/// <param name="imei">Código universal do dispositivo móvel</param>
/// <param name="quantidade">Quantidade de coletas a serem recuperadas</param>
/// <returns>Retorna um DataSet contendo as coletas.</returns>
[WebMethod]
public DataSet RecuperarUltimasCoordenadasDispositivoMovel(string identificador,
string imei, int quantidade)
{
    if (!ValidadorIdentificador(identificador))
        return null;

    Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

    try
    {
        databaseAccess.Connect();
        DataSet ds = databaseAccess.GetDataSet("SELECT TOP " + quantidade + " DATA,
LATITUDE, LONGITUDE FROM COORDENADADISPOSITIVOMOVEL WHERE IMEI = '" + imei + "' ORDER
BY DATA DESC");
        return ds;
    }
}

```

```

        catch
        {
            databaseAccess.Disconnect();
            databaseAccess = null;
            GC.Collect();
            return null;
        }
        finally
        {
            databaseAccess.Disconnect();
            databaseAccess = null;
            GC.Collect();
        }
    }

    /// <summary>
    /// Método utilizado para retornar as coordenadas geográficas conforme os
    parâmetros passados.
    /// </summary>
    /// <param name="identificador">Chave de Validação</param>
    /// <param name="imei">Código universal do dispositivo móvel</param>
    /// <param name="dataInicial">Data de início das coordenadas</param>
    /// <param name="dataFinal">>Data de fim das coordenadas</param>
    /// <returns>Retorna um DataSet contendo as coletas.</returns>
    [WebMethod]
    public DataSet RecuperarCoordenadasPorPeriodoDispositivoMovel(string
    identificador, string imei, DateTime dataInicial, DateTime dataFinal)
    {
        if (!ValidarIdentificador(identificador))
            return null;

        Core.DatabaseAccess databaseAccess = new Core.DatabaseAccess(); ;

        try
        {
            databaseAccess.Connect();
            DataSet ds = databaseAccess.GetDataSet("SELECT DATA, LATITUDE, LONGITUDE FROM
            COORDENADADISPOSITIVOMOVEL WHERE IMEI = '" + imei + "' AND DATA BETWEEN #" +
            dataInicial + "# AND #" + dataFinal + "# ORDER BY DATA DESC");
            return ds;
        }
        catch
        {
            databaseAccess.Disconnect();
            databaseAccess = null;
            GC.Collect();
            return null;
        }
        finally
        {
            databaseAccess.Disconnect();
            databaseAccess = null;
            GC.Collect();
        }
    }

    /// <summary>
    /// Método utilizado para realizar consultas no banco de dados Microsoft SQL
    Server

```



```

{
    // Objeto para estabelecer conexão com o banco de dados Microsoft SQL Server.
    private System.Data.SqlClient.SqlConnection connection;

    /// <summary>
    /// Construtor.
    /// </summary>
    public DatabaseAccess()
    {
    }

    /// <summary>
    /// Método responsável por realizar a conexão com o banco de dados.
    /// </summary>
    /// <returns>true = Conectou / false = Erro de Conexão</returns>
    public bool Connect()
    {
        // Recupera a string de conexão que contém os parametros para estabelece conexão
        // com o banco de dados.
        // A definição da mesma encontra-se no arquivo Web.Config deste Web Service.
        string strConnection =
System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].Connec
tionString;

        // Instancia e abre conexão.
        try
        {
            connection = new System.Data.SqlClient.SqlConnection(strConnection);
            connection.Open();
            return true;
        }
        catch (Exception exp)
        {
            connection.Dispose();
            throw exp;
        }
    }

    /// <summary>
    /// Método responsável por realizar um teste de conexão com o banco de dados.
    /// </summary>
    /// <returns>true = Conectou / false = Erro de Conexão</returns>
    public string ConnectTest()
    {
        //// Recupera a string de conexão que contém os parametros para estabelece
        // conexão com o banco de dados.
        // A definição da mesma encontra-se no arquivo Web.Config deste Web Service.
        string strConnection =
System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].Connec
tionString;
        //System.Configuration.ConfigurationManager.AppSettings["ConnectionString"].ToLower();

        // Instancia e abre conexão.
        try
        {
            connection = new System.Data.SqlClient.SqlConnection(strConnection);
            connection.Open();
            return "ok"; // Se conectou sem erros retorna a string "ok".
        }
    }
}

```

```

        catch (Exception exp)
        {
            connection.Dispose();
            return exp.Message; // Se houve erros retorna uma string com a mensagem de
erro.
        }
    }

