

## SISTEMA COLABORATIVO DE INFORMAÇÕES

Daniel Knevez

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
danknevez@terra.com.br

### **Resumo**

Este artigo apresenta o desenvolvimento de um aplicativo, denominado SisColab, baseado no sistema operacional Android, para servir de alternativa ao processo de notificação de ocorrências no sistema de iluminação pública, de modo a permitir que o cidadão munido de um *smartphone*, possa comunicar ao gestor do serviço o problema encontrado. O aplicativo irá se basear nas informações de geolocalização e fotos para enviar com agilidade e acuidade as informações coletadas. Para o desenvolvimento do aplicativo foram utilizadas as linguagens de programação Java e XML. Outro quesito do aplicativo está no aspecto colaborativo, em que o cidadão tem papel importante para a manutenção da qualidade do serviço de iluminação, diminuindo escalas intermediárias e burocracias, normalmente existentes nesses processos.

**Palavras-chave:** Android, colaborativo, geolocalização, *smartphone*

## COLLABORATIVE INFORMATION SYSTEM

### **Abstract**

*This article presents the development of an application, named SisColab, based on the Android operating system, to figure as an alternative to the occurrence notification in public illumination system, therefore allowing the citizen with a smartphone to communicate to the service manager the problem found. The application will be based on geolocation and photographic information to send with agility and acuity of the information gathered. For application development Java/XML programming languages were used. Another item of the application is the collaborative aspect, in which the citizen have an important role in maintaining the quality of the lighting service, reducing bureaucracy and intermediate scales, usually existing in these processes.*

**Keywords:** Android, collaborative, geolocation, *smartphone*

## **1 INTRODUÇÃO**

Uma dificuldade enfrentada pelas entidades responsáveis por prestar serviço à população é descobrir de forma ágil e eficiente onde estão localizados os problemas que atingem os cidadãos. Uma das maneiras possíveis seria que esses órgãos mantivessem serviços de vistoria periódica sobre aquilo que está sob sua jurisdição, designando funcionários para este serviço, maneira esta que oneraria em demasia o custo de tal manutenção. Outra maneira seria contar com o auxílio dos próprios usuários do serviço em informar os problemas por ele encontrados.

Ocorre que atualmente o cidadão que queira relatar algum problema ou ocorrência para a administração, deve se dirigir até o órgão responsável, o que pode ocasionar certa desmotivação ao cidadão em noticiar, ao ente público, incidentes ocorridos.

Com o desenvolvimento desse aplicativo almeja-se melhorar o canal de comunicação entre as partes, gerando economia de tempo ao cidadão e disponibilizando dados mais acurados a respeito do fato gerador, tornando mais célere o atendimento e reparação do acontecido.

O sistema permitirá ao usuário notificar, ao órgão público responsável, a ocorrência de alguma irregularidade, informando a sua localização exata, através de ferramenta de geolocalização por GPS (*Global Positioning System*), somada a um breve relato do ocorrido, podendo anexar também imagens fotográficas.

Inicialmente pretende-se focar o sistema para ocorrências no sistema de iluminação pública do município, mas podendo posteriormente ser expandido para outros tipos de ocorrências, sem que se exijam grandes modificações e ajustes no software.

## **2 REFERENCIAL TEÓRICO**

### **2.1 Sistema Colaborativo**

O modelo de um software colaborativo consiste em proporcionar para as organizações um desempenho mais ágil com o recebimento de informações dos cidadãos de uma cidade, visando facilitar a execução dos trabalhos em grupo e diminuindo as etapas de comunicação entre a sociedade e a administração municipal.

#### **2.1.1 Modelo colaborativo comunitário**

O modelo colaborativo comunitário nasceu a partir da compreensão que o Estado, sozinho, não consegue atender de pleno os anseios e necessidades do cidadão.

Nos bairros das cidades e em pequenas comunidades rurais em todo o mundo existe um movimento crescente que está centrado no cidadão. Esse movimento reflete uma nova percepção de responsabilidade local por determinar e produzir o futuro. O movimento está crescendo por dois motivos: primeiro, cada vez mais pessoas estão acreditando na democracia no mundo inteiro e esse movimento é a expressão local do compromisso para com a democracia. Segundo, as pessoas que moram nas comunidades estão começando a entender as limitações que as grandes instituições têm na produção do seu bem-estar. Vêm cada vez mais provas de que sistemas comerciais, governamentais ou militares não são capazes de "produzir" comunidades saudáveis, embora possam ajudá-las. (CURITIBA, 2002, p.6)

A partir dessas percepções, o presente trabalho visa utilizar a própria população para auxiliar as prefeituras envolvidas, de modo que os problemas serão reportados através do software desenvolvido e disponibilizado para instalação em telefones celulares tipo smartphones.

## **2.2 Serviços Públicos**

Uma definição sobre serviços públicos vem de Aragão:

“Serviços públicos são as atividades de prestação de utilidades econômicas a indivíduos determinados, colocados pela Constituição ou pela Lei a cargo do Estado, com ou sem reserva de titularidade, e por ele desempenhadas diretamente ou por seus delegatários, gratuita ou remuneradamente, com vistas ao bem-estar da coletividade”. (ARAGÃO, 2007, p.157)

São aqueles serviços convenientes à sociedade, que a Administração os presta diretamente ou permite que sejam prestados por terceiros dentro das condições que estabelece e sob seu controle.

Estes serviços facilitam a vida do indivíduo, proporcionando mais conforto, mas não são essenciais. Podem ser chamados de serviços pró-cidadão.

Exemplos: transporte coletivo, gás, telefone, energia elétrica.

## **2.3 Responsabilidades Públicas**

O Art. 175 da Constituição Federal incumbe ao Poder Público, na forma da lei, diretamente ou sob regime de concessão ou permissão, sempre através de licitação, a prestação de serviços públicos.

Mello (2001) em suas abordagens considera que “O Estado tem como seu dever assumir certas atividades denominadas serviços públicos, pois não é socialmente desejável que elas sejam regidas pela iniciativa privada.”, Mello (2001) ainda destaca que “Essas atividades são destinadas a satisfazer a coletividade em geral, pois tratam-se de prestações de utilidade e comodidade material”.

A partir destes enfoques, verifica-se que o estado é responsável por manter os serviços públicos.

## **2.4 Direitos do Cidadão**

Os serviços públicos permitem que o particular exija a sua prestação, caso esteja na área por eles abrangida.

Para Amaral (2006) regular a prestação dos serviços públicos também é direito do usuário, dessa maneira compete igualmente ao cidadão zelar pela boa prestação do serviço público, auxiliando naquilo que lhe houver cabimento, utilizando-se das formas e meios disponíveis para tal.

