



Fundação Educacional do Município de Assis  
IMESA - Instituto Municipal de Ensino Superior de Assis

**TIAGO QUEIROZ LOUREIRO**

**DESENVOLVIMENTO DE APLICATIVOS MÓVEIS COM O USO DA  
TECNOLOGIA ANDROID E GEOPROCESSAMENTO**

Assis  
2013

**TIAGO QUEIROZ LOUREIRO**

**DESENVOLVIMENTO DE APLICATIVOS MÓVEIS COM O USO DA  
TECNOLOGIA ANDROID E GEOPROCESSAMENTO**

Monografia apresentada como exigência para obtenção do grau de Bacharelado em Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA.

Orientado: Tiago Queiroz Loureiro

Orientador: Diomara Martins Reigato Barros

Assis

2013

## DEDICATÓRIA

Dedico esse trabalho a minha família e amigos por me apoiarem durante mais uma etapa da minha vida.

## AGRADECIMENTO

Agradeço a Deus antes de tudo por me permitir viver e desfrutar desse maravilhoso mundo.

Agradeço a os meus pais Carlos Ivonei Loureiro e Neusa Maira Queiroz Loureiro, a os meus irmãos Carlos Ivonei Loureiro Junior e Vanessa Queiroz Loureiro, que sempre estiveram ao meu lado me apoiando e incentivando.

Agradeço a minha orientadora Diomara Martins Reigato Barros e professores pela amizade e conhecimento adquirido durante esses anos.

## RESUMO

O objetivo do trabalho consiste em um estudo das tecnologias Android e Geoprocessamento, possibilitando assim a elaboração um sistema de coleta de dados para o auxílio da equipe de profissionais da Vigilância Epidemiológica Municipal. Com o desenvolvimento de um aplicativo para dispositivo móvel utilizado pela equipe no momento das visitas nas residências e possibilitando registrar através de coordenadas por GPS os locais que possam ocorrer à proliferação do mosquito, onde antes não era possível a identificação por nomes de ruas e números, locais como cidades litorâneas onde os moradores abitam os cerrados, em sítios, favelas entre outras áreas remotas que possam ocorrer casos da doença. Um segundo aplicativo desenvolvido para os computadores de mesa também chamado de desktop, onde são armazenadas todas as informações colhidas pelos agentes através do aplicativo móvel, gerando assim relatórios e uma melhor visualização dos pontos afetados através de mapas.

Palavras-chave: Android; Vigilância Epidemiologia; Saúde.

## ABSTRACT

The aim of the work is a study of Android and GIS technologies, thus enabling the development a system of data collection for the help of the professional staff of the Epidemiological Surveillance Hall. With the development of a mobile application for use by staff at the time of the visits in homes and register through enabling GPS coordinates for the locations that may occur to the proliferation of the mosquito, where before it was not possible to identify by street names and numbers, places like coastal cities where residents live in savannas, in places, slums and other remote areas that may occur cases of the disease. A second application developed for desktop computers also called desktop, which stores all the information gathered by agents through the mobile app, generating reports and better visualization of affected points through maps.

Keywords: Android; Surveillance Epidemiology Health

## LISTA DE FIGURAS

Figura 1 - Boletim de Coleta de dados .....	11
Figura 2 - HTC Dream G1 - Primeiro celular com Sistema .....	13
Figura 3 - Open Hendset Alliance .....	14
Figura 4 - Net. Framework.....	19
Figura 5 - Ilustração Entity framework.....	20
Figura 6 - Mensagem SOAP .....	22
Figura 7 - Activity Principal.java .....	24
Figura 8 - Tela de login e opções do sistema.....	24
Figura 9 - View principal.xml .....	25
Figura 10 - Castro de Imóvel e registro .....	26
Figura 11 - Método Enviar dados Web Service .....	27
Figura 12 - String de Conexão Web Service .....	27
Figura 13 - Obter Coordenadas.....	28
Figura 14 - Método Webservice.....	28
Figura 15 - Gerenciador do ISS.....	29
Figura 16 - Exemplo Entity framework .....	30
Figura 17 - Tela Cadastro de endereço.....	31
Figura 18 - Mapa do Endereço.....	32
Figura 19 - URL solicitando mapa .....	33
Figura 20 - Tela de registro .....	33
Figura 21 - Mapa de inclusão de registro .....	34
Figura 22 – Diagrama de Casos de Uso Aplicativo móvel .....	35
Figura 23 - Casos de Uso do software Centralizador.....	38
Figura 24 Diagrama de entidade e relacionamento.....	43

## LISTA DE ABREVIATURAS E SIGLAS

CLR - Common Language Runtime.

GPS - Global Positioning System (Sistema de Posicionamento Global)

IDE - Integrated Development Environment (Ambiente Integrado de desenvolvimento)

CTS - Common Type System

MSIL - Microsoft Intermediate Language

JIT - Just-in-time compilation

ORM - Object/Relational Mapping

LINQ - Language-Integrated Query

IIS - Internet Information Services

SOAP - Simple Object Access Protocol

URI - Uniform Resource Identifier

XML - Extensible Markup Language

*URL* - Uniform Resource Locator



**Erro! Fonte de referência não encontrada.**

## SUMÁRIO

1. INTRODUÇÃO .....	10
1.1 Objetivos do projeto .....	11
1.2 Justificativa .....	11
1.3 Motivação .....	12
2. TECNOLOGIAS .....	13
2.1 Android .....	13
2.1.1 Open Handset Alliance .....	14
2.1.2 Kernel Linux.....	14
2.1.3 Código aberto e livre.....	15
2.2 Geoprocessamento .....	15
2.2.1 SIG - Sistema de informações geográficas .....	16
2.2.2 Dados Espaciais .....	16
2.2.3 Sensoriamento Remoto .....	17
2.3 Plataforma Microsoft .Net .....	17
2.3.1 Net Framework .....	18
2.3.2 Entity Framework.....	19
2.4 Web Service .....	20
2.4.1 XML (Extensible Markup Language).....	21
2.4.2 SOAP - Protocolo Simples de Acesso .....	21
3. DESENVOLVIMENTO E DESCRIÇÃO DOS PROTÓTIPOS .....	23
3.1 Aplicativo de Coleta .....	23
3.2 Serviços Web.....	28
3.3 Centralizador de Dados .....	29
3.3 Casos de Uso .....	35

3.3.1 Aplicativo Móvel.....	35
3.3.1.1 Efetuar Login.....	36
3.3.1.2 Inserir Registro.....	36
3.3.1.3 Cadastrar imóvel.....	37
3.2.2 Aplicativo Centralizador.....	38
3.2.2.1 Manter Usuários.....	38
3.2.2.2 Manter Imóveis.....	39
3.2.2.3 Manter Registros.....	39
3.2.2.4 Manter Setores.....	40
3.2.2.5 Manter Quadras.....	40
3.2.2.6 Manter Tipos de Larvas.....	41
3.2.2.7 Manter Tipos de Ocupações.....	41
3.2.2.8 Efetuar Login.....	42
3.2.2.9 Mostrar Mapa.....	42
3.2.3 Diagrama Entidade relacionamento.....	43
4. Conclusão.....	44

## 1. INTRODUÇÃO

O objetivo deste trabalho é demonstrar o desenvolvimento de uma solução para dispositivos móveis para auxílio dos Agentes de Saúde da Vigilância Epidemiológica em suas visitas diárias em residências e locais sem infraestruturas, em busca de focos do mosquito transmissor da doença *Aedes Aegypti*.

Um das vantagens em coletar os dados por um sistema móvel são os recursos oferecidos pelos dispositivos atuais como o GPS, locais difíceis de serem registrados por não haver nome de ruas ou numerações como favelas, serrados, litorais, locais de despejo de lixo entre outros, com a ajuda desse recurso poderão ser localizado facilmente a partir dos dados coletados pelo sistema de GPS, obtendo assim informações antes desconhecidas.

Para desenvolvimento do software de coleta de dados foi optado o uso da tecnologia Android, sistema operacional da Google, levando em consideração o baixo custo dos dispositivos, considerando que o maior número está com sistemas operacionais Android em relação a os seus concorrentes no mercado, o Windows Mobile da Microsoft, e o iOS sistema operacional de Apple, considerando também uma maior flexibilidade na área técnica para o desenvolvedor, como por exemplo, ferramentas gratuitas para o desenvolvimento de aplicativos.

Conceitos de Geoprocessamento como SIG (Sistema de Informações Geográficas), Dados Espaciais, Sensoriamento Remoto e outros conceitos que estão relacionados com as tecnologias utilizadas foram essenciais para o desenvolvimento do trabalho.

Com informações coletadas pelos agentes através de um SIG é possível obter informações detalhadas sobre a situação local, podendo assim prevenir o alastramento da doença em outras regiões através de relatórios e mapas com as informações e localizações detalhada dos locais e domicílios com focos de larvas, possibilitando assim um planejamento melhor das ações a fim de evitar um aumento no número de casos da doença.