    /// <summary>
    /// Método responsável por realizar a desconexão do banco de dados.
    /// </summary>
    /// <returns>true = Desconectou / false = Não conseguiu desconectar</returns>
    public bool Disconnect()
    {
        // Fecha e libera objeto de conexão.
        try
        {
            connection.Close();
            connection.Dispose();
            return true;
        }
        catch (Exception exp) // Recupera exceção.
        {
            connection.Dispose();
            throw exp; // Levanta a exceção apresentada.
        }
    }

    /// <summary>
    /// Método responsável por carregar e retornar um DataSet com a consulta passada.
    /// </summary>
    /// <param name="comand">Comando SQL</param>
    /// <returns>DataSet com o resultado do comando SQL. Retorna null se ocorreu
erro.</returns>
    public DataSet GetDataSet(string command)
    {
        // Verifica se foi passado um comando.
        if (command.Trim().Length == 0)
            return null; // Retorna nulo caso não tenha sido passado um comando.

        // Instancia um objeto do tipo DataSet, que irá receber os resultados da query
executada.
        DataSet dataSet = new DataSet();

        try
        {
            // Instancia um Data Adapter, que será utilizado para preencher o DataSet.
            IDbDataAdapter dataAdapter = new System.Data.SqlClient.SqlDataAdapter();
            this.Connect(); // Conecta ao banco de dados.

            if (connection != null) // Se o objeto de conexão não for nulo.
            {
                // Atribui query que será executada ao Data Adapter.
                dataAdapter.SelectCommand = new System.Data.SqlClient.SqlCommand(command,
connection);
                // Preenche o DataSet com o resultado da query.
                dataAdapter.Fill(dataSet);
                // Desconecta do banco de dados.
                this.Disconnect();
            }
        }
    }

```

```

    }
}
catch
{
    // Libera o DataSet caso tenha ocorrido algum erro na execução e desconecta do
banco de dados.
    dataSet = null;
    this.Disconnect();
}

// Retorna o resultado de execução da query ou nulo caso tenha ocorrido algum
erro.
return dataSet;
}

/// <summary>
/// Método responsável por executar uma query de Insert, Delete ou Update.
/// </summary>
/// <param name="command">Comando SQL</param>
/// <returns>Retorna uma string com a quantidade de registros afetados ou "Error"
caso tenha ocorrido erro.</returns>
public string ExecuteCommand(string command)
{
    // Verifica se foi passado um comando.
    if (command.Trim().Length == 0)
        return string.Empty; // Retorna uma string vazia caso não tenha sido passado
um comando.

    // Declara objeto que será utilizado para executar uma query DML.
    System.Data.SqlClient.SqlCommand sqlCommand = null;

    try
    {
        this.Connect(); // Conecta ao banco de dados.
        if (connection != null) // Se o objeto de conexão não for nulo.
        {
            // Atribui query ao SqlCommand.
            sqlCommand = new System.Data.SqlClient.SqlCommand(command, connection);
            // Retorna resultado da execução do comando.
            string resultado = sqlCommand.ExecuteNonQuery().ToString();
            // Libera SqlCommand.
            sqlCommand.Dispose();
            // Desconecta do banco de dados.
            this.Disconnect();
            // Retorna resultado.
            return resultado;
        }
        else
        {
            return "Não existe conexão com o banco de dados.";
        }
    }
    catch (Exception exp)
    {
        // Libera SqlCommand.
        sqlCommand.Dispose();
        // Deconecta do banco de dados.
        this.Disconnect();
        // Retorna mensagem de erro durante o processo de execução da query.
    }
}

```

```

        return "Erro: \r\n" + exp.Message.ToString();
    }
}

/// <summary>
/// Método responsável por retornar a primeira coluna da primeira linha da
/// consulta, ideal para consultas rápidas como count, sum, etc...
/// </summary>
/// <param name="command">Comando SQL</param>
/// <returns>Valor retornado pela consulta ou "Erro" caso tenha ocorrido erro.
</returns>
public string GetDataToString(string command)
{
    // Verifica se foi passado um comando.
    if (command.Trim().Length == 0)
        return null; // Retorna nulo caso não tenha sido passado um comando.