A percepção do usuário e sua manifestação poderão apontar caminhos para melhorar a prestação de serviços, diminuindo o descrédito e aumentando a confiança do usuário (HELD FILHO, 1999).

## **2.5 Qualidade do Atendimento**

Kotler (2002) afirma que a qualidade é a totalidade dos atributos e características de um produto ou serviço que afetam sua capacidade de satisfazer necessidades declaradas ou implícitas.

Desta maneira nota-se hoje cada vez mais a necessidade de controlar a qualidade dos serviços prestados, onde o usuário passa a ser o elemento central e a qualidade é definida também de acordo com a percepção do mesmo, como a previsão e a superação das suas expectativas.

## **2.6 Smartphone**

*Smartphone*, em tradução livre, significa telefone inteligente. É um telefone celular, com recursos semelhantes ao de um computador, permitindo ao usuário executar programas que se adaptam as suas necessidades. A maioria deles possui nativamente alguns recursos extras daqueles encontrados nos computadores, tais como câmera fotográfica embutida, sistema de geolocalização e envio de mensagens.

Conforme Batista (2011): “o *smartphone* é um telefone inteligente que surgiu de uma combinação entre celulares e agendas eletrônicas”, ainda para o autor, por possuírem seu sistema operacional escrito em código aberto, qualquer pessoa pode desenvolver, disponibilizar e utilizar seus programas.

Figura 1 – Modelos Smartphone

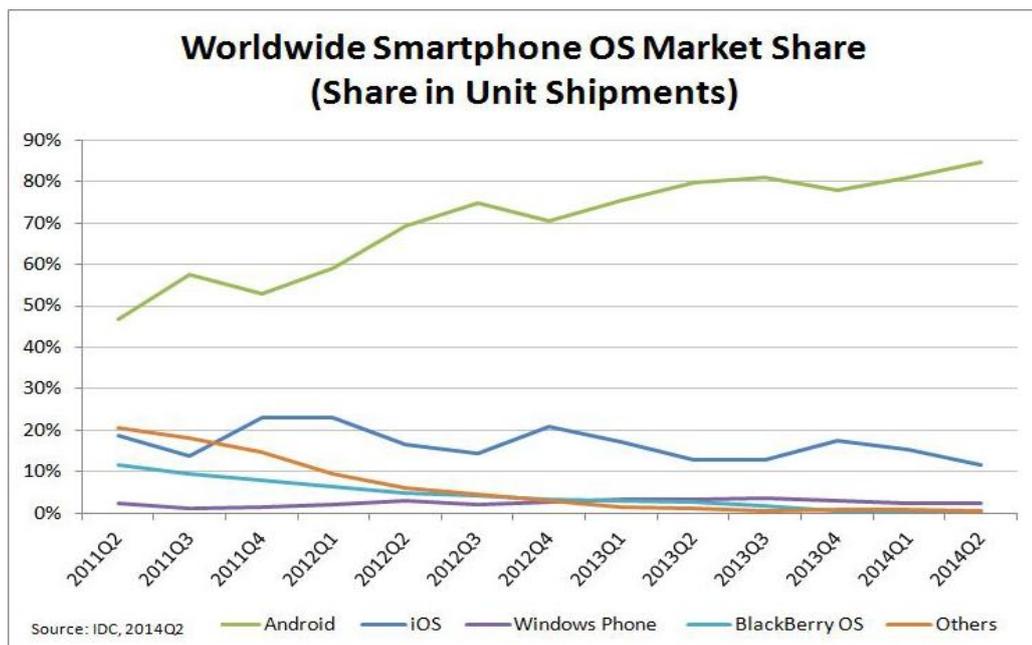


Fonte: mashable.com, 2014

O dicionário PRIBERAM (2014) define *smartphone* como: “Celular com conectividade e funcionalidades semelhantes às de um computador pessoal”

Um *smartphone* possui características de computador, pois são capazes de conectar a internet, ampliando assim, ainda mais a sua gama de recursos. Existem diversos modelos e marcas de *smartphones*, bem como sistemas operacionais diferentes que são utilizados pelos aparelhos, tais como o iOS da Apple, o Blackberry para *smartphone* do mesmo nome, o Windows Mobile para alguns celulares da marca Nokia e o Android da Google, que atualmente é o sistema mais popular utilizado por smartphones, conforme descrito na figura 2, em que aproximadamente 85% dos aparelhos são vendidos com sistema Android

Figura 2 – Vendas de Sistemas Operacionais nos Smartphones



Fonte: <http://br.idclatin.com>, 2014

O gráfico anterior foi fato determinante para a escolha do sistema operacional Android para o desenvolvimento deste projeto, pois com ele é possível atingir um maior volume de usuários.

## 2.7 Geolocalização e Geoinformação

A geolocalização é o termo que define a técnica que permite identificar a exata localização de um dispositivo dotado de algum tipo de uma conexão internet através do rastreamento do IP ou de outros métodos de identificação como GPS.

A geolocalização se baseia em dados gerados por endereço MAC, RFID (identificação de radiofrequência), conexão wi-fi (sem fio) e GPS. Smartphones utilizam o GPS integrado para envio de informações de localização física geográfica.

Figura 3 - Esquema de Geolocalização



Fonte: <http://www.tudocelular.com/curiosidade>, 2014

Conforme Câmara (2001), o objetivo da geolocalização é a geoinformação, que corresponde a uma informação unida com um atributo geográfico. Desta maneira a informação contém um endereço e carrega consigo as coordenadas (latitude, longitude e altitude) do local designado.

## 2.8 Plataforma Android

Para Lecheta (2009), o Android possui uma interface visual rica e flexível, e com ambiente de desenvolvimento inovador.

O Android é um software de fonte aberta, que utiliza na plataforma de desenvolvimento a linguagem Java. O que permite que os aplicativos desenvolvidos nesta linguagem sejam executados de maneira mais ágil.

“O Android tem muitos diferenciais interessantes e uma arquitetura realmente flexível focada na integração de aplicações. Não existe diferença entre uma aplicação nativa e outra desenvolvida por terceiros.” (LECHETA, 2009, p.25)

Figura 4 - BugDroid - logotipo Android



Fonte: Autor, 2014

### 2.8.1 Arquitetura

A plataforma Android possui a arquitetura definida conforme figura a seguir:

Figura 5 - Arquitetura da plataforma Android



Fonte: OHA, 2014

Na parte inferior da arquitetura encontra-se o Kernel Linux, responsável pelo gerenciamento das funções de hardware; as rotinas de aplicação e controle são executadas por uma JVM (máquina virtual Java) otimizada especialmente para dispositivos móveis denominada Dalvik. A plataforma possui uma coletânea de bibliotecas C/C++ de vídeo, áudio, banco de dados e outras, contendo os controles e definições para os mesmos serem definidos e configurados no *framework* de aplicação.