### 1.1 Objetivos do projeto

Junto aos estudos das tecnologias Android, Geoprocessamento e Microsoft .NET desenvolver uma solução tecnológica, para proporcionar aos profissionais da área maior agilidade na coleta de dados no trabalho realizado em campo no momento das visitas, desenvolvendo uma ferramenta de coleta de dados, utilizada na hora das visitas em campo pelos agentes epidemiológicos. Permitindo assim um auxílio na tomada de ações e a inclusão digital das atividades de vigilância e controle antes registradas via formulários conforme mostrado na figura abaixo:

**PREFEITURA MUNICIPAL**  
**SECRETARIA MUNICIPAL DE SAÚDE**  
**CAMPINAS - SP**

**BOLETIM DE ATIVIDADES DE VIGILÂNCIA E CONTROLE DE Aedes aegypti**

MUNICÍPIO CAMPINAS: \_\_\_\_\_ ÁREA: \_\_\_\_\_ SETOR: \_\_\_\_\_ SETOR CENSITÁRIO: \_\_\_\_\_ QUART: \_\_\_\_\_ EXECUÇÃO: \_\_\_\_\_ DATA: \_\_\_\_\_ FOLHA: \_\_\_\_\_ CÓDIGO: \_\_\_\_\_

ROTINA  PENDÊNCIA  DEMANDA

**ATIVIDADES**

1- PONTO ESTRATÉGICO  4- BLOQUEIO - CONTROLE CRIADOURO (BUSCAATIVA)  7- IMÓVEIS ESPECIAIS  10- CASA A CASA - INTENSIFICAÇÃO

2- PESQUISA DE ARMADILHA  5- BLOQUEIO - NEBULIZAÇÃO  8- CRIAD. ESPECÍFICO  11- OUTROS

3- DELIMITAÇÃO DE FOCO (CAD)  6- ARRASTÃO  9- CASA A CASA ROTINA  12- ADL

CADASTRO OU NÚMERO DE ORDEM	ENDEREÇO RUA, AVENIDA	NÚMERO COMPLEM. R A B	SITUAÇÃO DO IMÓVEL							RECIPIENTES				TRATAMENTO QUÍMICO				LARVAS			P R E S E N T E R A M A O		
			T	R	A	F	D	T	P	R	TIPO	EXIST	COM ÁGUA	COM LARVA	C M O E N C T A R N O I L L C E O	PRD ALTER	FOCAL	PERI FOCAL	NEBULL ZAÇÃO	A M O S T R A		EXAM	Ae seg
<b>TOTAL</b>																							

FOLHA: 01/01 - ALTERADO 08/11 - VIA ÚNICA - FORMADO - (020/25) - CÓDIGO MATERIAL: 12317

**TIPO DE RECIPIENTE**

A- DEPOSITO ELEVAÇÃO  
 1- LIGADO A REDE  
 2- NÃO LIGADO A REDE  
 8- DEPOSITO NÃO ELEVAÇÃO  
 3- LIGADO A REDE  
 4- NÃO LIGADO A REDE

C- IMÓVEIS  
 5- VASO DE PLANTA NA ÁGUA  
 6- VASO DE PLANTA (DIVERSOS)  
 7- PRATO / PRADEIRA  
 8- CONHEIMO ANIMAL  
 9- DEPOSITO PARA CONSTRUÇÃO  
 10- DEPOSITO PARA HORTICULTURA  
 11- PISCINA IMÓVEIS  
 12- LATA, FRASCO, PLÁSTICO UTILIZÁVEIS  
 13- GARRAFAS DESCONTÁVEIS  
 14- BALDE / REGADOR  
 15- BANDEJA DE LIXEIRA / AR COND.  
 16- MATERIAL DE CONSTRUÇÃO  
 17- OUTROS

D- FISSAS  
 18- RALO INTERIO  
 19- RALO EXTERNO  
 20- LIXE  
 21- CALHA  
 22- VASO SANITARIO / CASA DESCARGA  
 23- PISCINA  
 24- DEPOSITO PARA CONSTRUÇÃO  
 25- DEPOSITO PARA HORTICULTURA  
 26- CONHEIMO ANIMAL  
 27- OUTROS

E- PREVIS  
 28- OUTROS CORRELADOS

F- PASSIVEIS REBOLÇO / ALTERAÇÃO  
 29- LATA, FRASCO, PLÁSTICO  
 30- GARRAFAS DESCONTÁVEIS  
 31- LONA, ENCRADO, PLÁSTICO  
 32- ENVELOPE DE CONSTRUÇÃO  
 33- PEÇAS / BUCATAS  
 34- MASSERA  
 35- BARCO  
 37- OUTROS

G- NATURAIS  
 38- ODO DE JARDIM E BAMBO  
 39- BRUCELAS  
 40- OUTROS

PENDÊNCIA  
 P- FISCAL  
 D- DESOcupado  
 T- TEMPORARIA

CAMPO  
 RESPONSÁVEL  
 VISTO  
 LABORATORIO  
 DATA  
 VISTO  
 NÚMERO DE IMÓVEIS POSITIVOS  
 aegypti  
 albopictus  
 ambos

Figura 1 - Boletim de Coleta de dados

### 1.2 Justificativa

Nenhum órgão público ou federal disponibiliza um sistema de coleta de dados na área de Vigilância Epidemiológica para auxiliar os gestores e agentes de saúde dos municípios, devido a essa carência de um sistema de informações, os

municípios não possuem um controle sobre os dados coletados, dificultando assim o trabalho contra o contágio e proliferação do mosquito da dengue.

### 1.3 Motivação

O estudo interdisciplinar das tecnologias Android, Geoprocessamento e Microsoft .NET, associado a saúde pública foi a motivação desse trabalho. Unindo também a escassez no mercado de software, contribuindo para os municípios que hoje em dia não possuem um sistema de coleta de dados.

## 2. TECNOLOGIAS

Neste Capítulo será descrito as tecnologias e conceitos utilizados para o desenvolvimento do aplicativo descrito no trabalho.

### 2.1 Android

Teve início em outubro de 2003, na cidade de Palo Alto na Califórnia, foi inicialmente desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White junto com a fundação da Android Inc. empresa que desenvolvia sistemas operacionais para celulares, em agosto de 2005, a Google anunciou a compra da Android Inc, lançando o primeiro aparelho com o sistema Android no dia 22 de outubro de 2008, o HTC Dream G1 ilustrado na figura abaixo, com funções e aplicativos bem diferenciados em comparação a os sistemas da época, com uma central para download de aplicativos e jogos, o Android Market atual Google Play, e hoje o Android é o sistema operacional para dispositivo móvel mais popular no mundo, por sua capacidade de integrar diferentes componentes eletrônicos como GPS, bússola, Bluetooth, wireless, entre outros e um único dispositivo.

Sua tecnologia é desenvolvida com base no núcleo do Linux e é focado tanto para dispositivos móveis como embarcados, atualmente é utilizado em diversos produtos de vários fabricantes é mantido por um grupo de empresas denominado Open Handset Alliance com a liderança da Google. (Lacheta, 2013)



Figura 2 - HTC Dream G1 - Primeiro celular com Sistema

### 2.1.1 Open Handset Alliance

Levando em consideração a crescente popularização dos celulares que é hoje em dia o aparelho mais utilizado no mundo com mais de três bilhões de aparelhos em uso, ultrapassando o televisor que é tem cerca 1,5 bilhões de aparelhos.

Foi então fundado o Open Handset Alliance um grupo integrado pelas grandes empresas que atuam no mercado de telefonia móvel que até o momento em que foi elaborado esse do trabalho eram de 94 empresas, que se uniram para acelerar as inovações em tecnologia móvel e oferecer uma experiência móvel mais rica, mais barata e melhor, desenvolvendo e aprimorando o sistema operacional Android a primeira plataforma de código aberto para dispositivos move.

Abaixo algumas empresas do grupo OHA. (Alliance, 2013)



Figura 3 - Open Handset Alliance

### 2.1.2 Kernel Linux

O sistema operacional Android foi baseado no kernel 2.6 do Linux, e é responsável por gerenciar a memória, os processos, threads e a segurança dos arquivos e pastas, além de redes e drives.

Cada aplicativo no Android dispara um novo processo no sistema operacional e diversos processos podem ser executados simultaneamente, e o

kernel do sistema operacional é o responsável por realizar o controle de memória.