    // Instancia um objeto do tipo DataSet, que irá receber os resultados da query
    executada.
    DataSet dataSet = new DataSet();

    try
    {
        // Instancia um Data Adapter, que será utilizado para preencher o DataSet.
        IDbDataAdapter dataAdapter = new System.Data.SqlClient.SqlDataAdapter();
        this.Connect(); // Conecta ao banco de dados.

        if (connection != null) // Se o objeto de conexão não for nulo.
        {
            // Atribui query que será executada ao Data Adapter.
            dataAdapter.SelectCommand = new System.Data.SqlClient.SqlCommand(command,
connection);
            // Preenche o DataSet com o resultado da query.
            dataAdapter.Fill(dataSet);
            // Desconecta do banco de dados.
            this.Disconnect();
            // Se a execução da query retornou uma tabela com pelo menos uma linha...
            if ((dataSet.Tables.Count > 0) && (dataSet.Tables[0].Rows.Count > 0))
            {
                // ... retorna o valor da primeira coluna e primeira linha.
                return dataSet.Tables[0].Rows[0][0].ToString();
            }
            else
            {
                // ... retorna uma string vazia.
                return string.Empty;
            }
        }
        else
        {
            return string.Empty;
        }
    }
    catch
    {
        // Libera o DataSet caso tenha ocorrido algum erro na execução e desconecta do
        banco de dados.
        dataSet = null;
        this.Disconnect();
    }
}

```

```

        return string.Empty;
    }
}

/// <summary>
/// Propriedade da classe que retorna conexão corrente.
/// </summary>
public System.Data.SqlClient.SqlConnection CurrentConnection
{
    get
    {
        return connection;
    }
}
}
}

```

3. EMail.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Net.Mail;
using System.Net;

namespace GlobalTracker.WebService.Core
{
    /// <summary>
    /// Esta classe é responsável pelo envio de e-mails de alerta para os responsáveis
    por cada dispositivo móvel
    /// que ultrapassar os limites de uma cerca eletrônica.
    ///
    /// Ela foi criada para satisfazer diretamente os casos de uso:
    /// - M09: Obter coordenadas geográficas
    /// </summary>
    public class EMail
    {
        private bool access; // Variável que determina se houve sucesso na autenticação
        com o servidor.
        private bool hasAuthentication; // Variável que determina se é necessário fazer
        uma autenticação explícita.
        private bool enableSSL; // Variável que determina se utilizará SSL.
        private MailMessage message; // Variável que define o objeto de e-mail, contendo
        assunto, mensagem, etc...
        private SmtplibClient smtpClient; // Objeto para acesso ao servidor de SMTP.
        private NetworkCredential networkCredential; // Objeto de autenticação para envio
        do e-mail.

        /// <summary>
        /// Construtor.
        /// </summary>
        public EMail()
        {
            access = true;
            message = new MailMessage();
            smtpClient = new SmtplibClient();
        }
    }
}

```

```

    networkCredential = new NetworkCredential();
}

/// <summary>
/// Propriedade que retorna se houve sucesso na autenticação.
/// </summary>
public bool Access
{
    get
    {
        return access;
    }
    set
    {
        access = value;
    }
}

/// <summary>
/// Propriedade utilizada para determinar o uso do SSL.
/// </summary>
public bool EnableSSL
{
    get { return enableSSL; }
    set { enableSSL = value; }
}

/// <summary>
/// Propriedade utilizada para determinar se haverá autenticação ou não.
/// </summary>
public bool HasAuthentication
{
    get { return hasAuthentication; }
    set { hasAuthentication = value; }
}

/// <summary>
/// Define o e-mail de origem
/// </summary>
/// <param name="mail"></param>
public void SetFrom(string mail)
{
    try
    {
        message.From = new MailAddress(mail);
    }
    catch
    {
        if (access)
        {
            access = false;
            return;
        }
    }
}

/// <summary>
/// Define e-mails de destino.
/// </summary>

```

```

/// <param name="mails"></param>
public void SetTo(string mails)
{
    try
    {
        message.To.Clear();
        mails.Replace(" ", "");
        int i = 0;
        while (i < mails.Length)
        {
            if (mails[i] == ';')
            {
                string temp = mails.Substring(0, i);
                message.To.Add(temp);
                mails = mails.Remove(0, i + 1);
                i = 0;
            }
            else
            {
                if (i == mails.Length - 1)
                {
                    string temp = mails.Substring(0, i + 1);
                    message.To.Add(temp);
                    mails = mails.Remove(0, i + 1);
                    i = 0;
                }
                else
                {
                    i++;
                }
            }
        }
    }
    catch
    {
        if (access)
        {
            access = false;
            return;
        }
    }
}

/// <summary>
/// Define e-mails de cópia.
/// </summary>
/// <param name="mails"></param>
public void SetCC(string mails)
{
    try
    {
        message.CC.Clear();
        mails.Replace(" ", "");
        int i = 0;
        while (i < mails.Length)
        {
            if (mails[i] == ';')
            {
                message.CC.Add(mails.Substring(0, i));
                mails = mails.Remove(0, i + 1);
                i = 0;
            }
            else