O Android já vem equipado com diversas aplicações, escritas em linguagem Java. Dentre outras, há despertador, navegador internet, gerenciador de memória entre outros desenvolvidos por terceiros, estas aplicações estão disponíveis na parte superior e mais acessível da estrutura do sistema Android.

## 2.9 Aplicativos Android

Como descrito anteriormente, as aplicações para Android são escritas em linguagem Java. O código fonte desenvolvido, é compilado em bytecodes reconhecíveis pelo *Dalvik Virtual Machine*, o resultado desta compilação é empacotado juntamente com as bibliotecas necessárias para a execução em um arquivo com extensão APK, sendo este arquivo instalado pelo próprio usuário, ou disponibilizado pelo desenvolvedor diretamente na loja de aplicativos do fabricante, chamada de Google Play, para distribuição a qualquer usuário que tenha acesso a esta loja, podendo ser gratuito ou mediante pagamento.

## 3 METODOLOGIA

O sistema foi desenvolvido tomando por base a metodologia em cascata, em que existem fases sequenciais, nas quais as seguintes só podem ser iniciadas após a finalização da anterior.

Conforme Pressman (2002), para a utilização do modelo em cascata, o projeto deve seguir uma série de passos ordenados, em que ao final de cada fase, o projeto passa por uma revisão.

Para Sommerville (2007), a modelagem em cascata fornece uma sequência ordenada de etapas de desenvolvimento de modo a garantir a adequação do projeto e revisões de documentação.

A figura a seguir demonstra as etapas do desenvolvimento em cascata utilizadas neste projeto.

Figura 6 – Modelo de Desenvolvimento Utilizado



Fonte: Autor, 2014

### 3.1 Análise

Nesta etapa foram definidos os requisitos do sistema bem como suas funcionalidades, os quais são descritos a seguir:

- Desenvolvimento do sistema para sistemas operacionais Android;
- Obrigatoriedade de cadastro contendo: nome completo, endereço e CPF;
- Definição pelo Serviço de Iluminação Pública como foco inicial do projeto;
- Relacionados os principais tipos de ocorrência: lâmpada queimada, lâmpada sempre acesa, lâmpada piscando e demais problemas;
- Possibilidade de inserir problema não listado anteriormente;
- Utilização de geolocalização para informar local da ocorrência;
- Possibilidade de envio de fotografia anexada à ocorrência;
- Geração de número serial único baseado nas informações do usuário e da ocorrência;
- Finalização do registro de ocorrência com envio do e-mail para o órgão responsável.

Definiu-se também que o serviço de recebimento das ocorrências seria objeto de desenvolvimento futuro, pois depende de *lay-out* disponibilizado pelos responsáveis.

### 3.2 Modelagem

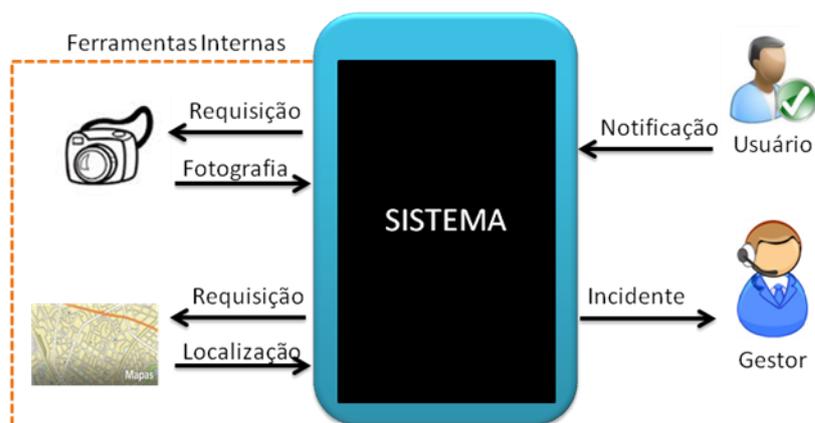
Para Pressmann (2002), esta etapa busca representar os requisitos por meio de um conjunto de representações gráficas ou descritas, visando facilitar a criação do código fonte e sua respectiva implementação.

Nesta etapa também são especificados todos os componentes necessários para o funcionamento do sistema.

Para uma melhor descrição desta etapa foram utilizados os diagramas de UML (*Unified Modeling Language*), muito utilizados para facilitar a comunicação e interpretação em todo projeto.

Tendo por base os requisitos levantados na fase anterior foram criados alguns diagramas, descritos a seguir:

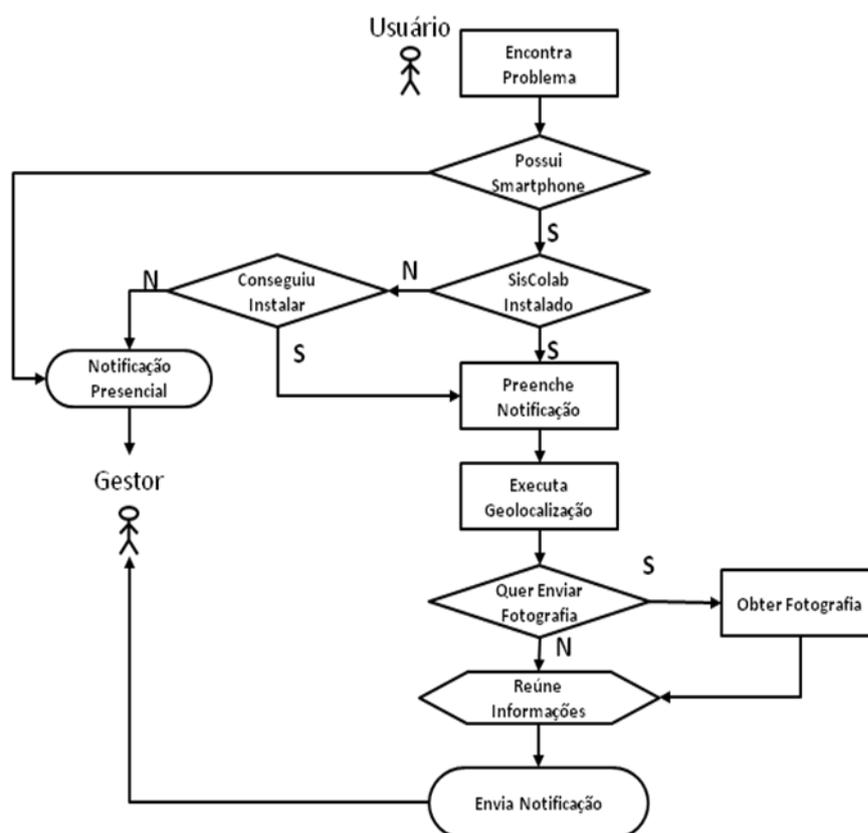
Figura 7 – Diagrama de Contexto



Fonte: Autor, 2014

Neste diagrama estão expressos os principais atores do projeto, bem como o contexto de utilização de cada um deles.