O sistema de segurança é baseado no Linux, cada aplicação é executada em um único processo e cada processo possui uma única thread dedicada. (Lacheta, 2013)

### 2.1.3 Código aberto e livre

O Android é a primeira plataforma para aplicações moveis completamente livres e de código aberto (open-source), uma boa vantagem para a sua evolução, pois diversos programadores do mundo contribuem para o seu desenvolvimento, uma vantagem também para o fabricante de celulares que utilizam do sistema sem custo algum. (Lacheta, 2013)

## 2.2 Geoprocessamento

O Geoprocessamento utiliza técnicas matemáticas e computacionais para o tratamento da informação geográfica e está influenciando de maneira crescente as áreas de Cartografia, Análise de Recursos Naturais, Transportes, Comunicações, Energia e Planejamento Urbano e Regional. As ferramentas computacionais para Geoprocessamento, chamadas de Sistemas de Informação Geográfica GIS, permitem realizar análises complexas, ao integrar dados de diversas fontes e ao criar bancos de dados geo-referenciados. Tornam ainda possível automatizar a produção de documentos cartográficos, como por exemplo, manipular maquina e equipamentos através do geoprocessamento, como por exemplo, tratores que trabalham sem motoristas utilizando um Sistema de informação geográfica. (Gilberto Câmara, 2001)



### 2.2.1 SIG - Sistema de informações geográficas

O termo Sistemas de Informação Geográfica (SIG) é aplicado para sistemas que realizam processamentos de dados computacionais geográficos e recuperam informações não apenas de bases convencionais, mas também através de sua localização espacial com o uso de uma base de dados Geográfica para o armazenamento de dados Espaciais, oferecendo assim ao gestor (urbanista, planejador, engenheiro) uma visão inédita de seu ambiente de trabalho, em que todas as informações disponíveis sobre um determinado assunto estão ao seu alcance, referenciadas por sua localização geográfica (Gilberto Câmara, 2001).

Para que isto seja possível, a geometria e os atributos dos dados num SIG devem estar geo-referenciados, localizados na superfície terrestre e representados numa projeção cartográfica. Para cada objeto geográfico, o SIG necessita armazenar seus atributos e as várias representações gráficas associadas. Devido a sua ampla gama de aplicações, que inclui temas como agricultura, floresta, cartografia, cadastro urbano e redes de concessionárias (água, energia e telefonia), há pelo menos três grandes maneiras de utilizar um SIG:

- como ferramenta para produção de mapas;
- como suporte para análise espacial de fenômenos;
- como um banco de dados geográficos, com funções de armazenamento e recuperação de informação espacial.

### 2.2.2 Dados Espaciais

Dado espacial se caracteriza pelo atributo da localização geográfica, e outros fatores importantes aos dados espaciais, mas a localização é preponderante. Um objeto qualquer (como uma cidade, uma casa, um ponto) somente tem sua localização geográfica estabelecida quando pode descrevê-lo em relação a outro objeto cuja posição seja conhecida ou quando se determina sua localização em relação a certo sistema de coordenadas. O estabelecimento de localizações sobre a

superfície terrestre sempre foi um dos objetos de estudo da Geodésia, ciência que se encarrega da determinação da forma e das dimensões da Terra. (Gilberto Câmara, 2001)

### 2.2.3 Sensoriamento Remoto

Sensoriamento remoto é o conjunto de técnicas que possibilita a obtenção de informações sobre alvos na superfície terrestre (objetos, áreas, fenômenos), através do registro da interação da radiação eletromagnética com a superfície, realizado por sensores distantes, ou remotos.

Geralmente estes sensores estão presentes em plataformas orbitais ou satélites, aviões. A NASA é uma das maiores captadoras de imagens recebidas por seus satélites. No Brasil, o principal órgão que atua nesta área é o Instituto Nacional de Pesquisas Espaciais - INPE. (Gilberto Câmara, 2001)

### 2.3 Plataforma Microsoft .Net

A plataforma Microsoft oferece suporte a aplicativos e soluções projetadas para os menores dispositivos, bem como para as maiores empresas e trabalhar com uma variedade de linguagens, padrões e ferramentas de programação, para criar aplicativos para computadores com Sistema operacional Windows, dispositivos moveis e paginas webs oferecendo ferramentas para desenvolvedores individuais ou para uma equipe de desenvolvimento. As principais linguagens de programação oferecidas para Microsoft são: Visual Basic, Visual C#, Visual C++, Visual J##, todas elas utilizando a plataforma. Net Framework (<http://msdn.microsoft.com>, 2013)

### 2.3.1 Net Framework

O Net. Framework é um conceito de desenvolvimento baseado em maquina virtual conhecido como CLR (Common Language Runtime), ou seja, todas as ações de um aplicativo passam primeiro pelo CLR e só depois vai para o sistema operacional, evitando inclusive dano aos mesmos, tal conceito denomina-se código gerenciado.

Ele tem em sua estrutura certa de 7000 mil classes disponíveis, com varias funcionalidades que facilitam no desenvolvimento de um uma aplicação, como: acesso a dados, comunicação e manipulação de arquivos, e serve como uma plataforma base que tem o objetivo de oferecer toda infra-estrutura para o desenvolvimento de aplicações.

Baseado totalmente em um modelo orientado ao objeto com as mais novas tecnologias do mercado possui o conceito de CTS (Common Type System), permitindo a utilização de diversas linguagens de programação, fazendo com que, qualquer que seja a linguagem escolhida, depois de compilada seja gerado um aplicativo .NET com as mesmas características, ou seja, independente da linguagem utilizada o resultado final será o mesmo.

Usa um padrão conhecido como MSIL(Microsoft Intermediate Language), que no momento da primeira utilização da aplicação será compilado automaticamente, usando um processo conhecido como JIT(Just-in-time compilation). (Spaki, Alves, Leal de Farias, Lino Neto, Althamann, & Kono, 2008)

Abaixo a ilustração representando as etapas percorridas de um aplicativo do desenvolvimento ate execução:

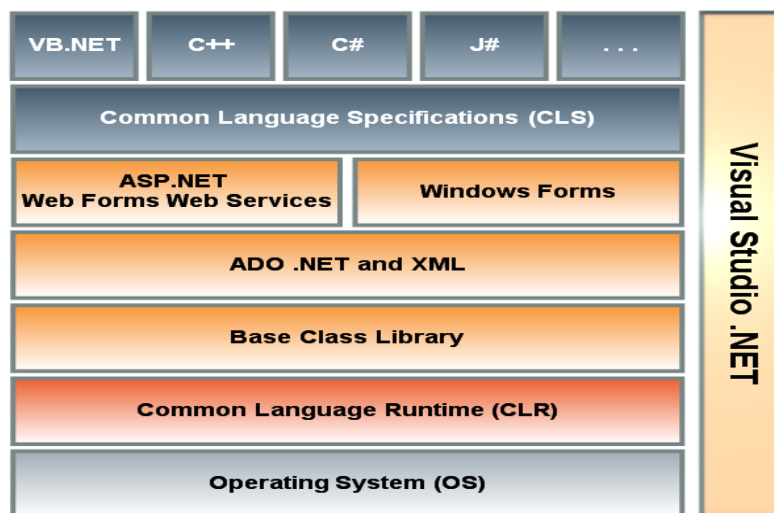


Figura 4 - Net. Framework

Fonte: <http://www.softidsolutions.com/services/technology/net.html>

### 2.3.2 Entity Framework

O Entity Framework é um framework do tipo ORM (Object/Relational Mapping) ou seja ele mapeia as classes existentes e se responsabiliza pela persistência em um banco de dados permitindo assim aos desenvolvedores trabalhar com dados relacionais como objetos de domínio específico, eliminando a necessidade de maior parte dos códigos de acesso de dados que os desenvolvedores geralmente precisam escrever. Com o Entity Framework, os desenvolvedores podem lançar consultas usando LINQ (Language-Integrated Query), e depois recuperar e manipular dados como objetos fortemente tipificados. O Entity Framework fornece serviços como rastreamento de alterações, resolução de identidades e tradução de consultas para que os desenvolvedores possam se concentrar na lógica de negócios de seus aplicativos em vez dos princípios básicos de acesso a dados.

Com uso do Entity Framework funciona com os bancos de dados mais populares do mercado como a Microsoft SQL Server, Oracle e DB2. Fornece também ferramentas integradas do Visual Studio para criar visualmente modelos de entidade

e gerar de forma autônoma modelos a partir de um banco de dados existente. (http://msdn.microsoft.com/, 2013) Abaixo a uma ilustração demonstrando o funcionamento do Entity Framework:

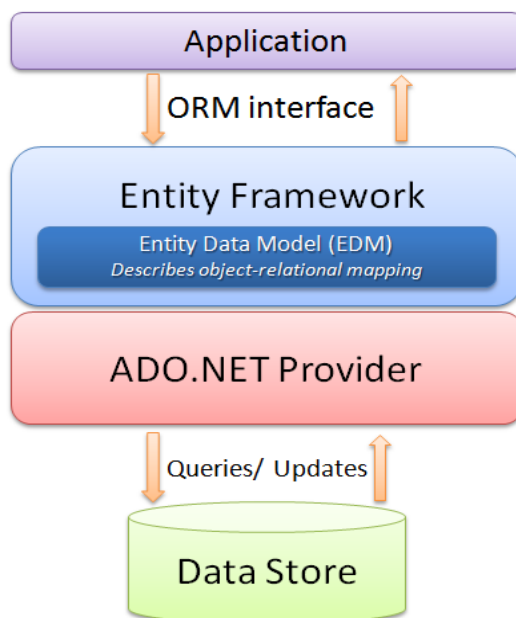


Figura 5 - Ilustração Entity framework

Fonte <http://msdn.microsoft.com/pt-br/data/aa937709.aspx>

## 2.4 Web Service

Web service é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis, são componentes que permitem às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato XML.