```

```

        if (i == mails.Length - 1)
        {
            message.CC.Add(mails.Substring(0, i + 1));
            mails = mails.Remove(0, i + 1);
            i = 0;
        }
        else
            i++;
    }
}
catch
{
    if (access)
    {
        access = false;
        return;
    }
}
}

/// <summary>
/// Método para envio de e-mail.
/// </summary>
public void SendMail()
{
    try
    {
        message.IsBodyHtml = true;
        smtpClient.EnableSsl = enableSSL;

        if (hasAuthentication)
        {
            smtpClient.Credentials = networkCredential;
            smtpClient.Send(message);
        }
        else
        {
            smtpClient.Send(message);
        }
    }
    catch (SmtpFailedRecipientException sfre)
    {
        access = false;
        throw new Exception("Verifique as permissões de acesso ao servidor ou nome do
usuário e senha.\r\n" + sfre.Message);
    }
    catch (SmtpException se)
    {
        access = false;
        throw new Exception("Erro ao enviar e-mail, verifique as configurações do
servidor SMTP e a porta.\r\n" + se.Message);
    }
    catch (Exception exp)
    {
        throw new Exception(exp.Message);
    }
}

/// <summary>

```

```

/// Método para envio de e-mail.
/// </summary>
/// <param name="subject">Assunto do e-mail</param>
/// <param name="body">Mensagem do e-mail</param>
/// <param name="from">Endereço de origem do e-mail</param>
/// <param name="to">Endereço de destino do e-mail</param>
/// <param name="smtpServer">Servidor de SMTP utilizado para o envio do e-
mail</param>
/// <param name="port">Porta utilizada para o envio do e-mail</param>
/// <param name="userName">Usuário para autenticação no servidor de SMTP</param>
/// <param name="password">Senha para autenticação no servidor de SMTP</param>
/// <returns>Retorna true se obteve sucesso no envio e false caso não.</returns>
public bool SendMail(string subject, string body, string from, string to, string
smtpServer, string port, string userName, string password)
{
    networkCredential.UserName = userName;
    networkCredential.Password = password;
    return this.SendMail(subject, body, from, to, smtpServer, port);
}

/// <summary>
/// Método para envio de e-mail.
/// </summary>
/// <param name="subject">Assunto do e-mail</param>
/// <param name="body">Mensagem do e-mail</param>
/// <param name="from">Endereço de origem do e-mail</param>
/// <param name="to">Endereço de destino do e-mail</param>
/// <param name="smtpServer">Servidor de SMTP utilizado para o envio do e-
mail</param>
/// <param name="port">Porta utilizada para o envio do e-mail</param>
/// <returns>Retorna true se obteve sucesso no envio e false caso não.</returns>
public bool SendMail(string subject, string body, string from, string to, string
smtpServer, string port)
{
    bool send = true;

    try
    {
        message.Subject = subject;
        message.Body = body;
        SetFrom(from);
        SetTo(to);

        smtpClient.Host = smtpServer;
        smtpClient.Port = Convert.ToInt32(port);
        SendMail();
    }
    catch (Exception exp)
    {
        if (access)
        {
            if (smtpServer == "" || port == "")
                throw new Exception("Deve se informar o servidor SMTP e a porta!");
            else
                throw new Exception(exp.Message);

            access = false;
            send = false;
        }
    }
}

```

```

    }
  }
  return send;
}
}
}

```

Anexo E – Código Fonte Comentado do Global Tracker Web

Abaixo encontra-se o código fonte comentado das principais classes do projeto GlobalTracker.Web.