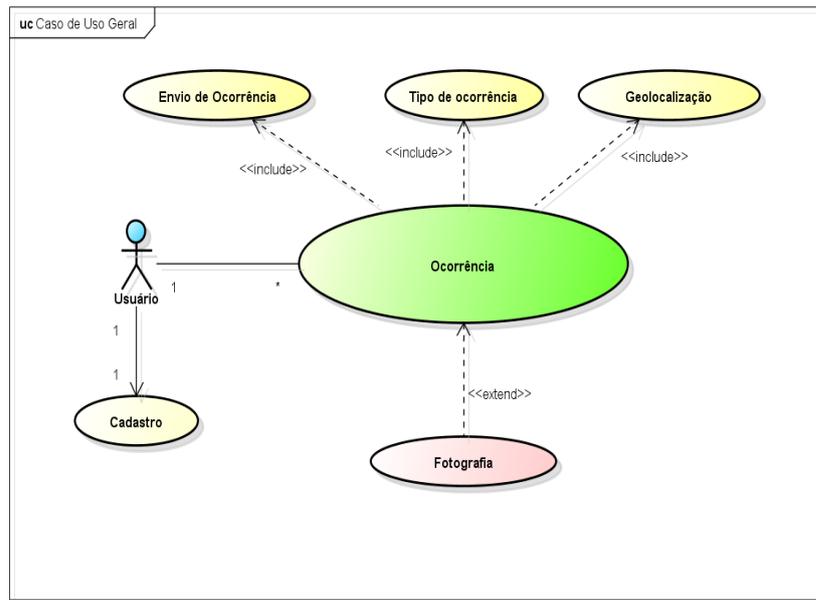
Figura 8 – Diagrama de Atividades



Fonte: Autor, 2014

O diagrama 8 expõe o fluxo de atividades que o usuário deve desenvolver para notificar uma ocorrência com a utilização do aplicativo.

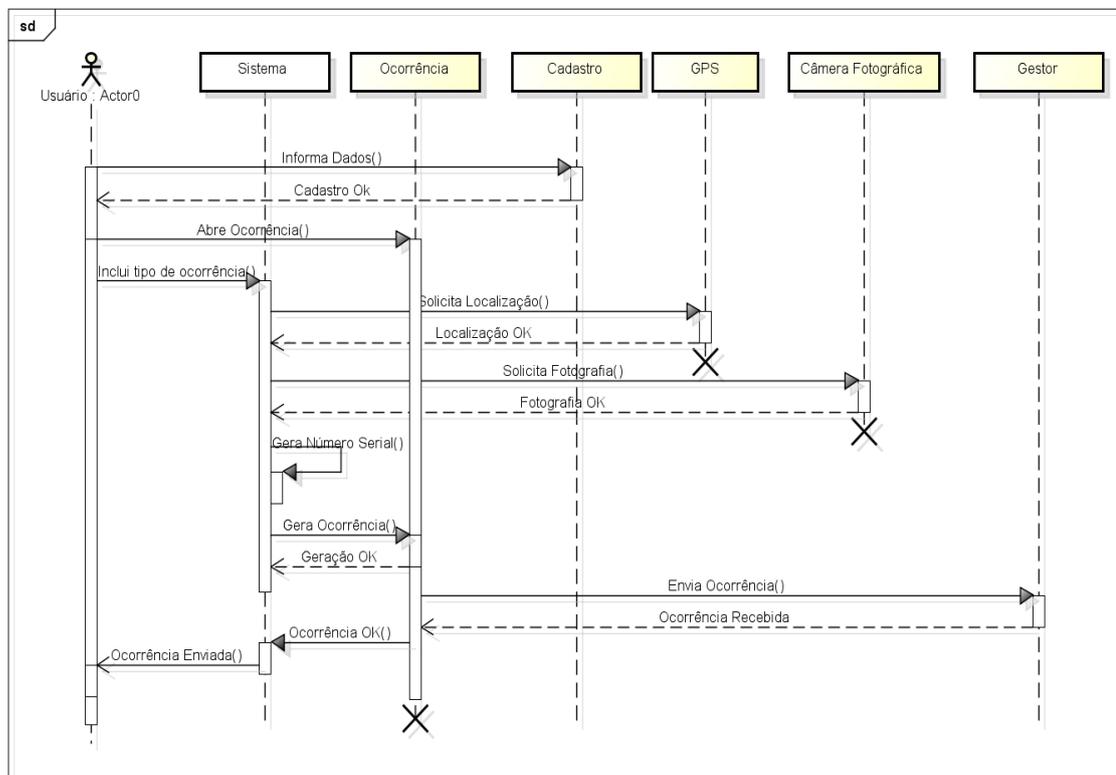
Figura 9 – Diagrama Conceitual de Caso de Uso Geral



Fonte: Autor, 2014

No diagrama 9 está definido o caso de uso geral em que o usuário executa as ações de cadastro e ocorrência, tendo como atividades obrigatórias a definição do tipo, o serviço de geolocalização e o envio da ocorrência e como opcional o envio de fotografia do evento.

Figura 10 – Diagrama de Sequência



Fonte: Autor, 2014

O diagrama de sequência demonstra a execução das atividades de registro de ocorrência dentro da linha de tempo.