Utilizando a tecnologia Web Service, uma aplicação pode invocar outra para efectuar tarefas simples ou complexas mesmo que as duas aplicações estejam em diferentes sistemas e escritas em linguagens diferentes. Por outras palavras, a Web Services faz com que os seus recursos estejam disponíveis para que qualquer

aplicação cliente possa operar e extrair os recursos fornecidos pelo Web Service que são identificados por um URI (Uniform Resource Identifier), descritos e definidos usando XML (Extensible Markup Language). Um dos motivos que tornam os Web Services atractivos é o facto de este modelo ser baseado em tecnologias Standards, em particular XML e HTTP (Hypertext Transfer Protocol) e são utilizados para disponibilizar serviços interactivos na Web, podendo ser acessados por outras aplicações usando, por exemplo, o protocolo SOAP (Simple Object Access Protocol). (wikipedia, 2013)

#### 2.4.1 XML (Extensible Markup Language)

O XML é uma linguagem de marcação para a criação de documentos com dados organizados hierarquicamente, tais como textos, banco de dados ou desenhos vectoriais. A linguagem XML é classificada como extensível porque permite definir os elementos de marcação.

Linguagem de marcação é um conjunto de códigos que podem ser aplicados a dados ou textos para serem lidos por computadores ou pessoas. Por exemplo, o HTML é uma linguagem de marcação para organizar e formatar um website, já o XML tem o mesmo conceito, mas para padronizar uma sequência de dados com o objetivo de organizar, separar o conteúdo e integrá-lo com outras linguagens.

O XML traz uma sintaxe básica que pode ser utilizada para compartilhar informações entre diferentes computadores e aplicações. Quando combinado com outros padrões, torna possível definir o conteúdo de um documento separadamente de seu formato, tornando simples para reutilizar o código em outras aplicações para diferentes propósitos.

Portanto, uma das suas principais características é sua portabilidade, pois, por exemplo, um banco de dados pode escrever um arquivo XML para que outro banco consiga lê-lo. (Pereira, 2009)

#### 2.4.2 SOAP - Protocolo Simples de Acesso

É um protocolo de comunicação baseado em XML que permite a comunicação de mensagens entre aplicações via HTTP, normalmente utilizado em Webservices. Uma das grandes qualidades desse protocolo é sua independência de plataforma e linguagem além de ser simples e extensível por utilizar XML. Uma mensagem SOAP é um documento XML comum contendo um elemento chamado Envelope que identifica o documento XML como uma mensagem SOAP, um elemento Header que contém informações sobre o cabeçalho do documento, e um elemento Body que é o corpo do documento contendo informações de chamada e resposta, dentro do corpo contém um elemento Fault que contém erros e informações de status. (Sampaio, 2006)

-Elemento Envelope: é o elemento raiz da mensagem SOAP. Este elemento define o documento XML como uma mensagem SOAP.

- Elemento Header: contém informações específicas do aplicativo da mensagem SOAP. É um elemento opcional, se ele estiver presente deve ser o primeiro elemento filho do elemento Envelope.

-O Elemento Body: contém a mensagem SOAP pretendida que o usuário espera.

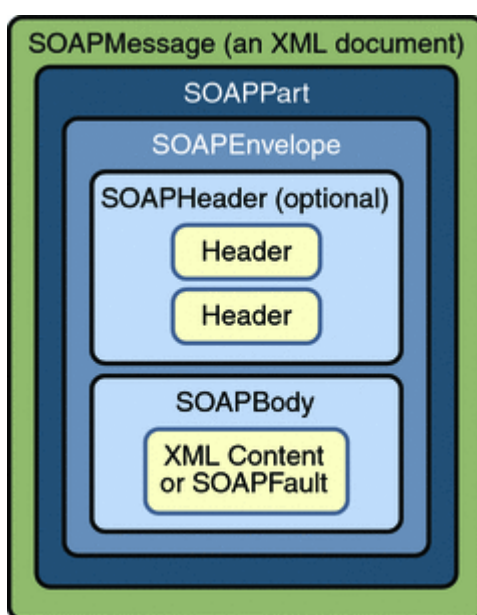


Figura 6 - Mensagem SOAP

### 3. DESENVOLVIMENTO E DESCRIÇÃO DOS PROTÓTIPOS

Para o desenvolvimento de protótipo foi utilizado os conceitos de programação Android, Geoprocessamento de dados e Microsoft .NET, com o auxílio de ferramentas da UML como diagrama de classe, casos de uso.

Com base nos estudos, dois sistemas e um serviço web foram desenvolvidos, um sistema de coleta de dados, com a tecnologia Android, um sistema que centraliza em uma base única os dados coletados, e um serviço web para uso do aplicativo móvel no momento da persistência no banco de dados desenvolvido com as tecnologias da Microsoft .NET .

#### 3.1 Aplicativo de Coleta

O sistema de coleta de dados foi desenvolvido em Android com a IDE Eclipse versão Índigo Service Release 2. Foram utilizados os conceitos básicos da plataforma de Google Android com uso das principais classes padrão.

A classe `android.app.Activity` que representa basicamente uma tela de uma aplicação, no projeto foram utilizados 4 activity em conjunto com a classe `android.view.View` na qual representa os elementos visuais das activity.

Essas duas classes sempre andam juntas, A activity define que existe uma tela, controla seu estado e a passagem de parâmetros de uma tela para outra, define métodos que serão chamados quando o usuário pressionar algum botão etc.

Mais a activity precisa exibir elementos visuais na tela , e esse é o papel da classe view, que define as finalidades de desenhar algo na tela.

Podemos dizer que a activity é o controle do padrão MVC (Model View controller), e a view, o próprio nome diz.

Um view pode ser um simples componente gráfico (botão, checkbox, imagem) ou uma view complexa, que atua como em gerenciador de layout, a qual pode conter varias views-filhas, e tem a função de organizar as mesmas na tela.

O método `setContentView(view)` é o que faz essa ligação entre a activity e a view e recebe como parâmetro a view que será exibida na tela, Sua chamada deve ser feita sempre no método `onCreate(bundle)` da activity, o método `onCreate(bundle)`



precisa ser implementado obrigatoriamente e é chamado de forma automática pelo Android quando a tela é Criada. (Lacheta, 2013)

```
package com.example.tccprojetomobile;

import com.example.tccprojetomobile.R.id;

public class Principal extends Activity {
    Button btEndereco = (Button) findViewById(R.id.btEndereco);
    Button btRegistro = (Button) findViewById(id.btRegistro);
    String idUsuario;
    String Usuario;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.principal);
        CarregaIntent();
        Listners();
    }
}
```

Figura 7 - Activity Principal.java

Como podemos ver no exemplo acima a classe activity Principal.java do projeto aonde se encontra as opções de cadastrar endereço e inserir registro. E abaixo na figura 7b a representação gráfica do arquivo principal.xml chamado pelo método `setContetView(R.Layout.principal)` para exibir a view responsável pelos componentes gráficos.

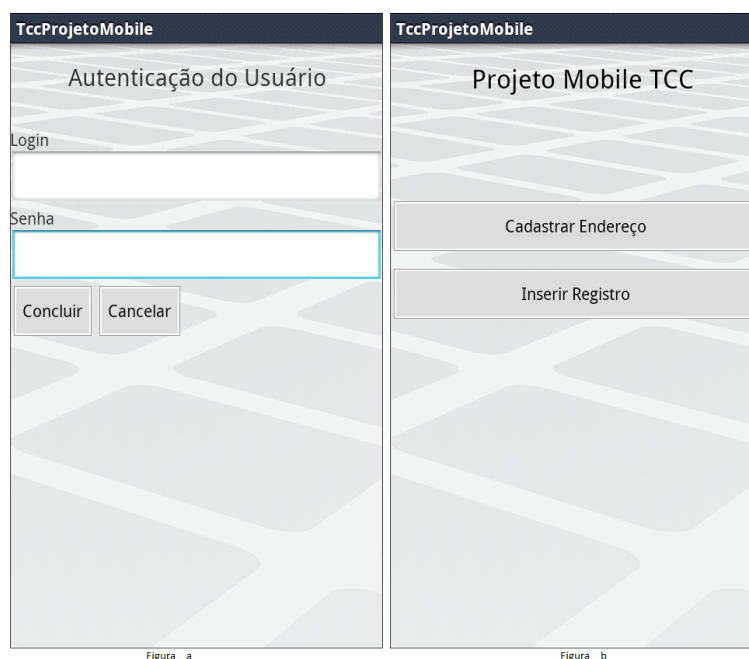


Figura 8 - Tela de login e opções do sistema

O método `setContentView(view)` é o que faz a ligação entre a activity e a view e recebe como parâmetro a view que será exibida na tela, sua chamada deve ser feita sempre no método `onCreate(bundle)` da activity.