1. DispositivosMoveis.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

namespace GlobalTracker.WebSite.AreaRestrita
{
    public partial class DispositivosMoveis : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                WebServiceReference.GeneralService service = new
                WebServiceReference.GeneralService();
                service.Url =
                GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceReference_GeneralService;

                DataSet ds;

                if (Session["TipoLogin"] == "Administrador")
                    ds = service.RecuperarListagem(string.Empty, "SELECT CODIGO \"Código\",
                    IMEI, SENHA \"Senha\", NOME \"Nome\", EMAIL \"E-Mail\", SISTEMAOPERACIONAL \"Sistema
                    Operacional\", INTERVALOMONITORAMENTO \"Intervalo\" FROM DISPOSITIVOMOVEL");
                else
                    ds = service.RecuperarListagem(string.Empty, "SELECT CODIGO \"Código\",
                    IMEI, SENHA \"Senha\", NOME \"Nome\", EMAIL \"E-Mail\", SISTEMAOPERACIONAL \"Sistema
                    Operacional\", INTERVALOMONITORAMENTO \"Intervalo\" FROM DISPOSITIVOMOVEL WHERE IMEI =
                    '" + Session["Login"].ToString() + "'");

                ASPxGridViewDispositivosMoveis.DataSource = ds.Tables[0];
                ASPxGridViewDispositivosMoveis.DataBind();

                service.Dispose();
                service = null;
            }
        }
    }
}

```

```

    }
  }
}

```

2. HistoricoAlerta.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

namespace GlobalTracker.WebSite.AreaRestrita
{
    public partial class HistoricoAlerta : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                WebServiceReference.GeneralService service = new
                WebServiceReference.GeneralService();
                service.Url =
                GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefer
                ence_GeneralService;

                DataSet ds;

                if (Session["TipoLogin"] == "Administrador")
                    ds = service.RecuperarListagem(string.Empty, "SELECT IMEI, DATA \"Data\",
                    LATITUDE \"Latitude\", LONGITUDE \"Longitude\" FROM HISTORICOALERTA ORDER BY 1, 2");
                else
                    ds = service.RecuperarListagem(string.Empty, "SELECT IMEI, DATA \"Data\",
                    LATITUDE \"Latitude\", LONGITUDE \"Longitude\" FROM HISTORICOALERTA WHERE IMEI = '" +
                    Session["Login"].ToString() + "' ORDER BY 1, 2");

                ASPxGridViewDispositivosMoveis.DataSource = ds.Tables[0];
                ASPxGridViewDispositivosMoveis.DataBind();

                service.Dispose();
                service = null;
            }
        }
    }
}

```

3. Index.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

using GlobalTracker.WebSite.Core;

namespace GlobalTracker.WebSite.AreaRestrita
{
    public partial class Index : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (IsPostBack)
            {
                if (TextBoxLogin.Text.Trim().Length == 0)
                {
                    LabelStatus.Text = "Informe o IMEI ou Administrador para realizar a
autenticação.";
                    return;
                }
                else
                {
                    if (TextBoxPassword.Text.Trim().Length == 0)
                    {
                        LabelStatus.Text = "Informe a Senha para realizar a autenticação.";
                        return;
                    }
                    else
                    {
                        WebServiceReference.GeneralService service = new
WebServiceReference.GeneralService();
                        service.Url =
GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefe
rence_GeneralService;
                        if (service.AutenticacaoUsuarioAdministrador(string.Empty,
TextBoxLogin.Text, TextBoxPassword.Text))
                        {
                            Session["TipoLogin"] = "Administrador";
                            Session["Login"] = TextBoxLogin.Text;
                            Session["Senha"] = TextBoxPassword.Text;

                            Response.Redirect("Principal.aspx");
                        }
                        else
                        {
                            if (service.AutenticacaoDispositivoMovel(string.Empty,
TextBoxLogin.Text, TextBoxPassword.Text))
                            {
                                Session["TipoLogin"] = "Dispositivo Móvel";
                                Session["Login"] = TextBoxLogin.Text;
                                Session["Senha"] = TextBoxPassword.Text;

                                Response.Redirect("Principal.aspx");
                            }
                            else
                            {
                                LabelStatus.Text = "Login ou Senha incorretos.";
                            }
                        }
                    }
                    service.Dispose();
                    service = null;
                }
            }
        }
    }
}

```

```

    }
  }
}

```

4. Security.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;

namespace GlobalTracker.WebSite.Core
{
    public class Security
    {
        public Security()
        {

        }

        public void IsAuthenticated(Page _page)
        {
            if (_page.Session["TipoLogin"] == null)
                _page.Response.Redirect("index.aspx");
        }
    }
}

```

5. CercaEletronica.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Subgurim.Controles;
using System.Data;

namespace GlobalTracker.WebSite.AreaRestrita
{
    public partial class CercaEletronica : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                WebServiceReference.GeneralService service = new
                WebServiceReference.GeneralService();
                service.Url =
                GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefe
                rence_GeneralService;
                string s = service.TesteConexao();
                DataSet ds;
            }
        }
    }
}

```