### 3.3 Desenvolvimento

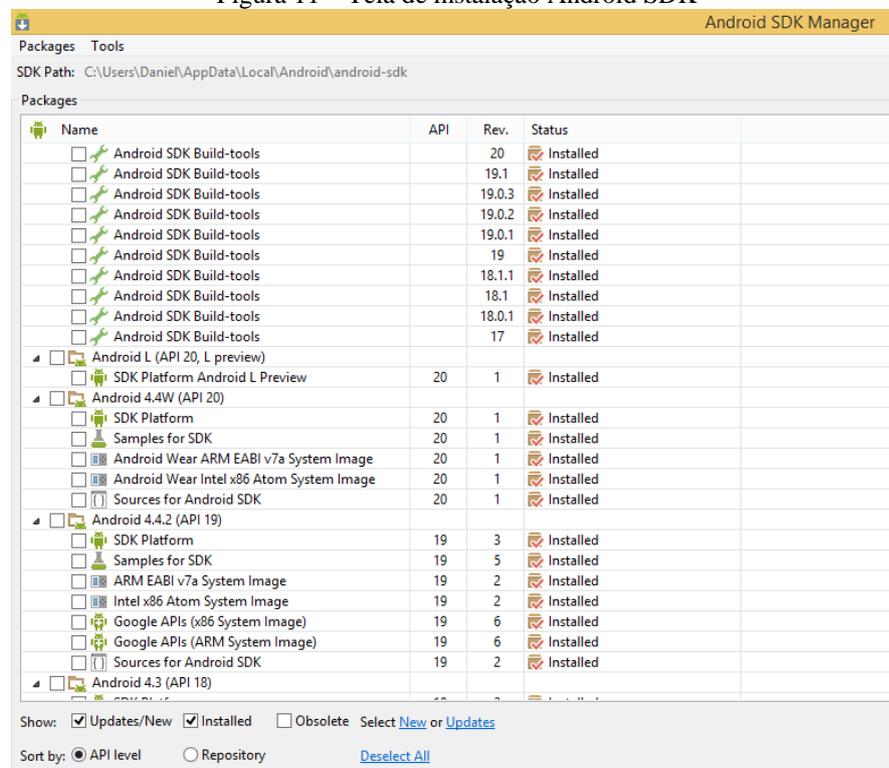
Para o desenvolvimento do aplicativo optou-se por utilizar algumas interfaces, tais como descrito a seguir, bem como a linguagem Java, que é nativa do Android, em conjunto com a linguagem de marcação XML.

#### 3.3.1 Ferramentas Utilizadas

- **Android SDK**

O Android SDK fornece as bibliotecas de API e as ferramentas necessárias para construir desenvolvedor, teste e depuração de aplicativos para o Android.

Figura 11 – Tela de instalação Android SDK



Fonte: Autor, 2014

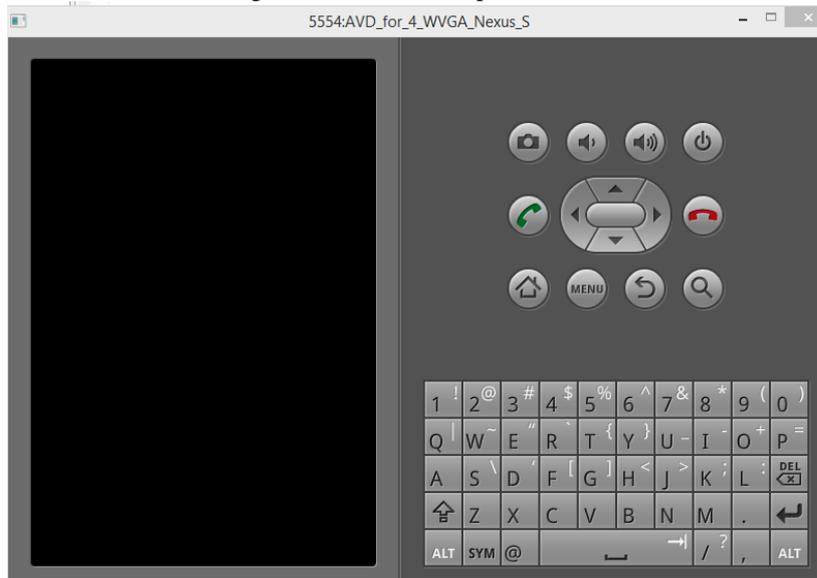
- **ADT Android Development Tools**

O Android Development Tools (ADT) é um plugin para o Eclipse IDE que é projetado para fornecer o ambiente integrado no qual pode-se construir aplicativos Android.

- **AVD Android Virtual Device**

O Android Virtual Device (AVD) é um simulador de ambiente Android, que simula as condições de um dispositivo real muito útil para dar fluidez no processo de edição e validação do software.

Figura 12 – Tela Principal do AVD

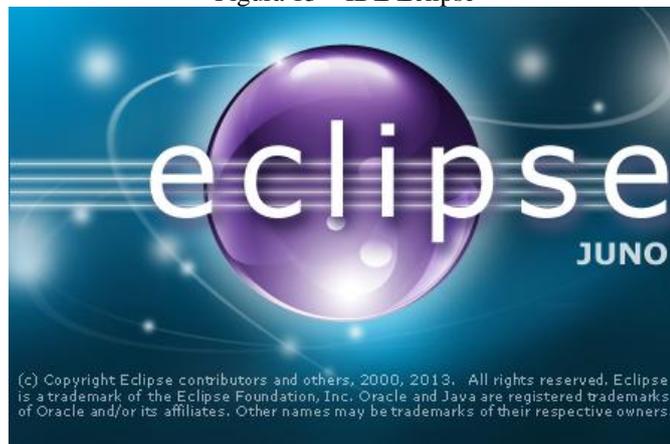


Fonte: Autor, 2014

- **Eclipse**

O Eclipse é uma IDE para desenvolvimento de varias linguagens, e por ser desenvolvido em Java esta é sua linguagem principal, motivo pela qual foi escolhida para ser a plataforma principal, conforme a empresa Google esta IDE faz parte do kit de desenvolvimento de software recomendado para desenvolvedores Android.

Figura 13 – IDE Eclipse



Fonte: Autor, 2014

- **SQLite**

O SQLite é um banco de dados transacional nativo do sistema Android disponibilizado gratuitamente e que não depende de instalação de servidores e não possui necessidade de configurações.