Ao construir a tela usando um arquivo XML de layout, surge a necessidade de recuperar os objetos definidos no arquivo XML, no caso da figura 6.b temos dois botões e duas variáveis do tipo String.

Para que recuperar essas informações é utilizado o método `findViewById(id)`. Esse método recebe o id do componente desejado e retorna uma subclasse de `android.view.View`, como as classes `Button`, `ImageView` etc.

O id componente está localizado no arquivo principal.xml na tag `<android:id>` de cada componente, podemos identificar abaixo as ids dos botões apresentados na figura 7b em seu formato XML.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

    <Button
        android:id="@+id/btEndereco"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="90dp"
        android:text="Cadastrar Endereço" />

    <Button
        android:id="@+id/btRegistro"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Inserir Registro" />

</LinearLayout>
```

Figura 9 - View principal.xml

Ao executar o aplicativo temos a tela inicial conforme figura 6. aonde o agente de saúde responsável pela coleta de dados será obrigado a digitar o usuário e senha para ter acesso às demais funções do sistema.

Com o login efetuado outras funções do sistema serão exibidas as opções de Inserir Endereço, Inserir Registro.

Abaixo a activity de a opção cadastro de endereço onde usuário deverá

informar as seguintes informações: quarteirão, rua e número, após o preenchimento dos dados o usuário deve confirmar a inclusão através de o botão salvar, nesse exato momento a localização geográfica é obtida pelo sistema e os dados são encaminhados para um serviço web responsável pela persistência no banco de dados, retornando um mensagem de sucesso.

The image displays two side-by-side screenshots of the TccProjetoMobile application interface. The left screenshot, titled 'Cadastro de Endereços', features a form with four input fields: 'Quadra' (containing a vertical bar), 'Rua', 'Número', and 'Usuario' (containing '1 - Tiago'). Below the fields are two buttons: 'Voltar' and 'Cadastrar'. The right screenshot, titled 'Registro de Atividade', features a form with seven input fields: 'Endereço', 'Pendecia' (dropdown menu with 'Normal'), 'Recipiente' (dropdown menu with 'Nenhum'), 'Quantidade de Recipiente' (containing '0'), 'Qtd recipiente com Agua' (containing '0'), 'Qtd recipiente com Larva' (containing '0'), and 'Tipo de Larva' (dropdown menu with 'Nenhuma'). Below the fields are two buttons: 'voltar' and 'Inserir'.

**Figura 10 - Castro de Imóvel e registro**

Nota se que o usuário não precisa informar a data de cadastro de endereço ou inclusão de registro, essa função é efetuada pelo serviço web no momento da persistência, registrando assim o momento exato da inclusão.

A comunicação entre o sistema Android e o serviço web foi feita atrás do uso da biblioteca ksoap2-android, uma biblioteca cliente SOAP leve e eficiente para a plataforma Android. Abaixo o código responsável por requisitar o serviço web utilizando a biblioteca ksoap2.

```

public void EnviaDadosWebService(String numero,String rua,String quadra,
    String usuario,String latitude,String longitude)
{
    try
    {
        SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
        SoapSerializationEnvelope sobre = new SoapSerializationEnvelope(SoapEnvelope.VER11);
        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);
        request.addProperty("quadra",quadra);
        request.addProperty("rua",rua);
        request.addProperty("numero",numero);
        request.addProperty("usuario",usuario);
        request.addProperty("latitude",latitude);
        request.addProperty("longitude",longitude);

        HttpTransportSE transporte = new HttpTransportSE(URL);

        transporte.call(SOAP_ACTION, sobre);

        resultado = (SoapPrimitive)sobre.getResponse();
        if(resultado.toString()=="Sucesso")
            MessageBox("Aviso", "Dados enviados com sucesso.");
        else MessageBox("Aviso", resultado.toString());
    }
    catch (Exception e) { }
}

```

Figura 11 - Método Enviar dados Web Service

No código acima o método recebe por parâmetro os dados de um endereço em formato do tipo texto, daí então é criado um objeto do tipo SoapObject(string,string) no qual a primeira string representa um endereço do Web Serviço (NAMESPACE) e a outra representa o método (METHOD\_NAME) que esta sendo requisitado. Abaixo um exemplo.

```

package com.example.tccprojetomobile;

import org.ksoap2.SoapEnvelope;

public class Registros extends Activity {

    SoapPrimitive resultado=null;
    private static final String SOAP_ACTION = "http://tempuri.org/inserirRegistro";
    private static final String METHOD_NAME = "inserirRegistro";
}

```

Figura 12 - String de Conexão Web Service

A opção Inserir Registro permite que o usuário do sistema inserira as informações colhidas de um determinado endereço no momento das visitas, as informações são: recipientes encontrados, imóvel pendente, quantidade de

recipientes, quantidade de recipientes com água, quantidade de recipientes com larva e tipo de larva. Após o preenchimento dos dados o usuário deve confirmar a inserção através do botão inserir, que chamara o método para obter as coordenadas geográficas e a função de requisição do Webservice para que sejam transferidas as informações e a persistência no banco de dados. Abaixo parte do código responsável por obter as coordenadas geográficas.

```
private void chamaGPS()
{
    try{
        Location loc;
        LocationManager LM = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        LocationListener ll = new NovaLocalizacao();
        LM.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, ll);
        loc = LM.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        lat = loc.getLatitude();
        longi = loc.getLongitude();

    }
    catch (Exception e) {
        MessageBox("Atencao", "Erro ao capturar coordenadas");
    }
}
}
```

Figura 13 - Obter Coordenadas

### 3.2 Serviços Web

O serviço web foi desenvolvido na IDE Visual Studio 2012 da Microsoft em linguagem C# e é utilizado com a função de receber os dados enviados pelo aplicativo de cole e persistir os dados na base local. Graças a esse serviço é possível uma interação entre o protótipo de coleta de dados e o protótipo de centralizador de dados. A figura abaixo mostra o método inserirEndereco que foi invocado pelo dispositivo móvel no exemplo da figura 11 do capitulo anterior.

```
[WebMethod]
public string inserirEndereco(string quadra, string rua, string numero, string usuario, string data, string latitude, string longitude)
{
    try
    {
        data = DateTime.Now.ToString();
        string strCon = @"Data Source=MACHINE\SQLEXPRESS;Initial Catalog=DbProjeto;user=sa;password=123";
        string sql = "insert into endereco(numero,Quarteirao_id,usuarios_id,rua,data,latitude,longitude) values "+
            "(" + numero + "," + quadra + "," + usuario + "," + rua + "," + data + "," + latitude + "," + longitude + ")";
        Data db = new Data();
        bool check = db.Exe(strCon, "Endereco", sql);
        return "Endereço Cadastrado com Sucesso.";
    }
    catch (Exception ex) { return ex.Message.ToString(); }
}
```

Figura 14 - Método Webservice

Para que o serviço esteja sempre disponível é necessário hospedar em um servidor web, no projeto o servidor escolhido foi o IIS fornecido pela Microsoft (Internet Information Services) por sua compatibilidade com o sistema operacional Windows 7 utilizado para desenvolver os protótipos. Abaixo a tela de gerenciamento fornecida pelo ISS.

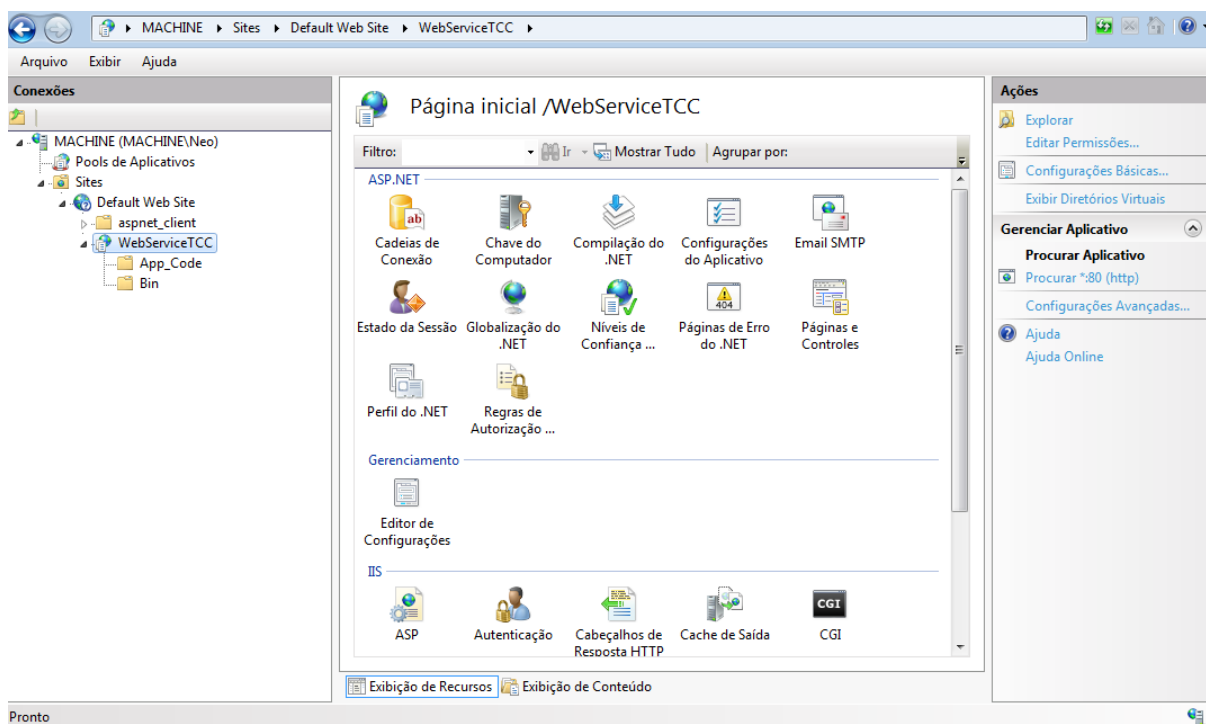


Figura 15 - Gerenciador do ISS

### 3.3 Centralizador de Dados

O sistema centralizador dados foi desenvolvido na IDE Visual Studio 2012 da Microsoft em linguagem C# utilizado com a função de unir todas as informações adquiridas pelo aplicativo de coleta em uma única base de dados.

Para o desenvolvimento do protótipo foi utilizada a tecnologia de mapeamento de objetos Entity Framework da Microsoft, responsável pelo mapeamento de todas as classes e persistência no banco de dados.

Na figura a seguir mostra um exemplo de um código inserindo, alterando e excluindo um objeto do banco de dados pelo uso do Entity Framework.