```

        if (Session["TipoLogin"] == "Administrador")
            ds = service.RecuperarListagem(string.Empty, "SELECT IMEI, NOME FROM
DISPOSITIVOMOVEL");
        else
            ds = service.RecuperarListagem(string.Empty, "SELECT IMEI, NOME FROM
DISPOSITIVOMOVEL WHERE IMEI = '" + Session["Login"].ToString() + "'");

        if (ds.Tables.Count > 0)
        {
            for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
            {
                ComboBoxDispositivoMovel.Items.Add(ds.Tables[0].Rows[i]["IMEI"].ToString()
+ " - " + ds.Tables[0].Rows[i]["NOME"].ToString(),
                ds.Tables[0].Rows[i]["IMEI"].ToString());
            }

            if (ds.Tables[0].Rows.Count > 0)
            {
                ComboBoxDispositivoMovel.SelectedIndex = 0;

                string[] arrayCoordenadas = service.RecuperarCercaEletronica(string.Empty,
                ComboBoxDispositivoMovel.SelectedItem.Value.ToString());

                if (arrayCoordenadas.Length > 0)
                {
                    List<GLatLng> pontosOriginal = new List<GLatLng>();
                    List<GLatLng> pontos = new List<GLatLng>();
                    string ss = string.Empty;
                    for (int i = 0; i < arrayCoordenadas.Length; i++)
                    {
                        string[] latLong = arrayCoordenadas[i].Split(';');
                        if (latLong.Length > 1)
                        {
                            pontosOriginal.Add(new
GLatLng(System.Convert.ToDouble(latLong[0].ToString()),
System.Convert.ToDouble(latLong[1].ToString())));
                        }
                    }

                    Response.Write(ss);
                    if (pontosOriginal.Count >= 3)
                    {
                        LimparAreaDefinida();
                        Session["PontosOriginal"] = pontosOriginal;

                        for (int i = 0; i < pontosOriginal.Count; i++)
                        {
                            pontos.Add(pontosOriginal[i]);
                        }

                        GPolygon poligono = new GPolygon(pontos, "557799", 3, 0.5, "237464",
0.5);

                        poligono.close();
                        GMapAreaMonitoramento.Add(poligono);
                    }
                }
            }
        }
    }
}

```

```

        else
        {
            LimparAreaDefinida();
        }
    }
}

ds.Dispose();
ds = null;
service.Dispose();
service = null;
}
}

protected string GMapAreaMonitoramento_Click(object s, GAjaxServerEventArgs e)
{
    if (CheckBoxAtivarMarcacaoArea.Checked)
    {
        List<GLatLng> pontos = new List<GLatLng>();
        List<GLatLng> pontosOriginal;
        if (Session["PontosOriginal"] == null)
        {
            pontosOriginal = new List<GLatLng>();
            pontosOriginal.Add(e.point);
            Session["PontosOriginal"] = pontosOriginal;
        }
        else
        {
            pontosOriginal = (List<GLatLng>)Session["PontosOriginal"];
            pontosOriginal.Add(e.point);
            Session["PontosOriginal"] = pontosOriginal;
        }

        if (pontosOriginal.Count >= 3)
        {
            LimparAreaDefinida();
            Session["PontosOriginal"] = pontosOriginal;

            for (int i = 0; i < pontosOriginal.Count; i++)
            {
                pontos.Add(pontosOriginal[i]);
            }

            GPolygon poligono = new GPolygon(pontos, "557799", 3, 0.5, "237464", 0.5);
            poligono.close();
            GMapAreaMonitoramento.Add(poligono);
        }
    }
    return default(string);
}

private void LimparAreaDefinida()
{
    try
    {
        GMapAreaMonitoramento.resetPolygon();
    }
    catch { }
}

```

```

    try
    {
        GMapAreaMonitoramento.resetControls();
    }
    catch { }

    try
    {
        GMapAreaMonitoramento.resetPolygon();
    }
    catch { }

    Session["PontosOriginal"] = null;
}

protected void ButtonLimpar_Click(object sender, EventArgs e)
{
    LimparAreaDefinida();
}

protected void ButtonSalvar_Click(object sender, EventArgs e)
{
    if (Session["PontosOriginal"] != null)
    {
        List<GLatLng> pontos = new List<GLatLng>();
        pontos = (List<GLatLng>)Session["PontosOriginal"];

        if (pontos.Count > 0)
        {
            string coordenadas = string.Empty;

            for (int i = 0; i < pontos.Count; i++)
            {
                coordenadas += pontos[i].lat.ToString() + ";" + pontos[i].lng.ToString() +
"#";
            }