### 3.3.2 Componentes do Android

Os componentes utilizados na plataforma Android possuem acesso às APIs utilizadas pelo núcleo da plataforma. A reutilização dos componentes foi projetada de modo a simplificar a publicação e utilização das capacidades de cada um destes componentes. Estes componentes podem ser de quatro tipos:

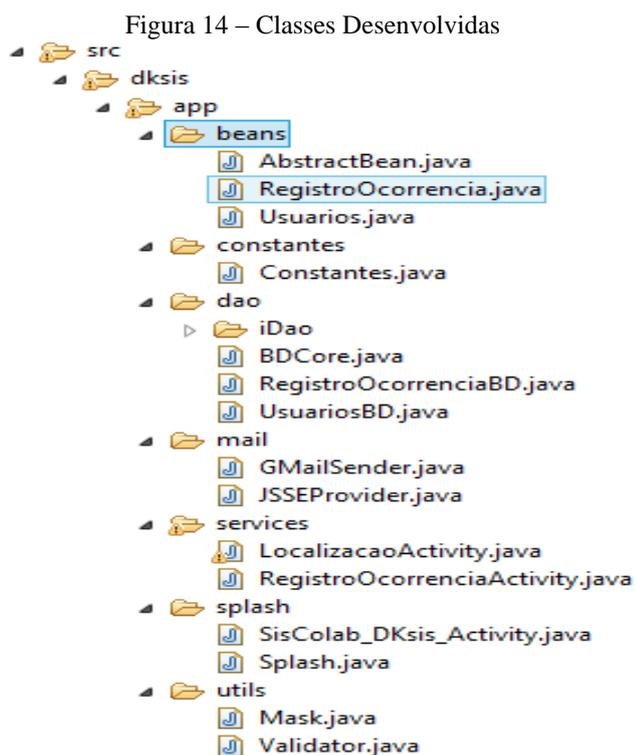
Quadro 1 – Principais componentes de Programação Android

<b>Activities</b> – Funcionam como mediadores que definem como as informações serão apresentadas ao usuário, além de controlar o fluxo da aplicação. Elas podem interagir com o usuário e trocar informações com outras <i>activities</i> ou <i>services</i> .
<b>Services</b> – É utilizada para executar um processamento em segundo plano por tempo indeterminado. Não precisa interagir com o usuário e conseqüentemente não precisa de interface gráfica.
<b>Broadcast Receivers</b> – São componentes que ficam aguardando a ocorrência de um determinado evento, pode-se entender como evento a inicialização do sistema operacional, uma chamada de voz, a chegada de um SMS, um evento disparado por uma aplicação.
<b>Content Providers</b> - São os compartilhadores de conteúdo entre as aplicações, uma aplicação pode requisitar informações de outra, por exemplo, uma aplicação pode receber dados da lista de contatos que é nativa do Android, e com base nesses dados, realizar algum processamento.

Fonte: Adaptado LECHETA, 2009

### 3.3.3 Classes Desenvolvidas

A figura 14 demonstra as classes criadas no projeto, cujas descrições serão explanadas a seguir:



Fonte: Autor, 2014

- AbstractBean – responsável pela criação de número serial utilizado no envio da mensagem, de modo a torná-la única;
- RegistroOcorrencia – responsável pela criação dos itens da ocorrência;
- Usuarios – responsável pela criação dos componentes do cadastro do usuário;
- Constantes – classe simples utilizada para definição dos itens de registro e câmera;
- BDCore – responsável pela manutenção do banco de dados do aplicativo, contendo as tabelas de clientes e ocorrências;
- RegistroOcorrenciaBD – responsável pelo armazenamento dos itens da ocorrência;
- UsuariosBD – responsável pelo armazenamento dos itens do cadastro;
- GMailSender – responsável pelo envio do email contendo os componentes da ocorrência;
- JSSEProvider – responsável pela configuração do provedor Java Secure Socket Extension
- LocalizacaoActivity – responsável pela criação da tela referente a geolocalizacao;
- RegistroOcorrenciaActivity – responsável pela criação da tela referente ao registro de ocorrência;
- SisColab\_DKsis\_Activity – Responsável pela criação da tela de cadastro e verificação da existência prévia de cadastro;
- Splash – tela inicial do aplicativo;
- Mask – responsável pela substituição de caracteres digitados nos campos do cadastro;
- Validator - responsável por validar os campos de CPF e de email.

### 3.4 Testes e Resultados

#### 3.4.1 Testes

Para Dias Neto (2007) “testar um software significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado”, partindo desta definição o conceito de teste remete a uma validação prévia do sistema desenvolvido antes de executar a etapa seguinte.

Para a execução dos testes existem essencialmente duas modalidades e cada uma destas modalidades tem suas vantagens e desvantagens:

- Modalidade Manual: onde o software é entregue a testadores para posterior *feedback*;
- Modalidade Automatizada: onde são utilizados *frameworks* construídos para essa finalidade.

Conforme Capani (2013), não existe uma técnica de teste ideal que inclua todas as possibilidades, e a melhor abordagem é a combinação dessas técnicas, na figura a seguir pode-se verificar um comparativo entre as modalidades.

Figura 15 – Vantagens e Desvantagens das Modalidades de testes

	✓	✗
Teste Manual	Usabilidade	Custo, tempo, confiabilidade
Emuladores e Simuladores	Custo (início do desenv)	Fidelidade
Virtualização de Serviços	Minimiza ambiente de testes, Custo, Velocidade, Independência das Camadas – nos dois sentidos Possibilita integração, Entrega contínua	Componente adicional, tecnologia, skills, fidelidade
Automação de Testes Funcionais	Cobertura dos testes, Confiabilidade e Repetibilidade, Possibilita integração, Entrega contínua	Fases finais do desenvolvimento, Depende da estabilização da aplicação
Cloud Test	Cobertura de devices	Custo, Complexidade

Fonte: Capani, 2013

Existe uma variedade de testes que podem ser executados para validar o software tais como descrito a seguir:

Quadro 2 – Tipos de Testes

<b>Teste de Unidade:</b> também conhecido como testes unitários. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.
<b>Teste de Sistema:</b> avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.
<b>Teste de Aceitação:</b> são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.
<b>Teste de Integração:</b> visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto.
<b>Teste de Regressão:</b> Teste de regressão não corresponde a um nível de teste, mas é uma estratégia importante para redução de “efeitos colaterais”. Consiste em se aplicar, a cada nova versão do software ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

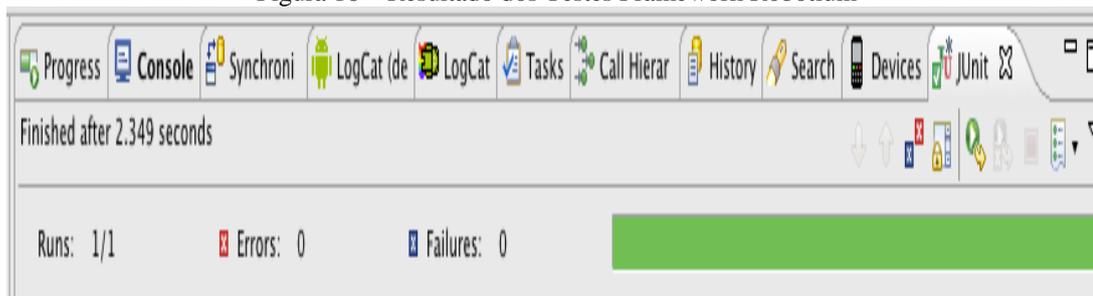
Fonte: Dias Neto, 2007

Quando se está desenvolvendo para aplicativos móveis, utilizam-se primariamente na validação do *software*, os testes unitários e os de interface gráfica.