```

namespace Projeto
{
    public class appUsuario
    {
        DbProjetoEntities contexto = new DbProjetoEntities();

        public void Inserir(USUARIOS oUsuario)
        {
            contexto.USUARIOS.Add(oUsuario);
            contexto.SaveChanges();
        }
        public void Alterar(USUARIOS oUsuario, int codigo)
        {
            oUsuario = contexto.USUARIOS.First(o => o.Id == codigo);
            oUsuario.Descricao = oUsuario.Descricao;
            contexto.SaveChanges();
        }
        public void Excluir(USUARIOS oUsuario, int codigo)
        {
            oUsuario = contexto.USUARIOS.First(o => o.Id == codigo);
            contexto.USUARIOS.Remove(oUsuario);
            contexto.SaveChanges();
        }
    }
}

```

Figura 16 - Exemplo Entity framework

A classe acima é responsável pela persistência assim como a alteração e exclusão de usuário no protótipo, e possui um objeto do tipo DbProjetoEntities, gerado pelo Entity framework com a função de vincular as classe ao banco de dados e dispor métodos para inclusão , consulta e exclusão de objetos.

O método inserir (USUARIOS oUsuario) recebe como parâmetro um objeto do tipo usuário e persiste na base de dados através do método contexto.USUARIOS.add (oUsuario) e grava as informações atrás do método contexto.SaveChanges ();

O método alterar (USUARIOS oUsuario) recebe como parâmetro um e compara através de uma função lambda se existe o mesmo id na base de dados caso atribuindo uma nova descrição ao objeto com o método contexto.SaveChanges();

O método alterar (USUARIOS oUsuario) recebe como parâmetro um objeto do tipo usuário e compara através de um função lambda se existe o mesmo id na base de dados e exclui o objeto com id através do método contexto.USUARIOS.Remove(oUsuario) e grava as informações atrás do método contexto.SaveChanges();

Nota-se em momento algum foi utilizado o uso de consultas sql.

A tela inicial oferece as seguintes opções de cadastro : setores, quadras ,tipo de ocupação ,recipientes, pendencias, tipo de larva, usuário e endereços, depois de tudo cadastrado é possível fazer a inclusão de registro em um determinado endereço.

Com a opção incluir registro o profissional pode incluir ou alterar um registro sem a utilização do equipamento de coleta movel.

Ao executar o aplicativo, o usuário devera informa o login e senha para liberar as outras opções.

Na opção cadastrar endereço é possível verificar as coordenadas de cada cadastro de endereço fornecido pelo sistema móvel de coleta de dados. Abaixo a tela de cadastro de endereços exibindo as coordenadas geográficas (destacada em vermelho) obtidas no momento do cadastro pelo protótipo do aplicativo de coleta de dados.

Id	Data	Rua	Numero	Setor	Quarteirão	Latitude	Longitude	Tipo Ocupação	Usuário
1011	01/11/201...	Fema Laboratorio de Redes...	8	1	1	-22.644163370...	-50.421452522...	Residencial	tiago queiroz
1015	04/11/201...	Tiradentes	1166	1	1	-22.409019470...	-50.576655864...	Residencial	tiago queiroz
-1									

Figura 17 - Tela Cadastro de endereço





A imagem acima é adquirida através da API do Google Mapas através de uma URL que solicita a os serviços Google um mapa com a coordenada centralizada e marcada por um ponto azul apresentado no código abaixo :