            WebServiceReference.GeneralService service = new
WebServiceReference.GeneralService();
            service.Url =
GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefe
rence_GeneralService;
            service.InserirCercaEletronica(string.Empty,
ComboBoxDispositivoMove1.SelectedItem.Value.ToString(), true, coordenadas);
        }
    }
}

protected void ComboBoxDispositivoMove1_SelectedIndexChanged(object sender,
EventArgs e)
{
    WebServiceReference.GeneralService service = new
WebServiceReference.GeneralService();
    service.Url =
GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefe
rence_GeneralService;

```



```

        if (!IsPostBack)
        {
            WebServiceReference.GeneralService service = new
            WebServiceReference.GeneralService();
            service.Url =
            GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefer
            ence_GeneralService;

            DataSet ds;

            if (Session["TipoLogin"] == "Administrador")
                ds = service.RecuperarListagem(string.Empty, "SELECT IMEI, NOME FROM
            DISPOSITIVOMOVEL");
            else
                ds = service.RecuperarListagem(string.Empty, "SELECT IMEI, NOME FROM
            DISPOSITIVOMOVEL WHERE IMEI = '" + Session["Login"].ToString() + "'");

            if (ds.Tables.Count > 0)
            {
                for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                {
                    ComboBoxDispositivoMovel.Items.Add(ds.Tables[0].Rows[i]["IMEI"].ToString()
            + " - " + ds.Tables[0].Rows[i]["NOME"].ToString(),
                    ds.Tables[0].Rows[i]["IMEI"].ToString());
                }

                if (ds.Tables[0].Rows.Count > 0)
                {
                    ComboBoxDispositivoMovel.SelectedIndex = 0;

                    CarregarMapa(service);
                }
            }

            service.Dispose();
            service = null;
        }
    }

    private void CarregarMapa(WebServiceReference.GeneralService service)
    {
        LimparAreaDefinida();
        DataSet dsCoordenadas =
        service.RecuperarUltimasCoordenadasDispositivoMovel(string.Empty,
        ComboBoxDispositivoMovel.SelectedItem.Value.ToString(),
        int.Parse(ASPxComboBoxColetas.SelectedItem.Value.ToString()));

        if (dsCoordenadas.Tables.Count > 0)
        {
            if (dsCoordenadas.Tables[0].Rows.Count == 1)
            {
                GMarker coordenada = new GMarker(new
            GLatLng(Convert.ToDouble(dsCoordenadas.Tables[0].Rows[0]["LATITUDE"].ToString()),
            Convert.ToDouble(dsCoordenadas.Tables[0].Rows[0]["LONGITUDE"].ToString())));
                GInfoWindow janela = new GInfoWindow(coordenada, "<center>" +
            dsCoordenadas.Tables[0].Rows[0]["DATA"].ToString() + "</center>", true);
                GMapAreaMonitoramento.setCenter(new
            GLatLng(Convert.ToDouble(dsCoordenadas.Tables[0].Rows[0]["LATITUDE"].ToString()),
            Convert.ToDouble(dsCoordenadas.Tables[0].Rows[0]["LONGITUDE"].ToString())));
            }
        }
    }

```

```

        GMapAreaMonitoramento.Add(janela);
    }
    else
    {
        List<GLatLng> pontos = new List<GLatLng>();

        for (int j = 0; j < dsCoordenadas.Tables[0].Rows.Count; j++)
        {
            if (j == 0)
                GMapAreaMonitoramento.setCenter(new
GMapAreaMonitoramento.GLatLng(Convert.ToDouble(dsCoordenadas.Tables[0].Rows[j]["LATITUDE"].ToString()),
Convert.ToDouble(dsCoordenadas.Tables[0].Rows[j]["LONGITUDE"].ToString())));

            pontos.Add(new
GMapAreaMonitoramento.GLatLng(Convert.ToDouble(dsCoordenadas.Tables[0].Rows[j]["LATITUDE"].ToString()),
Convert.ToDouble(dsCoordenadas.Tables[0].Rows[j]["LONGITUDE"].ToString())));
        }