Os testes unitários foram executados através da ferramenta denominada Robotium, que é um framework open-source de testes para a plataforma Android que tem suporte completo para aplicações nativas e híbridas.

Conforme a figura 16, após a execução da ferramenta de testes, os resultados obtidos não apresentaram erros validando a construção do software.

Figura 16 – Resultado dos Testes Framework Robotium



Fonte: Autor, 2014

Os testes de interface gráfica foram executados dentro da modalidade manual, no qual o aplicativo foi instalado em diferentes configurações de aparelhos Android:

- Tamanho da tela (10.1, 4.2 e 3.4 polegadas);
- Versão do Sistema operacional (*KitKat*, *Jelly Beans*, *Ice Cream Sandwich*);
- Capacidade de memória (0,5 a 4MB);
- Conexão *wifi* 10Mb e conexão celular 4G.

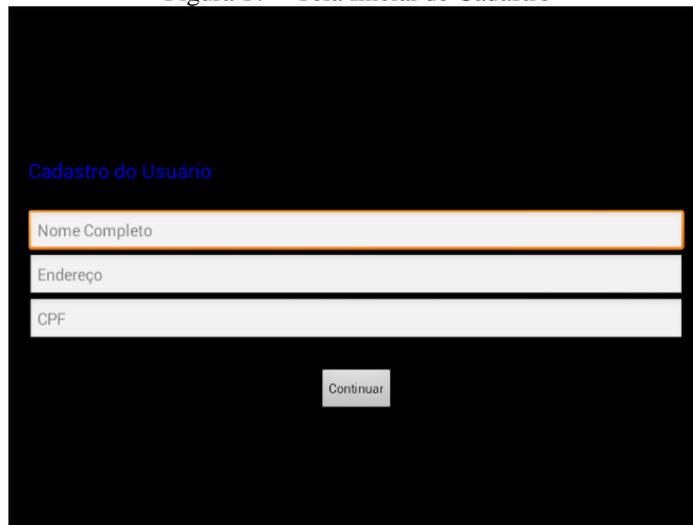
Nesta modalidade de teste, foram avaliados os itens constantes a cada tela. Conjuntamente com esse teste foi verificada a interação com os componentes internos do smartphone: giroscópio, GPS, câmera fotográfica. Em todas as combinações e testes executados o *software* foi executado conforme as especificações atendendo os requisitos.

### 3.4.2 Resultados

Esta seção apresenta os resultados obtidos com o desenvolvimento do software ao longo do projeto, com apresentação de algumas interfaces para o usuário e exemplos de uso.

Quando o usuário acessa o aplicativo pela primeira vez, a tela da Figura 17 é apresentada.

Figura 17 – Tela Inicial de Cadastro



Cadastro do Usuário

Nome Completo

Endereço

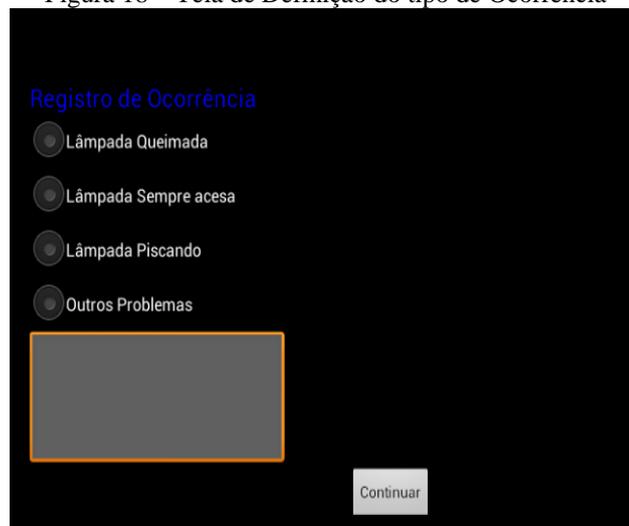
CPF

Continuar

Fonte: Autor, 2014

Esta tela é apresentada somente quando no primeiro acesso ao aplicativo, e tem por objetivo cadastrar o usuário no sistema com suas informações básicas de identificação. Nas utilizações seguintes o sistema será sempre direcionado para a tela da figura 18.

Figura 18 – Tela de Definição do tipo de Ocorrência



Registro de Ocorrência

Lâmpada Queimada

Lâmpada Sempre acesa

Lâmpada Piscando

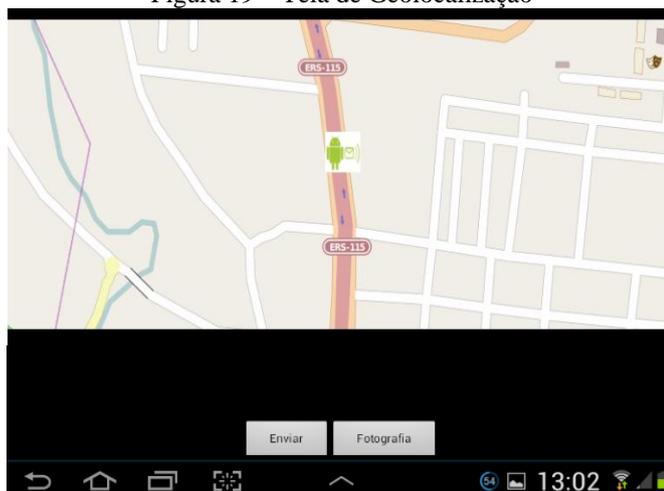
Outros Problemas

Continuar

Fonte: Autor, 2014

A interface apresentada na Figura 18 é a que possibilita ao usuário informar o tipo do problema a ser relatado para o órgão responsável. A tela fornece as opções mais comuns de ocorrência, sendo que em caso de o problema ser uma situação não listada, existe a possibilidade de informar outros tipos de problemas, no campo texto disponibilizado.

Figura 19 – Tela de Geolocalização



Fonte: Autor, 2014

Na figura 19 está expressa a tela em que o sistema de geolocalização do smartphone reconhece a localização do usuário dentro do mapa. Nesta etapa também está presente a opção de envio de uma fotografia, se o usuário assim entender necessário para melhor descrever o problema. O passo seguinte é o envio, propriamente dito, da ocorrência ao gestor do serviço fato esse que encerra todo o processo e o aplicativo.

Figura 20 – Modelo de E-mail enviado



Fonte: Autor, 2014

## 4 CONCLUSÃO

A utilização do aplicativo mostrou-se uma boa alternativa para a execução do processo proposto, pois disponibilizou com facilidade um novo meio de interação entre a comunidade e o órgão responsável pela manutenção do sistema.

A intenção do aplicativo não é a de substituir o modelo atual de registro de ocorrências, pois parte da população ainda não possui acesso a algum tipo de *smartphone*, sem contar com a barreira psicológica, visto que muitos cidadãos ainda têm restrições à ausência de comprovante físico de sua interação.

Na criação do aplicativo optou-se por uma interface simples com pouca necessidade de interação, em que o processo completo desde a instalação do *software* até o envio da ocorrência para o gestor emprega em torno de dois minutos, neste caso demonstrando a agilidade que se pretendeu.

Como o projeto focou apenas na área de iluminação pública, e que a gama de serviços prestados à população é vasta, existe uma ampla possibilidade de expansão do aplicativo para esses setores, bem como na melhoria da interface gráfica, desde que isso não implique em perda de agilidade no processo como um todo, visto que esse é um dos atrativos do projeto.

Outra implementação possível e que traria praticidade para o gestor do processo, é o desenvolvimento de rotina para automatização do recebimento das ocorrências. Porém, para desenvolver esta funcionalidade, seria indicado o contato com o gestor, a fim de se desenvolver a solução de acordo com as especificações fornecidas.

Mais uma possibilidade de melhoria é aquela relacionada à abrangência do aplicativo em termos de sistema operacional do *smartphone*, pois mesmo que o sistema Android seja o modelo dominante, ainda existem outros que possuem público cativo e que provavelmente gostariam de poder participar do processo colaborativo que o projeto se refere.

## REFERÊNCIAS

AMARAL, Antônio Carlos Cintra do. **Distinção entre Usuário de serviço Público e Consumidor**. Revista Eletrônica de Direito Administrativo, nº 6, mai/jun/jul de 2006.

ARAGÃO, Alexandre Santos de. **Direito Dos Serviços Públicos**. Editora Forense. 1a Ed. 2007. Rio de Janeiro p.157

BATISTA, Gabriela. **Saiba tudo sobre Smartphones**. Disponível em: <<http://www.artigonal.com/telefoniam-e-celular-artigos/saiba-tudo-sobre-os-smartphones-4601618.html>>. Acesso em: 14 de Abr. 2014.

CÂMARA, Gilberto; DAVIS, Clodoveu; MONTEIRO, Antônio Miguel Vieira. **Introdução à Ciência da Geoinformação**. São José dos Campos: Inpe, 2001 Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/introd/>>. Acesso em: 11 ago. 2014.

CAPANI, Patricia Montovani. **Soluções IBM Rational Mobile: Entrega mais Rápida de Apps com melhor Qualidade**. 2013. Disponível em: <<http://pt.slideshare.net/patriciamentovani714/testes-em-aplicativos-mveis-30216541?related=3>>. Acesso em: 24 out. 2014.

CURITIBA. Prefeitura Municipal. **Modelo colaborativo: experiência e aprendizados do desenvolvimento comunitário em Curitiba / Prefeitura Municipal de Curitiba; GETS - Grupo de Estudos do Terceiro Setor; United Way of Canada - Centraide Canada . - Curitiba: Instituto Municipal de Administração Pública, 2002. 76 p. : il.**

DIAS NETO, Arilo Claudio. **Introdução a Teste de Software**. Revista Engenharia de Software. 2007 disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035#ixzz2AgtFLEZ7>>. Acesso em 24 de out. 2014.

GOOGLE, Code. **Robotium: The world's leading Android™ test automation framework**. 2014. Disponível em: <<https://code.google.com/p/robotium/>>. Acesso em: 24 out. 2014.

GOOGLE PLAY DEVELOPER CONSOLE. 2013. **Google Play** disponível em: <<https://play.google.com/apps/publish/v2/signup/>>. Acesso em: 25 de Fev. 2014.

HELD FILHO, A. **Incorporando a percepção do paciente na avaliação do atendimento odontológico**. 1999. Dissertação (Pós-Graduação Mestrado em Odontologia) – Curso de Pós-Graduação em Odontologia, Universidade Federal de São Paulo, São Paulo.

IDC Brasil. **Worldwide Quarterly Mobile Phone Tracker**. Disponível em: <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp/>>. Acessado em 30 de setembro de 2014

JUNG, Carlos Fernando. **Metodologia para pesquisa & desenvolvimento: aplicada a novas tecnologias, produtos e processos**. Rio de Janeiro: Axcel Books, 2004.

KOTLER, Philip; HAYES, Tom; BLOOM, Paul. **Marketing de serviços profissionais**. 2 ed. São Paulo: Manole, 2002.

LECHETA, R. R. **Google Android** : Aprenda a criar aplicações para dispositivos móveis com Android SDK. São Paulo: Novatec Editora, 2009

MARIMOTO, Carlos E. **Smartphones** - Guia prático. GdH Press. 1.ed. São Paulo: 2009

MELLO, Celso Antonio Bandeira de; **Curso de Direito Administrativo**. Malheiros Editores. 14o Ed. 2001. São Paulo p. 599

OHA. Open Handset Alliance - **Alliance Overview**. Disponível em: <[http://www.openhandsetalliance.com/oha\\_overview.html](http://www.openhandsetalliance.com/oha_overview.html)>. Acesso em: 20 mar. 2014.

PRESSMAN, ROGER S. **Engenharia de software**. 5ª ed. Rio de Janeiro: McGraw-Hill, 2002.

PRIBERAM. **Dicionário Priberam da Língua Portuguesa**. 2014. disponível em: <<http://www.priberam.pt/DLPO/smartphone>>. Acesso em 27 de abr. 2014.

SOMMERVILLE (Org.). **Engenharia de Software**. 8. ed. São Paulo, Sp: Pearson Education Brasil, 2007. Disponível em: <[http://estacio.bv3.digitalpages.com.br/users/publications/9788588639287/pages/\\_1](http://estacio.bv3.digitalpages.com.br/users/publications/9788588639287/pages/_1)>. Acesso em: 26 set. 2014.

VEJA a diferença entre GPS, glonass e A-gps. Disponível em: <<http://www.tudocelular.com/curiosidade/noticias/n35593/diferenca-entre-gps-glonass-agps.html>>. Acesso em: 29 set. 2014.