```
private void localizar()
{
    String mapUrl = String.Format(CultureInfo.InvariantCulture,
        "http://maps.googleapis.com/maps/api/staticmap?center={0},{1}&size={5}x{6}&markers=color:blue|label:X{0},{1}&zoom={2}&maptype={3}&sensor=true",
        latitude.Replace(",", "."), longitude.Replace(",", "."), zoomLevel, mapType, apiKey, xSize, ySize);
    try
    {
        MapUrlBuilder builder = new MapUrlBuilder();
        builder.CenterCoordinate = coordinate;
        builder.MapType = "SATELLITE";
        builder.ZoomLevel = 16;
        builder.XSize = pcbMap.ClientRectangle.Width;
        builder.YSize = pcbMap.ClientRectangle.Height;
        LocationMap map = new LocationMap(mapUrl);
        pcbMap.Image = map.Map;
    }
}
```

Figura 19 - URL solicitando mapa

Na tela de inclusão de registros, oferece as opções de pesquisar endereço , incluir , altera e registro, mostrar mapa do local e atualizar lista . Abaixo imagem da tela Atividade e Controle:

Id	Data	Recipiente	Quantidade Recipiente	Recipiente Agua	Recipiente Larva	Tipo Larva	Pendencia	Latitude
1022	04/11/2013 ...	Laje	10	5	5	Aegypti	Normal	-22.4094164371...
1023	04/11/2013 ...	Pneu	10	5	5	Aegypti	Normal	-22.4094164371...
1024	04/11/2013 ...	Lata de refrigerante	10	5	5	Aegypti	Normal	-22.4094164371...
1025	04/11/2013 ...	Nenhum	0	0	0	Sem Larvas	Normal	-22.4094593524...
*								

Figura 20 - Tela de registro



### 3.3 Casos de Uso

Uma fase muito importante no processo de desenvolvimento de software consiste no Levantamento de Requisitos, aqui será mostrado os diagramas utilizados para a modelagem dos sistemas. Esta seção é dedicada a os diagramas e documentação dos casos de uso dos protótipos propostos no trabalho

#### 3.3.1 Aplicativo Móvel

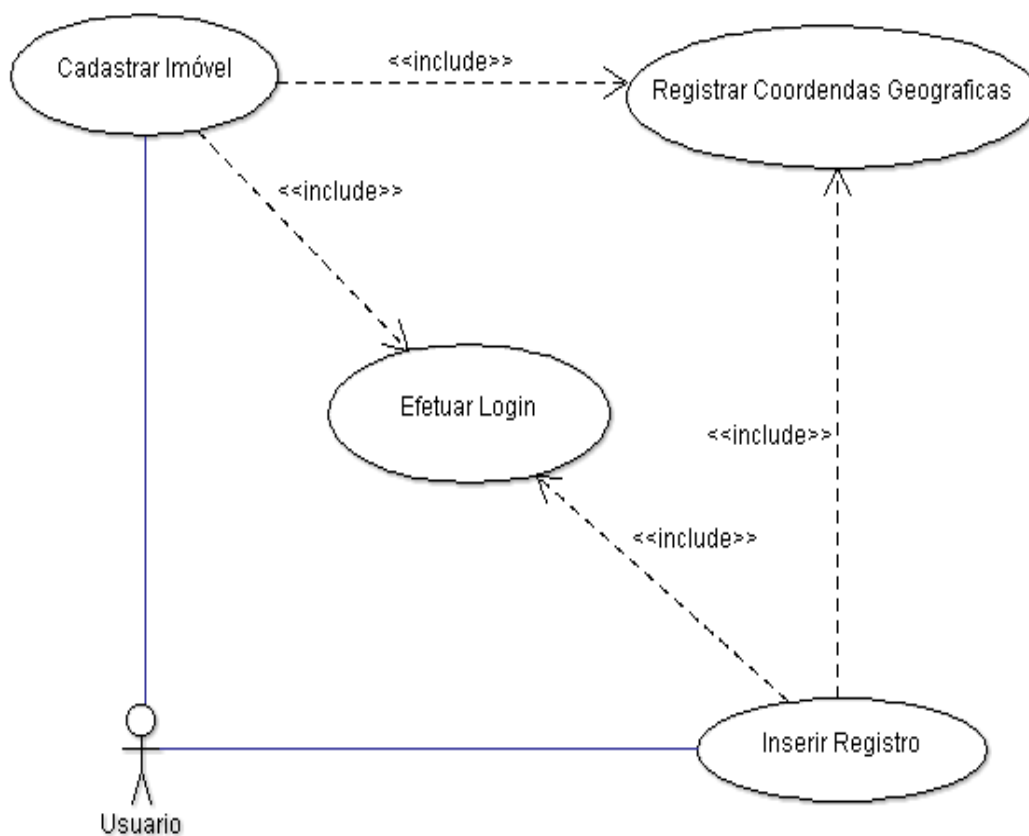


Figura 22 – Diagrama de Casos de Uso Aplicativo móvel

## 3.3.1.1 Efetuar Login

Nome do Caso de Uso	Efetuar Login	
Ator Principal	Usuário (Agente de Saúde)	
Resumo	Este caso descreve o modo em que o usuário faz o login no sistema.	
<b>Fluxo Principal</b>		
	Ações do Ator	Ações do Sistema
	1- Executar o aplicativo e efetuar login no sistema através Nome e senha com auxílio do teclado do dispositivo	Verificar na base de dados se os dados digitados são válidos.
	<b>Restrições / Validações</b>	1 – o usuário deve ser válido. 2 – Os campos login e senha são obrigatórios

## 3.3.1.2 Inserir Registro

Nome do Caso de Uso	Inserir Registro	
Ator Principal	Usuário (Agente de Saúde)	
Resumo	Este caso descreve os passos um agente registrar os dados coletados.	
Pré-condições	1 - O dispositivo possuir um meio de acesso à rede e sistema de posicionamento global (GPS)	
<b>Fluxo Principal</b>		
	Ações do Ator	Ações do Sistema
	1- Selecionar a opção de inserir registro	2- Apresentar formulário de preenchimento
	3- Preencher informações necessárias. E clicar em inserir	4- Validar informações e persistir no banco de dados e retornar um mensagem de sucesso
	<b>Restrições / Validações</b>	1 – um imóvel deve ser informado.

## 3.3.1.3 Cadastrar imóvel

Nome do Caso de Uso	Cadastrar imóvel	
Ator Principal	Usuário (Agente de Saúde)	
Resumo	Este caso descreve os passos para um agente cadastrar um imóvel	
Pré-condições	1- - O dispositivo possuir um meio de acesso à rede dados e sistema de posicionamento global (GPS)	
<b>Fluxo Principal</b>		
Ações do Ator	Ações do Sistema	
1-Selecionar a opção cadastrar imóvel	2- Apresentar formulário de preenchimento	
3 - Preencher os campos de pesquisas de acordo com os campos exigidos e clicar em cadastrar.	4 - O sistema retorna se o imóvel foi cadastrado com sucesso.	
Restrições / Validações	1 – uma quadra deve ser informada	

### 3.2.2 Aplicativo Centralizador

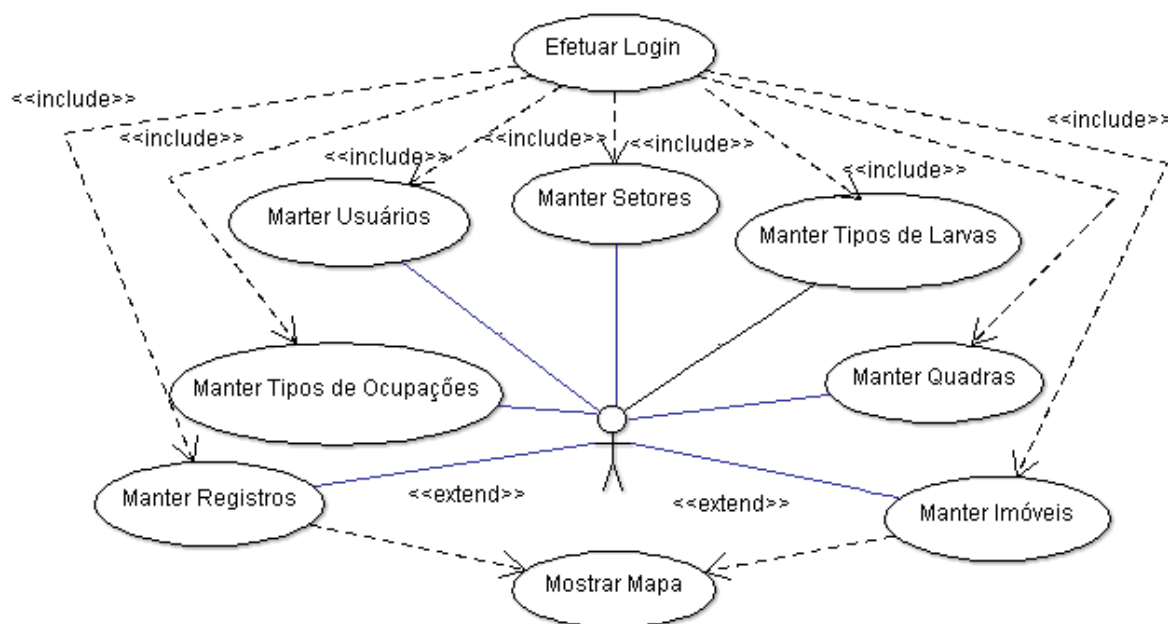


Figura 23 - Casos de Uso do software Centralizador

#### 3.2.2.1 Manter Usuários

Nome do Caso de Uso	Manter Usuário
Ator Principal	Usuário (Gestor de Saúde)
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de usuários, ou seja permite incluir consultar e excluir usuários
Pré-condições	
Pós-condições	
<b>Fluxo Principal</b>	
Ações do Ator	Ações do Sistema
1- Selecionar a opção cadastrar usuário	2- Apresentar a tela de Cadastro de e consulta de Usuários .
3- Se necessário incluir, alterar ou excluir um Usuário.	4- Se necessário gravar as alterações.

## 3.2.2.2 Manter Imóveis

Nome do Caso de Uso	Manter Usuário	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de imóvel, ou seja permite incluir consultar e excluir imóvel.	
Pré-condições		
Pós-condições		
<b>Fluxo Principal</b>		
Ações do Ator	Ações do Sistema	
1- Selecionar a opção cadastrar imóvel	2- Apresentar a tela de cadastro e consulta imóveis.	
3- Se necessário incluir, alterar ou excluir um imóvel.	4- Se necessário gravar as alterações.	

## 3.2.2.3 Manter Registros

Nome do Caso de Uso	Manter Registro	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de Registros, ou seja permite incluir consultar e excluir Registros.	
<b>Fluxo Principal</b>		
Ações do Ator	Ações do Sistema	
1- Selecionar a opção Inserir Registro	2- Apresentar a tela de consulta e inclusão de registro.	
3- Se necessário incluir, alterar ou excluir um Registro.	4- Se necessário gravar as alterações.	



## 3.2.2.4 Manter Setores

Nome do Caso de Uso	Manter Setores	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de Setores, ou seja permite incluir consultar e excluir Setores.	
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1- Selecionar a opção cadastrar Setores	2- Apresentar a tela de cadastro Setores.	
3- Se necessário incluir, alterar ou excluir um Setor.	4- Se necessário gravar as alterações.	
Restrições / Validações	1- Um tipo de ocupação deve ser selecionado para incluir um setor	

## 3.2.2.5 Manter Quadras

Nome do Caso de Uso	Manter Quadras	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de Quadras, ou seja permite incluir consultar e excluir Quadras.	
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1- Selecionar a opção cadastrar Setores	2- Apresentar a tela de cadastro de Quadras.	
3- Se necessário incluir, alterar ou excluir uma Quadra.	4- Se necessário gravar as alterações.	

## 3.2.2.6 Manter Tipos de Larvas

Nome do Caso de Uso	Manter Tipos de Larvas	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de Tipos de Larvas, ou seja permite incluir consultar e excluir Tipos de Larvas.	
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1- Selecionar a opção cadastrar Setores	2- Apresentar a tela de cadastro de e consulta de Tipos de Larvas.	
3- Se necessário incluir, alterar ou excluir um Tipos de Larva.	4- Se necessário gravar as alterações.	

## 3.2.2.7 Manter Tipos de Ocupações

Nome do Caso de Uso	Manter Tipos de Ocupações	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve as possíveis atividades de manutenção do cadastro de Tipos de Ocupações, ou seja permite incluir consultar e excluir Tipo de Ocupação.	
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1- Selecionar a opção cadastrar Setores	2- Apresentar a tela de cadastro e consulta de Tipos de Ocupações.	
3- Se necessário incluir, alterar ou excluir um Manter Tipos de Ocupação.	4- Se necessário gravar as alterações.	

## 3.2.2.8 Efetuar Login

Nome do Caso de Uso	Efetuar Login	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve os passos necessários para registrar o usuário	
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1- Informar Login e Senha	2- Apresentar a tela de cadastro e consulta de Tipos de Ocupações.	
3- Se necessário incluir, alterar ou excluir um Manter Tipos de Ocupação.	4- Se necessário gravar as alterações.	
Restrições/Validações	1 – O usuário tem que ser valido. 2 – Os campos Login e senha são obrigatórios	

## 3.2.2.9 Mostrar Mapa

Nome do Caso de Uso	Mostrar Mapa	
Ator Principal	Usuário (Gestor de Saúde)	
Resumo	Este caso de uso descreve os passos necessários para exibir a localização de um endereço ou registro	
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1- Selecionar um imóvel ou um registro .	2- Mostrar um mapa com o local do imóvel ou registro escolhido	
Restrições/Validações	1 – é obrigatório informar um imóvel ou registro. 2 – O registro ou imóvel deve conter as coordenadas geográficas	

### 3.2.3 Diagrama Entidade relacionamento

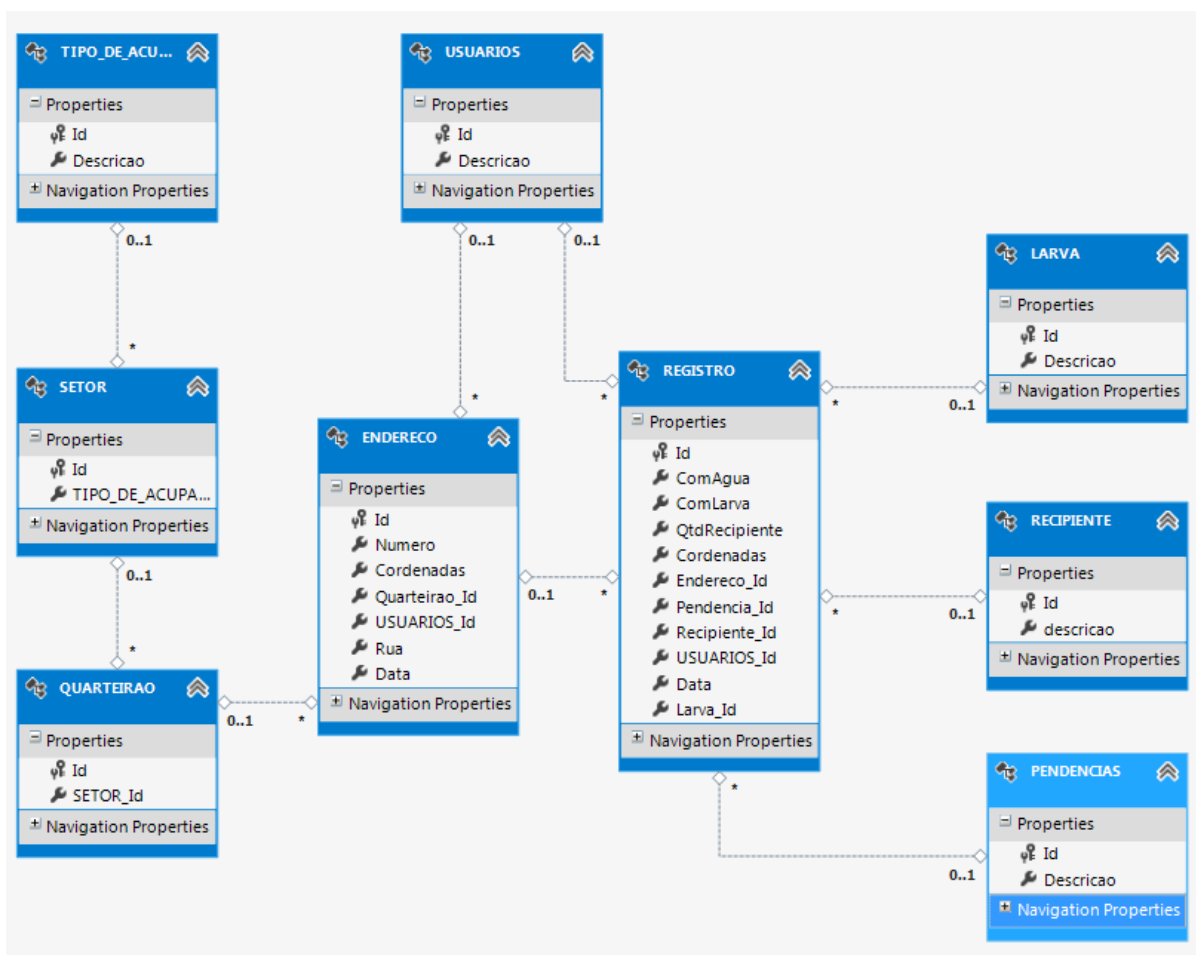


Figura 24 Diagrama de entidade e relacionamento

#### 4. CONCLUSÃO

Dois protótipos foram desenvolvidos para esse trabalho, um para plataforma móvel que tem a finalidade de coletar dados através de um dispositivo com o sistema operacional Google Android, e uma aplicação desktop com a função de reunir todas as informações adquiridas pelo sistema mobile.

Como foi demonstrado nos capítulos anteriores que a plataforma Android possibilita desenvolver aplicações que possuam características de um SIG para dispositivos móveis, e tem a capacidade de integrar vários componentes de hardware como GPS, redes 3G e WIFI oferecendo muitas possibilidades para o desenvolvimento de aplicativos.

Foram apresentadas principais características desta plataforma e as ferramentas de desenvolvimento disponíveis que podem ser utilizadas no desenvolvimento de uma aplicação.

Com isso, o conteúdo apresentado será útil aos interessados no desenvolvimento de aplicativos para dispositivos móveis, especialmente aqueles equipados com a plataforma Android.

Foi apresentada a importância de um serviço web para a integração de sistemas de diferentes plataformas, no caso do projeto em questão foi essencial à comunicação entre o aplicativo Google Android e a base de dados Microsoft SQL que se deu através de serviço web.

Foi mostrado no desenvolvimento do protótipo desktop o uso do Entity framework, com o mapeamento das classes para a manipulação de banco de dados, facilitando assim o desenvolvimento do protótipo.

O Entity Framework aos desenvolvedores trabalhar com dados na forma de propriedades e objetos específicos do domínio como clientes e produtos, sem ter que relacioná-los com as tabelas do banco de dados e as colunas onde os dados estão armazenados.

Isto é possível pela elevação do nível de abstração no qual os desenvolvedores podem trabalhar quando estão tratando com os dados e pela redução do código que é necessário para manter as aplicações orientadas a dados.

A realização do trabalho resultou em um grande aprendizado sobre a

plataforma Android. Além disto, foram assimilados importantes conceitos relacionados com a área de dispositivos móveis. E permitiu a aplicação do conhecimento adquirido durante o curso de graduação na construção de um protótipo, com utilidade prática.

## Bibliografia

Alliance, O. H. (10 de Julho de 2013). *Open Handset Alliance*. Acesso em 10 de Julho de 2013, disponível em Open Handset Alliance:  
<http://www.openhandsetalliance.com/>

Gilberto Câmara, C. D. (2001). *INTRODUÇÃO À CIÊNCIA DA GEOINFORMAÇÃO*. Acesso em 10 de 06 de 2013, disponível em INPE - Instituto Nacional de Pesquisas Espaciais: <http://www.dpi.inpe.br/gilberto/livro/introd/>

Guedes, G. T. (2010). *UML - UMA ABORDAGEM PRÁTICA*. São Paulo: Novatec.

<http://msdn.microsoft.com>. (2013). Acesso em 05 de 05 de 2013, disponível em MSDN MICROSOFT: <http://msdn.microsoft.com/pt-br/aa702903.aspx>

<http://msdn.microsoft.com/>. (06 de 2013). Acesso em 06 de 2013, disponível em MSDN Microsoft: <http://msdn.microsoft.com/pt-br/data/aa937709>

Lacheta, R. R. (2013). *Google Android : aprenda a criar aplicações para dispositivos móveis com Androind SDK*. São Paulo: Novatec.

Pereira, A. P. (03 de 18 de 2009). *Afinal, o que é XML?* Acesso em 2 de 05 de 2013, disponível em <http://www.tecmundo.com.br>:  
<http://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm>

Sampaio, C. (2006). *SOA e Web Service em Java*. Rio de Janeiro : BRASPORT.

Spaki, E., Alves, E., Leal de Farias, J., Lino Neto, J., Althamann, M., & Kono, R. (2008). *Desenvolvimento para Web usando o Visual Studio 2008*. Rio de Janeiro: BASPORT.

*wikipedia*. (01 de 01 de 2013). Acesso em 01 de 2013, disponível em wikipedia:  
[http://pt.wikipedia.org/wiki/Web\\_service](http://pt.wikipedia.org/wiki/Web_service)