        GPolyline linha = new GPolyline(pontos, "FF0000", 1);
        GMapAreaMonitoramento.Add(linha);
    }
}

private void LimparAreaDefinida()
{
    try
    {
        GMapAreaMonitoramento.resetPolygon();
    }
    catch { }

    try
    {
        GMapAreaMonitoramento.resetControls();
    }
    catch { }

    try
    {
        GMapAreaMonitoramento.resetPolylines();
    }
    catch { }

    try
    {
        GMapAreaMonitoramento.resetInfoWindows();
    }
    catch { }

    try
    {
        GMapAreaMonitoramento.resetPolygon();
    }
    catch { }

    Session["PontosOriginal"] = null;
}

```

```

protected void ASPxComboBoxColetas_SelectedIndexChanged(object sender, EventArgs
e)
{
    WebServiceReference.GeneralService service = new
WebServiceReference.GeneralService();
    service.Url =
GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefe
rence_GeneralService;
    CarregarMapa(service);
    service.Dispose();
    service = null;
}

protected void ComboBoxDispositivoMovel_SelectedIndexChanged(object sender,
EventArgs e)
{
    WebServiceReference.GeneralService service = new
WebServiceReference.GeneralService();
    service.Url =
GlobalTracker.WebSite.Properties.Settings.Default.GlobalTracker_WebSite_WebServiceRefe
rence_GeneralService;
    CarregarMapa(service);
    service.Dispose();
    service = null;
}
}
}

```

Anexo F – Script para Criação do Banco de Dados Global Tracker

```

CREATE DATABASE [GlobalTracker]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'GlobalTracker', FILENAME = N'c:\Program Files\Microsoft SQL
Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\GlobalTracker.mdf' , SIZE = 5120KB , MAXSIZE =
204800KB , FILEGROWTH = 5120KB )
    LOG ON
    ( NAME = N'GlobalTracker_log', FILENAME = N'c:\Program Files\Microsoft SQL
Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\GlobalTracker_log.LDF' , SIZE = 7168KB , MAXSIZE
= 102400KB , FILEGROWTH = 5120KB )
GO
ALTER DATABASE [GlobalTracker] SET COMPATIBILITY_LEVEL = 100
GO
CREATE TABLE [dbo].[CERCAELETRONICA](
    [IMEI] [varchar](255) NOT NULL,
    [PERMITIRPERIMETROCERCA] [varchar](3) NOT NULL,
    [LISTALATITUDELONGITUDE] [varchar](2000) NULL,
    CONSTRAINT [PK_CERCAELETRONICA] PRIMARY KEY NONCLUSTERED
    (
        [IMEI] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
ALTER TABLE [dbo].[CERCAELETRONICA] ADD DEFAULT ('Sim') FOR [PERMITIRPERIMETROCERCA]
GO
CREATE TABLE [dbo].[COORDENADADISPOSITIVOMOVEL](
    [CODIGO] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [IMEI] [varchar](255) NOT NULL,
    [LATITUDE] [float] NOT NULL,
    [LONGITUDE] [float] NOT NULL,
    [ENDERECO] [varchar](255) NULL,
    [DATA] [datetime] NOT NULL,
    CONSTRAINT [PK_COORDENADADISPOSITIVOMOVEL] PRIMARY KEY NONCLUSTERED
(
    [IMEI] ASC,
    [DATA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
CREATE TABLE [dbo].[DISPOSITIVOMOVEL](
    [CODIGO] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [IMEI] [varchar](255) NOT NULL,
    [SENHA] [varchar](255) NOT NULL,
    [NOME] [varchar](255) NULL,
    [EMAIL] [varchar](255) NULL,
    [SISTEMAOPERACIONAL] [varchar](255) NULL,
    [INTERVALOMONITORAMENTO] [int] NULL,
    CONSTRAINT [PK_DISPOSITIVOMOVEL] PRIMARY KEY NONCLUSTERED
(
    [IMEI] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
CREATE TABLE [dbo].[HISTORICOALERTA](
    [IMEI] [varchar](255) NOT NULL,
    [DATA] [datetime] NOT NULL,
    [LATITUDE] [float] NOT NULL,
    [LONGITUDE] [float] NOT NULL,
    [ENDERECO] [varchar](255) NULL,
    CONSTRAINT [PK_HISTORICOALERTA] PRIMARY KEY NONCLUSTERED
(
    [IMEI] ASC,
    [DATA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
CREATE TABLE [dbo].[USUARIO](
    [CODIGO] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [LOGIN] [varchar](255) NOT NULL,
    [SENHA] [varchar](255) NOT NULL,
    [NOME] [varchar](255) NOT NULL,
    [ATIVO] [varchar](3) NOT NULL,
    CONSTRAINT [PK_USUARIO] PRIMARY KEY NONCLUSTERED
(
    [LOGIN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO