

O CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RODOLPHO BRUNO DOS SANTOS NOGUEIRA

**DESENVOLVIMENTO DE UM SISTEMA DE APOIO A EVENTOS PARA
UMA PLATAFORMA ANDROID®**

MARÍLIA
2012

RODOLPHO BRUNO DOS SANTOS NOGUEIRA

DESENVOLVIMENTO DE UM SISTEMA DE APOIO A EVENTOS PARA
UMA PLATAFORMA ANDROID®

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Centro Universitário Eurípides de Marília (UNIVEM), mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do Título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fábio Dacêncio Pereira

MARÍLIA
2012

Nogueira, Rodolpho Bruno dos Santos

Desenvolvimento de um sistema de apoio a eventos para uma plataforma Android®/ Rodolpho Bruno dos Santos Nogueira; orientador: Fábio Dacêncio Pereira. Marília, SP: [s.n], 2012.
164 f.

Monografia (Graduação em Ciência da Computação) – Curso de Bacharelado em Ciência da Computação, Fundação de Ensino “Eurípedes Soares da Rocha”, mantenedora do Centro Universitário Eurípedes de Marília – UNIVEM, Marília, 2012.

1. Sistemas Operacionais 2. Aplicações móveis 3. Geolocalização
4. Integração Web 5. Banco de dados.

CDD: 005.43



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Rodolpho Bruno dos Santos Nogueira

DESENVOLVIMENTO DE UM SISTEMA MÓVEL BASEADO EM ANDROID*


Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.


Nota: 7,0 (Sóte)

Orientador: Fábio Dacêncio Pereira

1º. Examinador: Elvis Fusco

2º. Examinador: Ricardo José Sabatine





Ricardo Sabatine

Marília, 10 de dezembro de 2012.

Histórias nossas histórias, dias de luta, dias de glória!”

Charlie Brown Jr

AGRADECIMENTOS

Primeiramente a Deus, por ter me dado a oportunidade de conseguir mais uma grande vitória na minha vida.

Aos meus pais que sempre fizeram de tudo por mim na medida do possível, e que hoje não estão mais aqui fisicamente, mas acredito que estão em algum lugar felizes, como eu estou neste momento, com essa glória alcançada e que eles sempre almejaram pra mim.

Agradeço a minha toda a família que me dão apoio em todos os momentos.

Agradeço ao Professor Dr. Fábio Dacêncio Pereira por ter aceitado me orientar e por ter ajudado muito em meu trabalho. Desejo muito sucesso.

Ao meu amigo Arnaldo pela ajuda e companheirismo de sempre.

Agradeço aos meus amigos que são poucos, mas são os melhores.

NOGUEIRA, Rodolpho Bruno dos Santos, **Desenvolvimento de um aplicativo de apoio a eventos para uma plataforma Android®**, 2012. 164f. Monografia (Bacharelado em Ciência da Computação) – Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, Marília, 2012.

RESUMO

Atualmente, temos vivenciado o avanço exponencial da tecnologia embarcada em dispositivos portáteis, principalmente em smartphones e tablets. Conseqüentemente, há uma crescente demanda por aplicativos que integrem os numerosos recursos disponíveis nestes aparelhos eletrônicos, como GPS e acesso à Internet irrestrito. Assim, um sistema operacional moderno e uma plataforma desenvolvimento são imprescindíveis para o desenvolvimento de aplicativos desta natureza. A empresa Google vem se destacando neste segmento, com o sistema operacional a plataforma de desenvolvimento *Android®*, que oferece aos desenvolvedores uma ferramenta robusta com amplo suporte a diversas tecnologias. Desta forma, este trabalho propõe uma aplicação *Android®* que integre o serviço de geolocalização (GPS), a tecnologia de desenvolvimento *Web* e o suporte à dados (PHP e Mysql). O software desenvolvido contempla as exigências para disponibilização no serviço de downloads *Google Play*.

Palavras chaves: Sistemas Operacionais, Aplicações móveis, Geolocalização, Integração Web, Banco de dados.

NOGUEIRA, Rodolpho Bruno dos Santos, **Desenvolvimento de um aplicativo de apoio a eventos para uma plataforma Android®**, 2012. 164f. Monografia (Bacharelado em Ciência da Computação) – Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, Marília, 2012.

ABSTRACT

Currently, we have experienced exponential advancement of technology embedded in portable devices, especially smartphones and tablets. Consequently, there is a growing demand for applications that integrate the many resources available in these gadgets, such as GPS and unrestricted Internet access. Consequently, there is a growing demand for applications that integrate the many resources available in these gadgets, such as GPS and unrestricted Internet access. Thus, a modern operating system and platform development are essential for the development of such applications. The company Google has been highlighted in this segment, with operating system development platform Android ®, which offers developers a robust tool with extensive support for various technologies. Thus, this paper proposes an Android application that integrates geolocation service, technology development and support for Web data (PHP and MySQL). The software developed includes requirements for providing the service downloads Google Play.

Keywords: Operating Systems, Mobile Applications, Geolocation, Web Integration, Database.

LISTA DE ILUSTRAÇÕES

Figura 1 – Tela inicial do iOS.	24
Figura 2– Aparelhos Iphone, Ipad e Ipod da Apple com o iOS.	25
Figura 3 – Symbian Belle	27
Figura 4 – Camadas do Kernel do Symbian	27
Figura 5 – Sistema operacional Android® versão Jelly Bean	30
Figura 6 – Tablet Galaxy Tab 10.1 com o Sistema Operacional Android® versão 3.0.	31
Figura 7 – Aparelho BlackBerry 9790	32
Figura 8 – Sistema operacional BlackBerry 7	33
Figura 9 – Sistema Operacional Windows Mobile	34
Figura 10 – Windows Phone 8	35
Figura 11 – Tablet Surface com Windows 8	36
Figura 12 – Palm Pre com webOS.....	37
Figura 13 – Sistema Operacional MeeGoo v1.1.....	39
Figura 14 – Smartphone htc com o sistema operacional Brew MP.	40
Figura 15 – Sistema Operacional bada.	41
Figura 16 – Logotipo do Android® versão Cupcake.....	43
Figura 17 – Logotipo do Android® versão Donut.....	44
Figura 18 – Logotipo do Android® versão Eclair.	45
Figura 19 – Logotipo Android® versão Froyo.	46
Figura 20 – Logotipo do Android® versão Gingerbread.....	47
Figura 21 – Logotipo do Android® versão Honeycomb.	49
Figura 22 - Logotipo do Android® versão Ice cream Sandwich.	51
Figura 23 – Logotipo do Android® versão Jelly Bean.	53
Figura 24 – Logotipo do Android®.	56
Figura 25 - Arquitetura dos componentes do sistema Android®.....	59
Figura 26 - Implementação do Main.xml.	69
Figura 27 - Implementação da tela ListaCadastrados.xml.....	70
Figura 28 - Implementação da tela cadastro.xml.	70
Figura 29 - Escolhida a opção Cadastrar Pessoa.	72

Figura 30 - Cadastro e Confirmação de cadastro no Layout Vertical.	72
Figura 31 – Cadastrando o boleiro Messi no layout horizontal.	73
Figura 32 - Confirmação no Layout Horizontal.	73
Figura 33 - Escolhida a Opção Listar Pessoas.	74
Figura 34 - Visualização da Primeira pessoa cadastrada, Rodolpho.	74
Figura 35 - Visualização da segunda pessoa cadastrada, Messi.	75
Figura 36 - IDE do Eclipse adaptado pela Motorola®, cujo Motodev.	76
Figura 37 - APP Validator.	76
Figura 38 - Code Snippets.	77
Figura 39 - Code Generation Wizards.	77
Figura 40 - Database Management.	77
Figura 41 – A aplicação está destacando os Pontos Geográficos instanciados via código.	80
Figura 42 – Aproximação do ponto do Brasil – Fortaleza.	81
Figura 43 – Mensagem ao “clicar” no marcador.	81
Figura 44 – Aproximação do ponto Yokohama – Japão.	82
Figura 45 – Mensagem ao clicar no marcador.	82
Figura 46 – Aproximação do ponto Ciudad de México.	83
Figura 47 – Mensagem do Marcador.	83
Figura 48 – Localização Inicial – Trevo de Pongaí– SP, Latitude: -21.720833	84
Figura 49 – Leitura de GPS e os pontos que a pessoa esteve localizada recentemente.	85
Figura 50 – Notificação da Coordenada atual.	85
Figura 51 – Logotipo do Eloc.	87
Figura 52 – Ferramenta para a criação do repositório de chaves.	91
Figura 53 – Caminho do aplicativo Keytool.	92
Figura 54 – Gerando a impressão digital MD5 da chave privada.	93
Figura 55 – Gerando o layout xml para a utilização dos recursos de mapeamento.	94
Figura 56 - Menu principal do site.	96
Figura 57 – Fluxo de funcionamento do Codeigniter.	97
Figura 58 – Editor de Texto para a internet CKEditor.	99
Figura 59 – Script HTML gerado dos textos criados pelo usuário.	100
Figura 60 – Pagina php gerada com os dados inseridos no CKeditor buscadas no MYSQL.	101
Figura 61 – Tela de locais úteis é integrado com a aplicação web.	102

Figura 62 – Telas do Sistema Android® das abas Evento, Agenda do Evento e Organizadores.	103
Figura 63 –Tela inicial do sistema Eventbrite.	104
Figura 64 – Busca de eventos.	105
Figura 65 – Na busca efetuada foram listados 17 eventos.	105
Figura 66 – Imagens da descrição do evento e da cobrança da entrada.	106
Figura 67 – WhatsOn - Tela Inicial de pesquisa de localidade.	107
Figura 68 – WhatsOn Abas sobre o Evento.	108
Figura 69 – Logotipo do Evento FLISOL	110
Figura 70 – Acessar o Sistema online para a inserção das informações.	117
Figura 71 – Menu principal do Sistema Web.	118
Figura 72 - Inserindo os dados do evento (esquerda), ao clicar em visualizar pagina será possível ver o resultado (direita).	119
Figura 73 - Inserindo a agenda do evento (esquerda), ao clicar em visualizar pagina será possível ver o resultado (direita).	120
Figura 74 - Inserindo a agenda do evento (esquerda), ao clicar em visualizar pagina será possível ver o resultado (direita).	121
Figura 75 – Tipo de localizações disponíveis no sistema.	122
Figura 76 – Cadastro de Localizações.	123
Figura 77 - Criando uma nova localização.	124
Figura 78 – Grid de localizações com a localização criada.	125
Figura 79 – Grid de Geolocalizações da localização Univem.	126
Figura 80 – Criação da Geolocalização inserindo a latitude e a longitude.	127
Figura 81 – Site Find Latitude and Longitude que auxiliara na busca da geolocalização desejada.	128
Figura 82 – Grid das geolocalizações cadastradas.	129
Figura 83 – Sistema Web com todas as Localizações cadastradas e as geolocalizações inseridas.	130
Figura 84 – Grid do Cadastro de usuários.	131
Figura 85 – Inserindo o Usuário e a Senha.	132
Figura 86 – Grid de usuários com o usuário recém-cadastrado.	132
Figura 87 - Acessando o sistema com o usuário criado.	133

Figura 88 – Menu Principal do Sistema Online.....	134
Figura 89 – Carregando as Informações do Sistema Online.	135
Figura 90 – Aba Evento do Eloc recebendo a página dos dados do evento Flisol cadastrado no sistema online.	135
Figura 91 – Aba Agenda do Evento recebendo a página da agenda do evento Flisol cadastrado no sistema.	136
Figura 92 – Aba Organizadores recebendo os dados dos organizadores do evento Flisol cadastrado no sistema online	137
Figura 93 – Menu de Locais Uteis.....	138
Figura 94 – Geolocalização com o marcador mostrando Local do Evento e o endereço.....	139
Figura 95 – Geolocalização com os marcadores dos hotéis cadastrados. No para exemplo foi selecionado o hotel 1.	140
Figura 96 – Geolocalização com os marcadores dos restaurantes cadastrados. No para exemplo foi selecionada o restaurante 1.	141
Figura 97 - Geolocalização com os marcadores dos bancos cadastrados. No para exemplo foi selecionado o banco 1.	141
Figura 98 - Geolocalização com os marcadores dos hospitais cadastrados. No para exemplo foi selecionado o hospital.	142
Figura 99 - Geolocalização com o marcadores do terminal urbano cadastrado.	142
Figura 100 - Geolocalização com o marcador do terminal rodoviário cadastrado.	143
Figura 101 - Geolocalização com o marcador do aeroporto cadastrado.	143
Figura 102 – Mostra o funcionamento do GPS mostrando inclusive o endereço onde se está no momento, sem perder o marcador da busca efetuada.	144
Figura 103 – Verificações de recursos e dados disponíveis.	145
Figura 104 – Pagina do aplicativo desenvolvido o ELoc com opção para download.	146
Figura 105 – Gerando o .apk, selecionando o projeto e a chave.	147
Figura 106 – Ao gerar o apk. foi solicitada a senha da chave.	148
Figura 107– Ambiente para envio e atualização de Aplicativos Android®.....	148
Figura 108 – Repositório de versões lançadas.....	149
Figura 109 – Enviando uma nova versão.	150
Figura 110 - Ao homologar o arquivo enviado o Google Play, será pedida confirmação de envio da versão 1.4.	151

Figura 111– Informando ao usuário as alterações recente.s	152
Figura 112 – Nova versão em vigor no Google Play.	153
Figura 113 – Buscando o Aplicativo no Play Store e em seguida instalando o Eloc.	154
Figura 114 – Aplicativo ELoc baixado do Play Store e funcionando.	155
Figura 115 – Configurando a sincronização do conta Google.	155
Figura 116 – Pesquisando pelo Aplicativo no Google play.	156
Figura 117 - Eloc Listado no Google Play.	157
Figura 118 – Instalando o Aplicativo diretamente do site.	158
Figura 119 – Confirmação e download do ELoc.	159
Figura 120 – Baixando e Instalando o ELoc.	159
Figura 121 – Depois de baixado e instalado o ELoc está funcionando no Android.	160

LISTA DE TABELAS

Tabela 1 – Tabela de distribuição de versões.....	54
Tabela 2 – Tipologia de Eventos.....	86

LISTA DE GRÁFICOS

Gráfico 1 – Ilustra a distribuição das versões do Android®.	55
--	----

LISTA DE DIAGRAMAS

Diagrama 1 – Fluxo de Telas da Aplicação do Cadastro de Pessoas.	71
Diagrama 2 – Diagrama de Classe da classe MainActivity.	88
Diagrama 3 – Diagrama de Classes da classe MyWebViewClient.	88
Diagrama 4 – Diagrama de Classe da classe Mapa.	89
Diagrama 5 – Diagrama de Classe da classe HelloItemizedOverlay.	90
Diagrama 6 – Diagrama de Classes do sistema.	90

SUMÁRIO

Introdução	20
Objetivos	20
Objetivos Específicos	21
Organização do Trabalho	21
1. SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS	23
1.1. Introdução	23
1.2. iOS (Mac OSX) – Sistema Operacional utilizado pelo <i>iPhone, Ipad e Ipod</i>	24
1.3. Symbian	26
1.4. Android®	30
1.5. BlackBerry OS - RIM	31
1.6. Windows Phone	34
1.7. <i>Palm webOS</i>	36
1.8. MeeGo	38
1.9. Brew MP	40
1.10. bada.....	41
2. ANDROID®	42
2.1. Histórico	42
2.2. Versões Estáveis Lançadas	42
2.2.1. Plataforma Android® Versão 1.5 - <i>Cupcake</i>	42
2.2.2. Plataforma Android® Versão 1.6 – <i>Donut</i>	44
2.2.3. Plataforma Android® Versão 2.1 – <i>Eclair</i>	45
2.2.4. Plataforma Android® Versão 2.2 – <i>Froyo</i>	46
2.2.5. Plataforma Android® Versão 2.3 – <i>Gingerbread</i>	47
2.2.6. Plataforma Android® Versão 3.0 – <i>Honeycomb</i>	48
2.2.7. Plataforma Android® Versão 4.0 – Ice Cream Sandwich	51
2.2.8. Plataforma Android® Versão 4.1 – Jelly Bean	53
2.2.9. Distribuição de versões	54
2.3. Android®: Conceitos Fundamentais	55
2.3.1. Características	57

2.4.	Arquitetura dos componentes do sistema <i>Android</i> ®	59
2.4.1.	<i>Linux Kernel</i>	60
2.4.2.	<i>Libraries</i> (Bibliotecas)	60
2.4.3.	<i>Android</i> ® <i>Runtime</i>	61
2.4.4.	<i>Applications</i> (Aplicativos)	62
2.5.	<i>Google Play</i>	64
2.5.1.	Histórico	64
2.5.2.	Como Publicar no <i>Google Play</i>	64
2.6.	Segurança em <i>Android</i> ®	66
2.6.1.	Introdução	66
2.6.2.	Recursos de Segurança do <i>Android</i> ®	66
3.	DESENVOLVIMENTO NA PLATAFORMA <i>ANDROID</i> ®	69
3.1.	Ferramentas para o Desenvolvimento	69
3.1.1.	IDE <i>ECLIPSE</i>	69
3.1.2.	<i>Motodev</i>	76
4.	PROCESSAMENTO DE MAPAS EM DISPOSITIVOS MÓVEIS	78
4.1.	<i>Google Maps API</i>	78
4.1.1.	Obtendo uma <i>MAP API KEY</i>	78
4.1.2.	Exemplo 1 - Fixar Marcadores em diversos pontos do Mapa	80
4.1.3.	Exemplo 2 – Localização Atual colhida via GPS do dispositivo	84
5.	SISTEMA DE APOIO A EVENTOS	86
5.1.	Introdução	86
5.2.	Requisitos	87
5.3.	Diagrama de Classes	88
5.3.1.	Classe <i>MainActivity.java</i>	88
5.3.2.	Classe <i>MyWebViewClient.java</i>	88
5.3.3.	Classe <i>Mapa.java</i>	89
5.3.4.	Classe <i>HelloItemizedOverlay</i>	90
5.3.5.	Diagrama de classes relacionadas	90
5.4.	Procedimento para uso do <i>Google Maps API</i>	91
5.5.	Aplicação Web	95
5.5.1.	Introdução	95

5.5.2. Tecnologias Utilizadas	96
5.6. Integração Aplicação Android® e Aplicação Web	102
5.6.1. Integração pelo banco de dados online - Mysql	102
5.6.2. Integração entre as páginas php geradas pelo Sistema Web - php	102
5.7. Comparação de aplicativos da mesma categoria no Google Play	104
5.7.1. Eventbrite	104
5.7.2. WhatsOn	107
5.7.3. Comparações	109
6. RESULTADOS ALCANÇADOS	110
6.1. Cenário de Testes	110
6.1.1. Sobre o Evento	110
6.1.2. Programação Local do Evento	111
6.1.3. Organizadores do Evento	112
6.1.4. Parceiros “Fictícios” do Evento e locais úteis	112
6.2. Usabilidade	117
6.2.1. Sistema Web	117
6.2.2. Cadastros do Sistema Web	119
6.2.3. Sistema Android® - Eloc	135
6.3. Publicação no Google Play	146
6.3.1. Publicando o aplicativo (.apk)	147
6.3.2. Instalando o aplicativo	154
7. CONCLUSÕES	161
8. REFERÊNCIAS BIBLIOGRÁFICAS	162

Introdução

Atualmente, a telefonia móvel representa um mercado em franco crescimento, ultrapassando 250,8 milhões de linhas ativas em todo Brasil. Sua utilização varia desde a utilização básica do telefone, como chamadas e mensagens de texto, até mesmo a instrumentação de empresas por meio do dispositivo.

Com a crescente evolução da tecnologia os aparelhos celulares estão passando por uma grande revolução, Tais dispositivos passaram a executar funções multimídia, como a reprodução de áudio e vídeo, além de possuírem outros recursos, como câmeras, sensores e GPS. Portanto, tornaram-se verdadeiros computadores pessoais portáteis, com processadores de alto desempenho e memórias com grande capacidade de armazenamento. Assim, surgiu a necessidade do desenvolvimento de sistemas operacionais tão robustos quanto os encontrados em computadores desktop.

Por assumir um expressivo mercado em expansão, grandes empresas estão investindo cada vez mais na área da mobilidade, o Google desenvolveu uma plataforma baseada em Linux chamada *Android*®, segundo o *Google* com o objetivo de ser uma plataforma expansível, flexível e aberta, sendo os aplicativos para *Android*® escritos na linguagem *Java*, portanto essa plataforma baseia-se em uma robusta solução de software livre.

Objetivos

Este estudo tem como objetivo abordar a construção de um aplicativo para dispositivos móveis utilizando a plataforma *Android*® que utiliza *JAVA* e *XML*, utilizando o framework *MOTODEV* da *Motorola*® baseado do *IDE Eclipse*, criando rotinas parametrizadas promovendo a integração com o serviço de geoprocessamento do *Google* o *Google Maps API* como a utilização dos recursos como mapas, localização e navegação por GPS e obtenção de endereços. Posteriormente esse software será publicado no *Google Play* antigo *Android*® *Market*, obedecendo aos padrões e requisitos mínimos para a publicação. Trata-se de uma pesquisa exploratória e descritiva, com revisão bibliográfica para o desenvolvimento de um aplicativo de suporte a eventos que proverá informações essenciais aos participantes sobre o evento, agenda do evento e os dados sobre os organizadores. O sistema trará mapas com os locais como, o local do evento, serviços e utilidades como

hotelaria, restaurantes, bancos e hospitais e a rede de transporte com informações sobre a localização dos terminais rodoviários e aeroportos no qual será desenvolvida uma aplicação web que dará o suporte a aplicação *Android*®, cadastrando e disponibilizando essas informações para o sistema *Android*®. E trará também a localização atual do usuário por GPS.

Objetivos Específicos

- Verificar os *frameworks* disponíveis para o desenvolvimento do projeto.
- Encontrar alguns estudos de caso para desenvolvimento para consequentemente obter familiarização com a linguagem de programação.
- Desenvolver um estudo de caso onde é promovida a integração do *Android*® com os serviços *Web* como *Google Maps Api*.
- Desenvolver o aplicativo de apoio a eventos.
- Publicar o aplicativo desenvolvido no Play e documentar o processo.

Organização do Trabalho

No capítulo 1 serão abordados os sistemas operacionais para dispositivos móveis disponíveis no mercado.

O capítulo 2 abordará o sistema operacional *Android*®, plataforma na qual será desenvolvido o aplicativo, serão descritas as características, conjunto de recursos, funcionalidades do sistema operacional e segurança do sistema operacional.

O capítulo 3 abordará o ambiente de desenvolvimento para o *Android*® onde foram criados dois aplicativos exemplo, base em estudos de casos definidos.

Um recurso de mapas que será importante e integrado no projeto será o *Google Maps Api* onde será abordado no capítulo 4.

O *Android*® projeto para o A será abordado no capítulo 5 que será o sistema de apoio a eventos onde haverá a integração do sistema desenvolvido em *Android*® com serviços *Web* e um site que também foi desenvolvido para integrar os dados do evento e as coordenadas geográficas com o aplicativo *Android*®.

No capítulo 6 serão discutidos os resultados alcançados com o cenário de testes,

usabilidade e conseqüentemente a publicação do aplicativo no *Google Play*.

No capítulo 7 será descrita a conclusão dos estudos realizados.

1. SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS

1.1. Introdução

“[...] Quando se desenvolve uma aplicação móvel, é importante considerar o sistema operacional do dispositivo móvel. O sistema operacional afeta a linguagem, ferramentas e tecnologias que você utiliza para desenvolver a sua aplicação móvel, bem como sua capacidade de dar suporte a manutenção da aplicação [...]”, (VALENTINO; SCHNEIDER; ROBBIE; 2005; p.53).

Dada modernização dos dispositivos móveis com o avanço da tecnologia, os sistemas operacionais devem à altura dos dispositivos que os recebem, pois quanto mais robusto e poderoso for o dispositivo móvel, maior será a expectativa sobre o sistema operacional do aparelho para que consiga suprir totalmente as necessidades, seja do usuário leigo até o mais experiente.

Assim a demanda por sistemas operacionais que aliem qualidade, intuitividade e desempenho será cada vez maior, e com esses sistemas operacionais esses dispositivos serão capazes de desempenhar todo o tipo de atividade ou tarefas a que lhe for atribuída de maneira satisfatória, agradando a todos os tipos de usuários.

Abaixo uma lista de alguns sistemas operacionais disponíveis no mercado:

- iOS (Mac OSX) – *iPhone* da *Apple*®
- *Android*®
- *Symbian*®
- *BlackBerry*® - RIM
- *Windows Mobile*
- *Palm webOS*®
- *MeeGo*®
- *Brew MP*®
- *Bada*®

Esses sistemas operacionais se dividem entre dois tipos:

Sistemas Operacionais Proprietários: São aqueles que em o sistema operacional desenvolvido é patenteado, podendo ser utilizado apenas por dispositivos da empresa fabricante ou por quem quiser pagar pela sua utilização, geralmente não é disponibilizado o código fonte.

Sistemas Operacionais de Código Aberto: São os sistemas operacionais em que todos poder tem acesso ao código fonte e é livremente distribuído para alterações, melhorias ou elaboração de outro novo com base no código fonte.

No caso do Windows Phone o código fonte do sistema não é aberto.

1.2. iOS (Mac OSX) – Sistema Operacional utilizado pelo *iPhone, Ipad e Ipod*

O iOS é uma versão otimizada para dispositivos móveis do sistema operacional do sistema operacional *Mac OS* denominada Mac OSX, que foi fundamentado na sólida base *UNIX*, sendo o SO carregado de maneira criptografada para a memória RAM, somente aplicações assinadas podem acessar ao *Kernel*. É baseado na arquitetura é ARM.



Figura 1 – Tela inicial do iOS.

Fonte: (APPLE, 2012).

O iOS atualmente na versão 6, é a base do *iPhone*, do *iPad* e do *iPod touch*. Como a Apple faz o hardware e o sistema operacional do *iPad*, *iPhone* e *iPod touch*, tudo é projetado para funcionar em conjunto. Assim, os apps aproveitam ao máximo todos os recursos de hardware como a tela Retina, a interface Multi-Touch, o acelerômetro, o giroscópio com três eixos, processadores gráficos acelerados e muito mais. O FaceTime é um exemplo perfeito disso. Ele usa a câmera frontal e a traseira, a tela, o microfone e a sua conexão *Wi-Fi*. As atualizações do iOS são gratuitas. Elas podem ser baixadas no *iPhone*, *iPad* ou *iPod touch* assim que são lançadas, (APPLE, 2012).

Tratando de segurança os recursos de *hardware* e o *firmware* de baixo nível protege contra *malware* e vírus; os recursos de alto nível do OS ajudam a proteger o acesso a informações pessoais e dados corporativos. Para proteger a privacidade, os aplicativos que solicitam informações sobre a localização e dados do Calendário, Contatos, Lembretes e Fotos precisam primeiro obter autorização. Pode definir um bloqueio por senha para evitar o acesso não autorizado ao dispositivo e configurá-lo para que todos os dados contidos nele sejam excluídos após um determinado número de tentativas de entrada de senha equivocada. Essa senha codifica e protege automaticamente os e-mails armazenados e também permite que apps de terceiros codifiquem seus dados armazenados. O iOS aceita comunicação de rede codificada para a proteção de informações durante a transmissão das mesmas. Caso o dispositivo seja perdido ou roubado, o aplicativo Buscar *iPhone* pode localizá-lo em um mapa e apagar todos os seus dados remotamente. Quando recuperar o dispositivo, é possível recuperar todo o conteúdo a partir do backup mais recente, (APPLE, 2012).



Figura 2 – Aparelhos iPhone, iPad e iPod da Apple com o iOS.

Fonte: (APPLE, 2012).

As ferramentas necessárias para o desenvolvimento, teste, execução, depuração e customização de aplicativos pra o iOS, estão no *Xcode*. Fazem parte do *Xcode* as seguintes ferramentas: o SDK, *Xcode* IDE, *Interface Builder*, *Instruments*, *iPhone Simulator*. O *Xcode*, é o ambiente utilizado para desenvolvimento, em termos de aplicabilidade pode ser comparado ao Eclipse ou Visual Studio. O *Interface Builder* é utilizado para criação de telas e componentes. O *Instruments* é a ferramenta de análise de consumo de recursos computacionais como memória, e processamento. O *iPhone simulator*, é utilizado para simular como o aplicativo será executado no iPhone ou iPad, mas isso não descarta a necessidade de testar o aplicativo no próprio dispositivo que é o ambiente real onde o aplicativo será executado, (OLIVEIRA, 2011).

1.3. Symbian

O sistema Symbian, foi um projeto concebido pela parceria de um grupo de fabricantes: Nokia®, Ericsson®, Motorola® e Panasonic® como maior objetivo a rapidez e acessibilidade, época em que o sistema era de código proprietário das empresas desse consorcio. Hoje o Symbian inteiramente à Nokia que distribui o Symbian para empresas licenciadas, é um sistema aberto e de baixo custo, possui o gerenciamento de recursos em casos pouca bateria e suporta a aplicações desenvolvidas em C++.

O Symbian tem um design modular. Assim, por exemplo, os serviços de disco principal são executados pelo serviço de arquivos, ao mesmo tempo em que os serviços de tela e de entrada do usuário são executados pelo serviço de janela. No entanto, há um elemento que você pode pensar em como o coração do sistema operacional o elemento que é responsável pelo gerenciamento de memória, gerenciamento de tarefas e agendamento de tarefas. Esse elemento é claro, o *kernel*, EKA2(SALES, 2006, p.3). Atualmente o sistema está na versão Belle.



Figura 3 – Symbian Belle.

Fonte: (Próprio Autor, 2012).

O *kernel* é construído em camadas. As camadas de uma estrutura de núcleo refletem a funcionalidade de uma parte do núcleo. Aplicar das camadas internas de base, funções primitivas, de tal forma a executar rapidamente. Camadas mais internas são também as camadas mais privilegiadas, capaz de acessar todos os componentes do sistema operacional sempre que necessário, (JIPPING, 2007, p.48). O *Kernel* atual é o EKA2 é uma evolução dos antigos EPOC e EKA1.

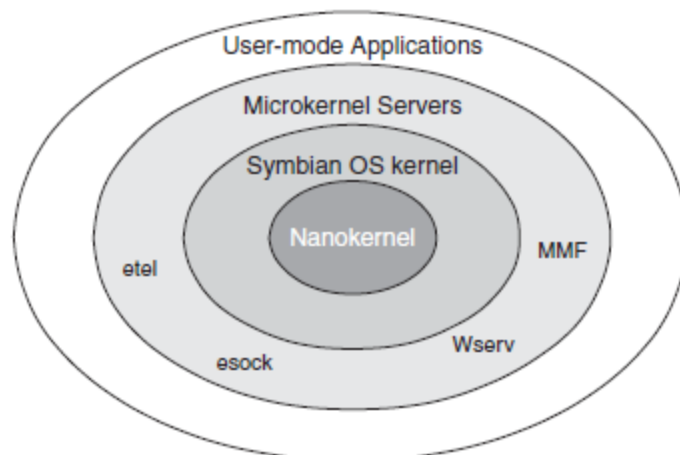


Figura 4 – Camadas do Kernel do Symbian.

Fonte: (SALES, 2006).

Uma breve descrição das camadas:

NANOKERNEL

O *nanokernel* fornece algumas das funções mais básicas no Symbian, operam serviços de modo privilegiado e de forma primitiva. Incluídos entre as implementações deste nível são de programação, operações de sincronização, manipulação de interrupção e objetos de sincronização chamados *mutexes* e semáforos. A maioria das funções implementadas neste nível pode ser antecipada. Funções a este nível são tão primitivas que o *nanokernel* não deve operar qualquer tipo de operação complicada, tais como alocação de memória dinâmica, (JIPPING, 2007, p.49).

SYMBIAN OS KERNEL

O Symbian OS é a camada do *kernel* que fornece funções necessárias para o resto do sistema operacional. Cada operação a este nível é um privilégio da operação e combina as operações primitivas do *nanokernel* para implementar tarefas mais complexas. Complexos serviços de objetos de modo de usuário, agendamento de processos, e troca de contexto, memória dinâmica, bibliotecas carregadas dinamicamente e comunicação entre objetos de sincronização complexos são apenas algumas das operações executadas por essa camada, (JIPPING, 2007,p.49).

MICROKERNEL SERVERS

Esta camada de é típica de arquiteturas *microkernel*, Baseia-se das funções, tipicamente para governar funcionalidades específicas, tal como manusear a tela ou a trabalhar com sockets, é geralmente executado como modo de usuário de serviços. Estas áreas de funcionalidade exigem operações de *kernel* apenas esporadicamente. (JIPPING, 2007, p.49).

USER MODE

Camada onde as aplicações são executadas quase completamente, fazendo interação com outras camadas ou faz chamadas de sistema, (JIPPING,2007,p.49).

Hoje a devido à grande crise financeira mundial e ressecção econômica da Europa (a Nokia® é uma empresa norueguesa) a *Nokia*® teve grandes prejuízos e conseqüentemente quedas nas vendas e nisso a empresa presou por inovações e está com os esforços voltados em ajudar a Microsoft em desenvolver e homologar o projeto do Windows 8 para dispositivos móveis e conseqüentemente utiliza-lo em seus aparelhos, assim o Symbian deixou de ter prioridade para a Nokia®, mas o projeto Symbian não foi abandonado, mas está afetando no lançamento atualizações e novas versões acarretando numa possível defasagem de tecnologia comparando a outras plataformas móveis.

1.4. Android®

A plataforma *Android*® é a plataforma mais popular atualmente, foi inicialmente desenvolvida pelo Google e posteriormente pelo *Open Handset Alliance* que é um consórcio de mais de 80 empresas segundo a empresa, o *Google* gerencia o projeto que foi construído partir do zero para permitir que desenvolvedores criem aplicativos atraentes, e aproveitar ao máximo tudo o que os dispositivos móveis têm para oferecer. A plataforma pode ser usada *Smartphones, tablets*.



Figura 5 – Sistema operacional Android® versão Jelly Bean.

Fonte: (ANDROID, 2012).

São fornecidas gratuitamente ferramentas necessárias para começar o desenvolvimento de aplicativos na plataforma *Android*® usando a linguagem de programação Java.



Figura 6 – Tablet Galaxy Tab 10.1 com o Sistema Operacional Android® versão 3.0.

Fonte: (ANDROID, 2012).

O Android® foi criado a partir da abertura do *Kernel* do Linux, e utiliza uma máquina virtual personalizada a *Dalvik VM* que foi criada para aperfeiçoar os recursos de memória e hardware em um ambiente móvel. Posteriormente serão abordados mais detalhes sobre a plataforma.

1.5. BlackBerry OS - RIM

O termo *Smartphones* sempre foi um termo associado aos aparelhos da empresa canadense RIM - *Research in Motion* onde se integrou pela primeira vez em um aparelho celular serviços como acesso à internet, e-mail, tecnologia Ipv6 e editor de texto. Os aparelhos da RIM são caracterizados possuem um botão que funciona como um mouse de computador e um mini teclado alfanumérico que permite a introdução de textos.



Figura 7 – Aparelho BlackBerry 9790.

Fonte: (BLACKBERRY, 2012).

Seu sistema operacional o *BlackBerry OS* é um sistema operacional proprietário desenvolvido apenas para o *BlackBerry*. Caracterizado por ser de uso corporativo e profissional com um motor buscas fácil e possui serviço próprio e exclusivo de e-mail disponibilizado pela RIM, chamado BBM (*Blackberrymessenger*) que para envio e recepção de e-mails em velocidade chega até 200kbps, utilizando a tecnologia EDGE.

Desenvolvedores independentes criaram uma grande variedade de outros programas para o *BlackBerry*, como jogos e aplicativos de produtividade. Na página do fabricante é disponibilizado o ambiente para o desenvolvimento de aplicativos.



Figura 8 – Sistema operacional BlackBerry 7

Fonte: (BLACKBERRY, 2012).

1.6. Windows Phone

Windows Phone é uma plataforma para dispositivos móveis, baseada no *Windows CE 5.0* e usado em uma ampla variedade de hardware de terceiros, tais como assistentes digitais (PDAs) e smartphones. O *Microsoft Visual Studio 2008* e o *Windows Mobile SDK (Software Development Kit)*, possibilitam a criação de software para a plataforma *Windows Mobile* em código nativo (*Visual C++*) ou em linguagens com código gerenciado (*Visual C#, Visual Basic.NET*), (MSDN, 2008).

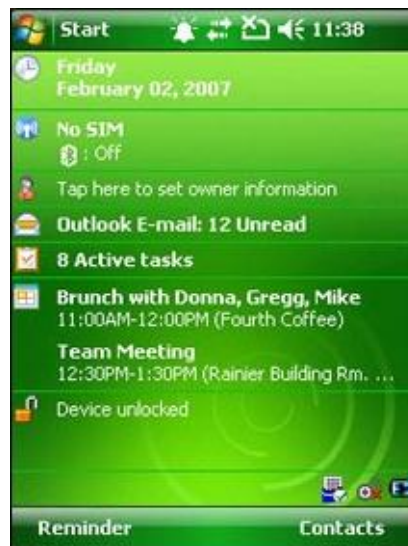


Figura 9 – Sistema Operacional *Windows Mobile*.

Fonte: (MSDN, 2008).

O *Windows Phone* estende a familiaridade do área de trabalho *Windows* para dispositivos *Windows Mobile*. O *Windows Phone* baseia-se em *Windows Embedded CE* e oferece suporte a *.NET Compact Framework*, (MSDN, 2008).

Trabalha para 2 tipos de dispositivos: Dispositivos com tela sensível ao toque os *Pocket PC* e *Smartphone* e dispositivos sem telas sensível ao toque.

Usando a plataforma *Windows Mobile*, é possível utilizar os mais variados aplicativos para dispositivos móveis. A plataforma oferece recursos como conectividade de dados que é uniforme com suporte a *Bluetooth*, internet e recursos *multithreading*. Para rodar o sistema operacional o hardware do aparelho precisa ser robusto, pois o SO exigente bastante do hardware.

Windows Phone 8

Apostando em integração entre smartphone, tablet, vídeo game e computadores em geral de dispositivos a Microsoft lança o *Windows Phone 8*. Desenvolvido em parceria com a Nokia, os recursos do sistema vem organizados em pequenos blocos para deixar as funcionalidades e recursos do sistema com fácil acesso e deixando a experiência do usuário mais simples e agradável e deixando ao mesmo tempo atualizado com as redes sociais.



Figura 10 – Windows Phone 8.

Fonte: (Windows Phone 8, 2012).

Tablet Surface

Com crescimento dos *tablets* a Microsoft lançou o *Surface* para combater para tentar abocanhar o mercado concorrendo com o *Ipad* e o *Samsung Galaxy Tab*. O que merece ser um diferencial desse tablet é que ele vem com teclado portátil magnético. O sistema operacional será o *Windows 8*, que promoverá a integração com dispositivos em que houver o mesmo sistema operacional.



Figura 11 – Tablet Surface com Windows 8.

Fonte: (SURFACE, 2012).

1.7. Palm webOS

A *Palm* foi adquirida recentemente pela HP a *Hewlett & Packard* numa transação de aproximadamente 1,2 bilhões de dólares portando investimento e novas ideias não serão pouco para que a plataforma faça concorrência para os SO do Google e *Apple* e RIM.

No Brasil a plataforma e os aparelhos não sejam tão popular para o usuário convencional, a os aparelhos da Palm são mais requisitados para uso profissional assim como os *BlackBerry*

O *webOS* é o substituto da *Palm* para o envelhecido *Palm OS*, o sistema usado pelos celulares inteligentes Treo e Centro. Desenvolvido para satisfazer as necessidades criativas dos usuários multitarefas modernos, o *webOS* foi concebido para acionar a mais

recente geração de celulares inteligentes da Palm, que inclui os modelos de aparelho Pre e o Pixi, (BRIGGS,2009).



Figura 12 – Palm Pre com webOS.

Fonte: (WEBOS, 2011).

Conta com uma interface multimídia de fácil uso semelhante à oferecida pelo iPhone. O modelo Pre, é o primeiro aparelho a ser lançado com o novo sistema operacional, mostra arranjo semelhante ao do iPhone, com os aplicativos alinhados em fileiras de três e uma barra de acionamento rápido. Em termos gráficos, o webOS é agradável esteticamente. As cores são ricas e os aplicativos parecem nítidos e bem detalhados, (BRIGGS, 2009).

Os engenheiros do webOS criaram um sistema operacional que pode ser usado apenas com tela de toque ou em combinação com um teclado físico. Os programadores do webOS usam a linguagem *Ajax*. Tradicionalmente, os aplicativos para Web têm base HTML e usam o HTTP como principal veículo de comunicação. Assim que um pedido é enviado, o aplicativo espera a resposta antes de se atualizar, o que desacelera o programa. Mas os aplicativos *Ajax* conduzem diretamente as interações dos usuários e transferem informação de forma assíncrona, (BRIGGS,2009).

Como muitos outros dos sistemas operacionais modernos para celulares inteligentes, o *webOS* também oferece ferramentas para que programadores escrevam software. As aplicações podem ser escritas em C++ ou em linguagens *Web* como *Ajax* e *JavaScript*.

1.8. MeeGo

O MeeGo é um projeto aberto e colaborativo entre os fundadores do projeto (Nokia e Intel), e vários parceiros comerciais e não-comerciais, com o objetivo de acelerar a adoção do *Linux* em uma magnitude de dispositivos e o enriquecimento técnico da plataforma *Linux* como a plataforma de escolha para dispositivos de computação móvel, (HADDAD, 2011).

A plataforma é projetada para dar aos desenvolvedores a mais ampla gama de segmentos de dispositivos para direcionar para as suas aplicações, incluindo netbooks, dispositivos portáteis, dispositivos de informação e entretenimento, TVs inteligentes. O MeeGo é um projeto arcado pela *Linux Foudation*, (HADDAD, 2011).

O MeeGo é uma plataforma de código aberto seguindo as melhores práticas de desenvolvimento open source e inclui:

1. Núcleo do sistema operacional.
2. Interface do usuário (UI) bibliotecas e ferramentas.
3. Referência a experiência do usuário para dispositivos múltiplos.
4. Conjunto padrão de interfaces de programação de aplicativos (APIs) em todos os tipos de dispositivos de destino.
5. Um *Software development kit* (SDK) que permite que os desenvolvedores de aplicativos para desenvolver, instalar e depurar e executar aplicativos, seja em dispositivos de referência ou em um ambiente emulado.



Figura 13 – Sistema Operacional MeeGoo v1.1.

Fonte (HADDAD, 2011).

A próxima geração de *smartphones* permitirá aos usuários desfrutar de uma experiência de Internet rica e dinâmica, assistir filmes em HD, e atividades multitarefa como nunca antes em um dispositivo pequeno. A plataforma MeeGo é projetada especificamente para permitir a aplicação uma gama de serviços para esses móveis, (HADDAD, 2011).

O MeeGo v1.1 fornece a tecnologia que oferece ao aparelho celular, conectividade, sensores e navegação móvel, chamado de voz, mensagens SMS, navegador Web, reprodução de música e vídeo, visualização de fotos e gerenciamento de conexão, (HADDAD, 2011).

O MeeGo SDK 1.1 beta para desenvolvimento foi lançado em 10 de novembro de 2010, e está disponível para download juntamente com as instruções de mego.com/desenvolvedores. Ele permite que os desenvolvedores de aplicativos para desenvolver, instalar e depurar aplicativos, bem como executar aplicativos no Nokia N900, Netbook, e aparelhos Java com MeeGo. Para os desenvolvedores de hardware sem referência, QEMU (qemu.org) fornece um ambiente emulado pra depurar e testar aplicações, (HADDAD, 2011).

1.9. Brew MP

O *Brew Mobile Platform* ou Brew MP é um sistema operacional concebido pela *Qualcomm*, fabricante de chips para smartphones como o *Snapdragon*. Trata-se de um *chip* que, funcionando juntamente com o sistema operacional, é capaz de transformar radicalmente qualquer telefone móvel em um dispositivo cheio de funções e suporte a aplicativos, (VALLIN, 2010).



Figura 14 – Smartphone htc com o sistema operacional Brew MP.

Fonte: (VALLIN, 2010).

O *Brew Mobile Platform*, se destaca pela facilidade de acesso à redes sociais e simplicidade dos comandos, para um usuário pouco exigente e também privilegia aparelhos com menos recursos. O código do sistema não é aberto mas o SDK é amplo, para não limitar os desenvolvedores.

1.10. bada

Samsung bada é uma plataforma foi criada para uma vasta gama de equipamentos da *Samsung*, lançado em 2010. Com o *bada* é prezado o custo benefício aos clientes por ser um sistema desenvolvido pela própria empresa. O código fonte do sistema não é aberto. Há bastante semelhança com o *Android*®.



Figura 15 – Sistema Operacional bada.

Fonte: (BADA, 2012).

O bada é a Sistema operacional padrão dos *Smartphones* Samsung que não oferecem outra opção de sistema operacional. Samsung bada tem uma longa história de quase 10 anos. Durante este tempo, tem sido plataforma proprietária da Samsung, sendo implantado em muitos aparelhos de sucesso. Foram adicionados recursos, como multi-touch, gráficos 3D, uma interface de usuário melhorada e downloads de aplicativos.

A Samsung também oferece o SDK para desenvolvimento de aplicativos para o bada, a linguagem utilizada é o Java, o ambiente em que os programadores podem oferecer os programas desenvolvidos é o Samsung Apps onde os usuários podem baixar ou comprar os aplicativos.

2. ANDROID®

2.1. Histórico

A história do Sistema Operacional *Android*® começa em Outubro de 2003, quando os empresários Andy Robin, Rich Miner e Chris White criaram a empresa *Android*® Inc. em Palo Alto na Califórnia com o objetivo de criar sistemas operacionais para dispositivos móveis. Visando se firmar no mercado, em 2005 o Google adquiriu a empresa.

Deste então o *Android*® virou sinônimo de sucesso, tanto que é a plataforma móvel mais popular atualmente com 400 milhões de dispositivos ativos, graças as contribuições da comunidade Linux, parceiros e operadoras, pois além de ser uma plataforma aberta existem muitas empresas que lhe da suporte tecnológico, há consorcio de mais 80 empresas que unidas criaram a *Open Handset Alliance*(OHA) que trabalham no desenvolvimento, aprimoramento de novos recursos e também na distribuição da plataforma, juntos a OHA reúnem ideias para melhorar cada vez mais plataforma. Curiosamente cada versão lançada ganha um nome de uma guloseima.

2.2. Versões Estáveis Lançadas

2.2.1. Plataforma Android® Versão 1.5 - Cupcake

Lançada em Maio de 2009 o versão *Android*® *Cupcake*, incluía novos recursos para os usuários e desenvolvedores. Para os desenvolvedores foi disponibilizado um SDK para o desenvolvimento de aplicações na plataforma. Essa é a Api Nível 3. O nível de api determina se um aplicativo desenvolvido é compatível com a versão do *Android*® no dispositivo antes de efetuar a instalação, (ANDROID, 2012).



Figura 16 – Logotipo do Android® versão Cupcake.

Fonte: (Próprio Autor, 2012).

Destaques da Versão 1.5

- Refinamentos na Interface de usuário
- Melhorias de desempenho
- Novos Recursos: Tela de teclado virtual, *Widgets* da Tela inicial e *Câmera & Gallery*: Gravação de vídeo, Reprodução de vídeo (MPEG-4 e 3GP formatos)
- *Bluetooth*
- Navegador atualizado
- Contatos
- *Novo* kernel do Linux (versão 2.6.27)
- Sistema de Arquivos *SD card* com auto verificação e reparação
- Aplicação *SIM Toolkit* 1.0

2.2.2. Plataforma Android® Versão 1.6 – Donut

Em outubro de 2009 foi lançada uma pequena versão do *Android*® com atualização e inserção de novos recursos de acessibilidade, opções de pesquisa de conteúdo, visualizador do *status* de bateria, notificações do *Google Play*, recursos de conectividade para o sistema operacional. Essa é a Api nível 4, (ANDROID, 2012).



Figura 17 – Logotipo do Android® versão Donut.

Fonte: (Próprio Autor, 2012).

Destaques da Versão 1.6

- Caixa de pesquisa rápida
- Câmera, Filmadora e Galerias de Imagem e Videos
- VPN, 802.1x
- Indicador de uso da bateria
- Acessibilidade
- Atualizações do *Google Play*
- Estrutura de pesquisa expandida
- *Text-to-speech*: Recurso no qual um aplicativo consegue interpretar e falar um conjunto de palavras escritas do teclado ou selecionadas
- Gestos
- Acessibilidade

- Suporte expandido para densidades de tela e resoluções
- Suporte para telefonia CDMA
- Nova versão do *OpenCore*
- 2.6.29 do *kernel Linux*

2.2.3. Plataforma Android® Versão 2.1 – *Eclair*

Em janeiro de 2010 foi lançada a versão 2.1 que traz recursos onde é possível o sincronismo de contatos de e-mails e redes sociais, e os navegadores da plataforma passam a contar com suporte a HTML5. Essa é a API nível 7, (ANDROID, 2012).



Figura 18 – Logotipo do Android® versão Eclair.

Fonte: (Próprio Autor, 2012).

Destaques da Versão 2.1

- Sincronismo de Contatos de emails e redes sociais
- Visualização dinâmica E-mail
- Mensagens MMS e SMS
- Melhorias nos Recursos de Câmeras
- Otimização do Teclado Virtual e do dicionário de palavras

- Atualização do Navegador com suporte a HTML5 e do Calendário
- Otimização das bibliotecas gráficas e do *Bluetooth*.

2.2.4. Plataforma Android® Versão 2.2 – *Froyo*

Em maio de 2010 foi lançada a versão 2.2, nela há destaques como a nova tela inicial que deixa a interação com os atalhos mais dinâmica, tem também a opção de transformar o dispositivo como um mini-*hostpot* onde é possível rotear internet para outros dispositivos com notebook ou outro smartphone que receba internet e a otimização da máquina virtual Dalvik. Essa é a API nível 8, (ANDROID, 2012).



Figura 19 – Logotipo Android® versão Froyo.

Fonte: (Próprio Autor, 2012).

Destaques da Versão 2.2

- Nova tela Inicial
- Suporte ao *Exchange*
- Câmera e Galerias de mídia
- *Hotspot Portátil*
- Vários idiomas no teclado

- Otimização de desempenho: Otimização *Dalvik MV*, desempenho do navegador e do *Kernel*
- *Bluetooth*
- *Kernel* Atualizado

2.2.5. Plataforma Android® Versão 2.3 – *Gingerbread*

Em novembro de 2010 foi lançada a versão 2.3 com suporte a serviços baseados em VoIP, aperfeiçoamento no serviço de sensores e orientação do dispositivo, refinamento no serviço de câmeras e interface ao usuário. Essa é a API nível 9, (ANDROID, 2012).



Figura 20 – Logotipo do Android® versão *Gingerbread*.

Fonte: (Próprio Autor, 2012).

Destaques da Versão 2.3

- Refinamentos de interface do usuário para a simplicidade e velocidade
- Entrada de texto mais rápida e intuitiva
- Melhoramento da função de copiar e colar
- Melhor gerenciamento de energia
- Melhor controle sobre as aplicações.
- Novas formas de comunicação, organização: Chamadas pela *Internet*, *Near Field Communications* (NFC): Componente que trata da comunicação e troca

de informações entre dispositivos diferentes.

- Gestão de Downloads
- Acesso a todas as Câmeras do dispositivo
- Telefonia via Internet
- Aprimoramento dos efeitos de áudio
- Suporte para novos formatos de mídia
- Atualização do *Kernel Linux* para 2.6.35
- Atualização da máquina virtual *Dalvik*

Pouco tempo depois o Google anunciou mais duas versões a 2.3.3 e a 2.3.4 adicionando correções de *bugs* e inserção de *patches*, não houve qualquer mudança na versão 2.3.3 Ambas foram lançadas em fevereiro de 2011. Ambas são API nível 10, (ANDROID, 2012).

Destaques da Versão 2.3.3

- Correções no NFC
- Plataforma de suporte a conexões *Bluetooth*
- Melhora nos Gráficos
- Melhora na Api de reconhecimento de voz

Destaques da Versão 2.2.4

- Inserção Biblioteca Aberta de Acessórios: Promove a integração e gerenciamento de dispositivos conectados via USB.

2.2.6. Plataforma Android® Versão 3.0 – *Honeycomb*

A versão *Honeycomb* é a primeira versão da Plataforma Android® que foi otimizada especialmente para dispositivos que possuem a tela maior, particularmente os *tablets*. Ele apresenta um novo design de interface virtual e elegante modo de interação com o conteúdo. Ele é multitarefa, notificações detalhadas, há opções de personalização da tela inicial e nova *widgets*.



Figura 21 – Logotipo do Android® versão Honeycomb.

Fonte: (Próprio Autor, 2012).

O *Honeycomb* suporta aplicativos desenvolvidos em versões anteriores. Essa é a Api nível 11 lançada em fevereiro de 2011. A versão oferece também SDK para o desenvolvimento de aplicativos para a versão.

Posteriormente em abril de 2011 e junho de 2011 foram lançadas as versões 3.1 API nível 12 e 3.2 API nível 13 respectivamente com novos componentes e ajustes pontuais:

Destaques da versão 3.0

- Barra de título dos aplicativos contem o ícone e menu de opções
- Todos aplicativos contém a opção de copiar e colar
- Opção de transferir dados de um aplicativo para o outro
- Aplicativos mais iterativos na tela inicial de usuários
- Atualização da barra de notificação do *Android®*
- Facilidade do carregamento assíncrono de conteúdos
- Nova estrutura de Animações
- Nova interface gráfica
- Suporte para novos tipos de mídia
- Suporte para Teclado
- Interface Multi-toque

- Novos recursos para navegação na internet
- *Kernel Linux* (Atualizado para 2.3.36)
- Dalvik VM atualizada

Destaques da versão 3.1

- Atualização no suporte a USB
- Suporte para novos dispositivos de entrada e eventos de movimento
- Janelas redimensionáveis
- Atualização no quadro de animações
- Atualização no quadro de interface
- Otimização de Gráficos
- Novas animações
- Otimização nos componentes de ambiente de rede
- Atualização no Gerenciador de *Download*
- Notificação de lançamento primeira aplicação e atualização de aplicativos

Destaques da Versão 3.2

- Otimizações para uma gama mais vasta de dispositivos
- Zoom de compatibilidade para aplicativos de tamanho fixo
- *Media Sync* do cartão SD, sincronismo com o cartão de armazenamento.
- Redimensionamento de telas conforme especificações do dispositivo
- Densidade nova para 720p

2.2.7. Plataforma Android® Versão 4.0 – Ice Cream Sandwich

Na versão 4.0 o sistema operacional é multitarefa, possuem notificações ricas em detalhes, telas iniciais personalizáveis, *widgets* redimensionáveis, e interatividade a adição de novas formas poderosas de comunicação e de compartilhamento em diversas redes sociais. Essa é a Api nível 14.



Figura 22 - Logotipo do Android® versão Ice cream Sandwich.

Fonte: (Próprio Autor, 2012).

Destaques da versão 4.0

- Nova Interface refinada
- Início teclas virtuais de atalho e com botão de favoritos
- Os *widgets* redimensionáveis
- Atualizações nas barras notificações, tarefas, e as abas do navegador
- Acesso rápido para as chamadas recebidas
- Novas ações da tela de bloqueio
- Entrada de texto melhorado com verificação ortográfica
- Mecanismo comando de voz
- Maior controle sobre os dados da rede
- Projetado para acessibilidade

- Recursos para Comunicação e compartilhamento em redes sociais
- Integração de serviços na nuvem oferecidos pelo Google
- Serviços *Wi-Fi Direct e Bluetooth*
- Novas funcionalidades de câmera
- Ferramentas para manipulação de imagens e vídeo
- Novo pacote atualizado de *codecs*
- Melhorias nos layouts das telas
- Uso de rede eficiente
- Segurança para aplicações e conteúdo
- Otimização dos recursos de VPN

O Android® versão 4.0.3 é uma versão incremental da plataforma *Android®* família 4.x (*Ice Cream Sandwich*). Sua API é nível 15

Destaques da Versão 4.0.3

- Otimização do sincronismo entre contatos e redes sociais
- Atualização do provedor de calendário
- Ajustes nos *Widgets* de tela Início
- Otimização da verificação ortográfica
- Atualização nos recursos de *bluetooth*
- Atualização nos recursos gráficos
- Atualização nos recursos de acessibilidade
- Atualização nos recursos de Câmera

2.2.8. Plataforma Android® Versão 4.1 – Jelly Bean

A versão 4.1 é a versão mais recente e oferece melhor desempenho e experiência de usuário aprimorada. Foram adicionados novos recursos para usuários e desenvolvedores. Essa é a Api nível 16.



Figura 23 – Logotipo do Android® versão Jelly Bean.

Fonte: (Próprio Autor, 2012).

Destaques da versão 4.0

- Suporte a vários usuários no sistema operacional
- Serviços isolados por usuário
- Gerenciamento de memória
- Mais fácil encontrar e instalar *Live Wallpapers* – Plano de fundo dinâmico
- Atualização do *Codecs* de mídia
- Tratamento na imagem da câmera com o foco automático
- Serviços de descoberta de rede baseado em DNS com *Wi-Fi Direct*
- Suporte a múltiplos dispositivos de entrada (Teclado, *joystick*)
- Acessibilidade melhorada
- Suporte para usuários internacionais (idioma)
- Redimensionamento de *Widgets* melhorado

- Maior resolução fotos de contato da agenda
- Gerenciamento de Banda Rede
- USB Áudio – Suporte a saída de áudio via USB
- Opção de Gravação de áudio com base em mídia existente
- Suporte a áudio multicanal
- Pré-processamento de áudio captado em caso de gravação

2.2.9. Distribuição de versões

A seguir será mostrado um gráfico e uma tabela mostrando proporção distribuição de cada versão para cada dispositivo que acessou o *Google Play* em um período de 14 dias e foi encerrado em 01 de novembro, seguem os dados abaixo:

Version	Codename	API	Distribution
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.3%
2.1	Eclair	7	3.1%
2.2	Froyo	8	12%
2.3 - 2.3.2	Gingerbread	9	0.3%
2.3.3 - 2.3.7		10	53.9%
3.1	Honeycomb	12	0.4%
3.2		13	1.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.8%
4.1	Jelly Bean	16	2.7%

Tabela 1 – Tabela de distribuição de versões.

Fonte: (ANDROID, 2012).

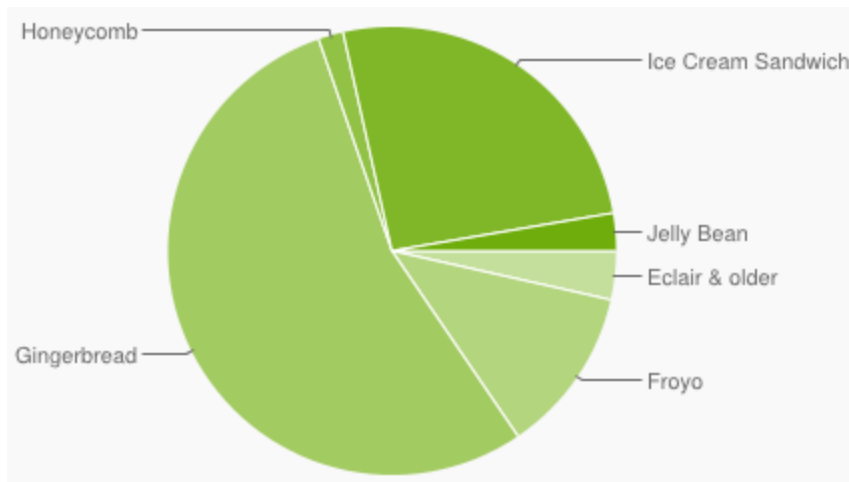


Gráfico 1 – Ilustra a distribuição das versões do Android®.

Fonte: (ANDROID, 2012).

Como pode ser visualizado a versão *Gingerbread* V. 2.3.3 - 2.3.7 é a mais utilizada pelos dispositivos conforme a amostra com 53.9 % seguida a *Ice Cream Sandwich* com 25.8% dos dispositivos. Mas no cenário atual mostra que a versão mais obsoleta domina a quantidade de dispositivos ativos, por terem preços mais acessíveis aos consumidores, portanto as versões mais atuais estão em dispositivos robustos e consequentemente mais caros menos acessíveis em questão de preço para grande maioria dos consumidores.

2.3. Android®: Conceitos Fundamentais

Segundo o Google *Android® Developers*, o *Android®* é um software para dispositivos móveis que inclui um sistema operacional, middleware e aplicativos. O *Android® SDK* fornece as ferramentas e APIs necessárias para começar o desenvolvimento de aplicativos na plataforma *Android®* usando a linguagem de programação Java. Esse sistema operacional está revolucionando o mercado mundial de telefones celulares, pois ele é *Opensource* está se tornando cada vez presente em diferentes modelos de dispositivos móveis em torno de todo o mundo.

A plataforma *Android®* foi inicialmente desenvolvida pelo Google e posteriormente pelo *Open Handset Alliance*, o Google gerencia o projeto.

Segundo o *Open Handset Alliance*, o *Android®* foi construído partir do zero para

permitir que desenvolvedores criem aplicativos atraentes, e aproveitar ao máximo tudo o que os telefones têm para oferecer.

Foi construído para ser verdadeiramente livre. Por exemplo, um aplicativo pode chamar a qualquer das funcionalidades do núcleo do telefone, como fazer chamadas, enviar mensagens de texto ou usar a câmera do aparelho ao mesmo tempo em que se está utilizando o navegador ou executando o player de música. Os desenvolvedores podem criar interfaces mais ricas e mais coesas para os usuários. O *Android*® foi criado com a abertura do *Kernel* do *Linux*. Além disso, utiliza uma máquina virtual personalizada (Dalvik VM) que foi criada para aperfeiçoar os recursos de memória e hardware em um ambiente móvel.

O *Android*® é *Open Source*, que pode ser livremente ampliado para incorporar novas tecnologias de ponta que forem surgindo. A plataforma continuará a evoluir à medida que a comunidade de desenvolvedores trabalha em conjunto para construir aplicações móveis inovadoras segundo o *Open Handset Alliance*.



Figura 24 – Logotipo do Android®.

Fonte: (Próprio Autor, 2012).

2.3.1. Características

- **Características do Android® segundo o Open Handset Alliance.**

Todos os aplicativos são criados iguais.

O *Android*® não faz diferença entre os aplicativos do telefone e aplicativos de terceiros. Todos eles podem ser construídos para ter igualdade de acesso aos recursos do celular proporcionando aos usuários um amplo ambiente de serviços. Com os dispositivos construídos sobre a plataforma *Android*®, os usuários são capazes de adaptar completamente o telefone para os seus interesses. Eles podem trocar tela inicial do telefone, o estilo do marcador, ou qualquer um dos aplicativos. Eles podem instruir os seus telefones para usar a sua fotografia favorita e um aplicativo de visualização para lidar com a visualização de todas as fotos.

Quebrando limites do aplicativo

Por exemplo, um desenvolvedor pode combinar informações da web com os dados no celular de um indivíduo tais como os contatos do usuário, calendário ou localização geográfica, para fornecer ao usuário uma experiência mais relevante. Com o *Android*®, um desenvolvedor pode criar um aplicativo que permite aos usuários visualizar a localização de seus amigos e ser alertado quando estão nas proximidades dando-lhes uma chance de se conectar.

O desenvolvimento rápido e fácil de aplicações.

Android® possui uma gama de bibliotecas e ferramentas úteis que podem ser usados para a construção de variadas aplicações.

▪ Características do *Android*® segundo o Google

O *framework* de aplicação permite a reutilização e substituição de componentes interpretados pela Máquina Virtual Dalvik que é otimizada para dispositivos móveis. O Browser Integrado é baseado no software de código-fonte aberto *WebKit*. Os Gráficos otimizados são alimentados por uma biblioteca de gráficos personalizados 2D. Gráficos 3D são baseados no OpenGL ES 1.0. O SQLite é o banco de dados para armazenamento de dados estruturados. Oferece media de suporte para áudio comum, vídeo e imagem estática nos formatos (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF). Tecnologia GSM de telefonia, comunicação Bluetooth, EDGE, 3G e WIFI (dependente de hardware). Suporta Câmera, GPS, bússola e acelerômetro (dependente de hardware). Tem um rico ambiente de desenvolvimento, incluindo um dispositivo emulador, ferramentas de depuração, memória e desempenho de perfis, e um *plugin* para o Eclipse IDE e *Net Beans* IDE.

2.4. Arquitetura dos componentes do sistema *Android*®.

O diagrama da figura 1.1 mostra os principais componentes do sistema operacional *Android*®:

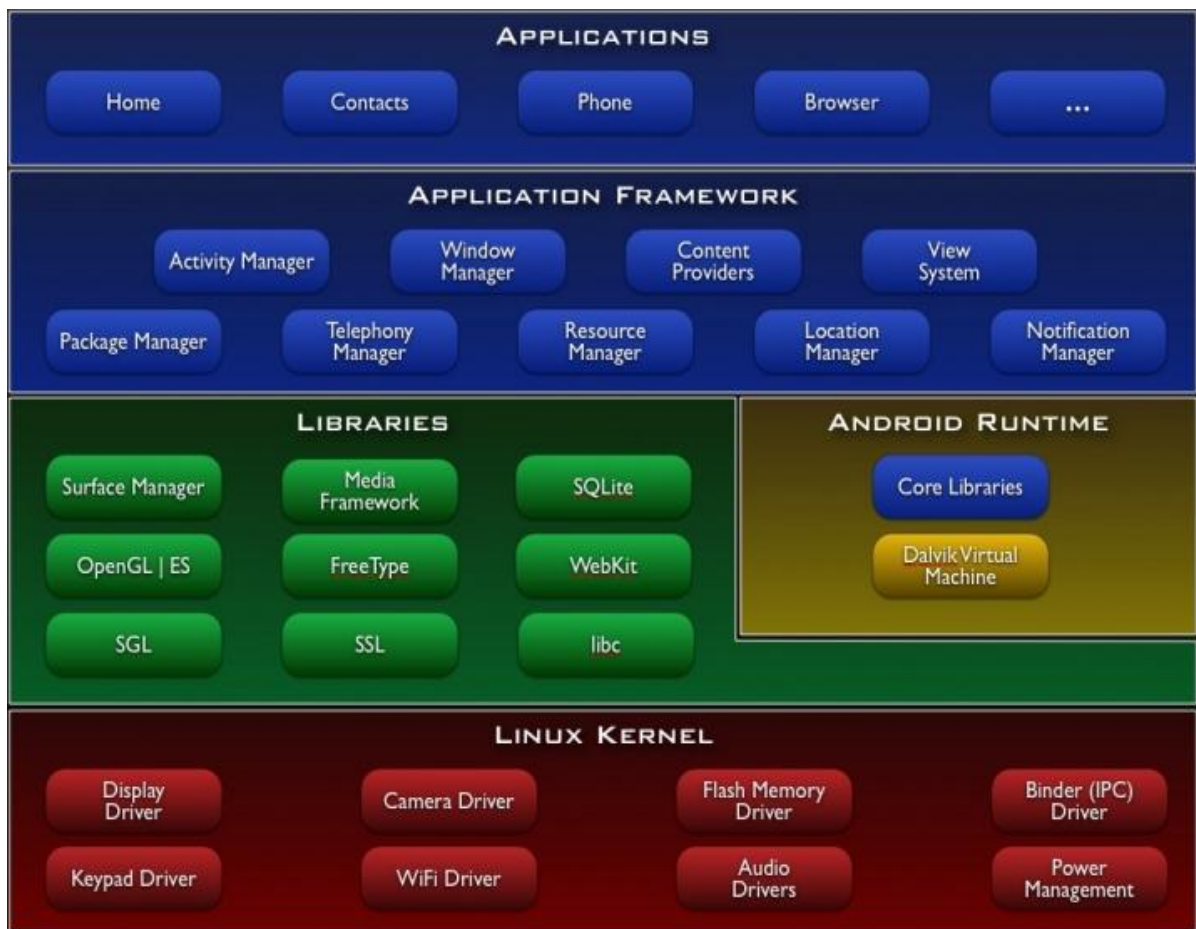


Figura 25 - Arquitetura dos componentes do sistema *Android*®.

Fonte: (AOSP, 2012).

2.4.1. *Linux Kernel*

O *Android*® é executado no sistema operacional *Linux* versão 2.6 ou posterior desenvolvido por Linus Torvalds que é do tipo *Unix* como sistema central de serviços, gerenciamento de memória, processos, segurança e etc.

2.4.2. *Libraries (Bibliotecas)*

O *Android*® inclui um conjunto de C / C++ bibliotecas usadas por diversos componentes do sistema *Android*®. Estas capacidades são expostas a desenvolvedores através da estrutura de aplicativos do *Android*®. Algumas das principais bibliotecas:

- ***System C library***

Uma implementação de derivados do BSD do sistema de biblioteca padrão C (libc), para dispositivos baseados em *Linux*.

- ***Media Libraries***

Baseada no *PacketVideo's OpenCORE*, a reprodução de apoio às bibliotecas e gravações de áudio e muitos formatos populares de vídeo, bem como arquivos de imagem estática, incluindo MPEG4, H.264, MP3, AAC, AMR, JPG e PNG.

- ***Surface Manager***

Gerencia o acesso ao display subsistema e sem problemas compósitos 2D e 3D gráfico camadas de múltiplas aplicações.

- ***WebKit***

É um moderno navegador web embutido.

- **OpenGL ES 1.0 | ES: Biblioteca gráfica 2D e 3D**

Uma aplicação baseada em OpenGL ES 1.0 APIs; as bibliotecas podem usar tanto o hardware de aceleração 3D (quando disponível) ou o otimizado software 3D;

- *FreeType*

Biblioteca de *Bitmaps* e vetores de renderização.

- *SQLite*

Um banco de dados relacional poderoso e leve à disposição de todos os aplicativos.

- *SGL*

É o subsistema de gráficos usado pelo *Android*®. *SGL* é o baixo nível de biblioteca gráfica implementada em código nativo que lida com a renderização. Ele trabalha em conjunto com outras camadas de nível superior para implementar a total de gráficos.

- *SSL*

Trabalha em conjunto com o *WebKit* para garantir a integridade dos dados trafegados na web.

2.4.3. *Android*® Runtime

O *Android*® inclui um conjunto de bibliotecas que fornece a maioria das funcionalidades disponíveis nas principais bibliotecas da linguagem de programação *Java*.

Cada aplicação *Android*® roda em seu próprio processo, com sua própria instância da máquina. O *Dalvik VM* executa os arquivos em executável *Dalvik* (. DEX) formato que é otimizada para o consumo mínimo de memória. A VM é baseada em registradores, e executa classes compiladas por um compilador de linguagem *Java* que foram transformadas para o formato dex. Incluído pela "dx" ferramenta. O *Dalvik VM* invoca o *kernel do Linux* para a funcionalidade subjacente como encadeamento e de baixo nível de gerenciamento de memória.

- ***Dalvik Virtual Machine***

A máquina virtual *Dalvik* é uma máquina virtual baseada em registradores, projetada e escrita por Dan Bornstein com contribuições de outros engenheiros do Google, como parte da plataforma de telefonia móvel *Android*®.

Ela é projetada para as necessidades de pouca memória, e também para permitir várias instâncias VM para executar uma vez, contando com o sistema operacional subjacente para o isolamento de processos, gerenciamento de memória.

A *Dalvik VM* é muitas vezes referida como uma Máquina Virtual *Java*, mas isto não é exato, pois em que opera não é *bytecode* *Java*. Em vez disso, uma ferramenta chamada *dx*, incluída no SDK do *Android*®, transforma os arquivos de classes *Java* compiladas por um compilador *Java* regular, em outro formato de arquivo de classe (o formato de *dex*).

2.4.4. *Applications* (Aplicativos)

É o conjunto de aplicativos básicos que são disponibilizados na plataforma, incluindo um cliente de e-mail, programa de envio de SMS, calendário, mapas, navegador, contatos, calendário, discador e etc.

- ***Application Framework* (Bloco de aplicação)**

É se localizam as *APIs* do *Android*® que são normalmente utilizadas pelas aplicações que executam sobre a plataforma. Ao fornecer uma plataforma de desenvolvimento aberta, o *Android*® o oferece aos desenvolvedores a capacidade de criar aplicações extremamente ricas e inovadoras. Os desenvolvedores estão livres para aproveitar o hardware do dispositivo, as informações de localização, execução de serviços, definir alarmes, para adicionar notificações na barra de status. São componentes de aplicação que são base no funcionamento dos aplicativos, são quatro.

- ***Activity***

Uma *Activity* ou atividade representa uma única tela com uma interface de usuário. Por exemplo, um aplicativo de e-mail pode ter uma atividade que mostra uma lista de e-mails novos, outra atividade para compor um e-mail, e outra atividade para ler e-mails. Embora as atividades possam trabalhar juntas para formar uma experiência de usuário coeso na aplicação de e-mail, uma é independente da outra. Como tal, uma aplicação diferente pode começar em qualquer uma dessas atividades. Por exemplo, uma aplicação de câmera pode iniciar a atividade na aplicação de correio eletrônico que compõe uma nova mensagem, a fim de que o utilizador a compartilhar uma imagem, (ANDROID DEVELOPERS, 2012).

- ***Broadcast Receiver***

É um componente que responde aos anúncios de todo o sistema. Muitos chamados do sistema, por exemplo, um chamado anunciando que a tela tiver desligado, a bateria está fraca, ou uma imagem foi capturada. Os aplicativos também podem iniciar as transmissões, por exemplo, para que outras aplicações saibam que alguns dados foram transferidos para o dispositivo e está disponível para eles usarem. Embora o receptor não exiba uma interface de usuário, eles podem criar uma notificação da barra de status para alertar o usuário quando um evento ocorre, (ANDROID DEVELOPERS, 2012).

- ***Services***

É um componente de aplicação que pode executar operações de longa duração e não fornece uma interface de usuário. Pode-se iniciar um serviço outro e que o serviço do aplicativo anterior vai continuar a executar em segundo plano, mesmo se o usuário mude para outro aplicativo. Além disso, um componente pode se ligar a um serviço para interagir com ele e até mesmo realizar a comunicação entre processos (IPC). Por exemplo, um serviço pode lidar com transações de rede, tocar música, executar arquivos, ou interagir com um provedor de conteúdo, (ANDROID DEVELOPERS, 2012).

▪ *Content Provider*

Seu objetivo é gerenciar o acesso a um conjunto estruturado de dados, encapsular os dados e proporcionar mecanismos para a definição de segurança dos dados. Os provedores de conteúdo são a interface padrão que conecta dados em um processo com código em execução em outro processo, (ANDROID DEVELOPERS, 2012).

2.5. *Google Play*

2.5.1. Histórico

O *Android® Market* foi criado no dia 23 de outubro de 2008 para ser o ambiente em que o Google disponibilizou para os programadores divulgarem os aplicativos desenvolvidos

Nesse repositório de aplicativos os desenvolvedores da plataforma podem publicar os aplicativos gratuitamente ou vender o aplicativo. Devido a estratégias comerciais no dia 6 de março de 2012 o nome passou de *Android® Market* para *Google Play*

2.5.2. Como Publicar no *Google Play*

Para o programador poder publicar no *Google play* será necessário ter uma conta no *gmail*, logo após acessar o domínio <https://play.google.com/apps/publish> denominado como *Developer Console* esse é o local onde o programador poderá subir o aplicativo.

Deverão ser seguidos os seguintes procedimentos:

1. O aplicativo (.apk) deve ser compilado em conjunto com uma chave privada.
2. O *package name (com.projeto)* é único para todos os projetos no *Google play*, portanto escolher um nome definitivo para o projeto, caso já exista um aplicativo com o mesmo nome de pacote o *Google play* não aceitará o .apk.
3. Disponibilizar as informações:

- Detalhes do Produto
 - Título.
 - Descrição.
- Recursos Gráficos
 - Duas Capturas de Telas do sistema.
 - Ícone de alta resolução
- Categorização
 - Tipo do Aplicativo
 - Categoria
 - Classificação do conteúdo
- Detalhes do Contato
 - Site
 - Email

- Política de Privacidade (Ainda não obrigatório)

Caso nunca tenha usado o *Google play* para submeter aplicativos aparecerá uma caixa de dialogo solicitando que seja paga uma taxa única de registro de \$25 dólares. O ambiente para pagamento disponibilizado pelo Google chama-se *Google Wallet* onde há a opção de pagar com cartão de credito, é aceito apenas cartão de crédito internacional.

2.6. Segurança em Android®

2.6.1. Introdução

O *Android*® é uma plataforma que nasceu com o intuito de ser uma plataforma aberta, tudo isso também com o intuito de que os aplicativos desenvolvidos em *Android*® desfrutem dos mais avançados recursos de hardware e software que a plataforma tem a oferecer. Sendo assim a plataforma deve oferecer um ambiente que garanta a segurança dos usuários, dados, aplicativos e a rede em que este conectado. Portanto é exigida uma arquitetura de segurança robusta e alguns programas de segurança rigorosos, (A.O.S.P., 2012).

A plataforma foi projetada para aumentar a segurança dos usuários e reduzir a probabilidade de ataques de softwares mal-intencionados.

2.6.2. Recursos de Segurança do Android®

A plataforma disponibiliza os seguintes recursos de segurança:

- Segurança robusta no nível do sistema operacional através do *kernel* Linux
- *Sandbox* de aplicação obrigatória para todas as aplicações
- Comunicação segura entre processos
- Assinatura de aplicativos
- Aplicação definidos pelo usuário e permissões

- **Segurança do Linux Kernel**

A base da plataforma *Android*® é o *kernel* do Linux. O *kernel* do Linux em si tem sido usado em larga escala por muitos anos, e é usada em milhões ambientes sensíveis de segurança. Através de sua história de ser constantemente pesquisado, estudado e aderido por milhares de desenvolvedores, o Linux tornou-se um núcleo estável e assegura a confiança de muitas empresas e profissionais de segurança. Adaptado para a plataforma *Android*®, o Linux fornece os seguintes recursos de segurança:

- Modelo de permissões baseado em usuários
- O isolamento dos processos
- Mecanismo extensível para seguro IPC
- A capacidade de remover partes desnecessárias e potencialmente inseguras do kernel.

O objetivo fundamental de segurança do *kernel* do Linux é isolar os recursos de todos os usuário com o objetivo de melhor proteger os recursos de cada usuário.

▪ **Sandbox de Aplicação**

O modelo de *sandbox* requer que um aplicativo declare explicitamente quais os dados ou recursos que deseja compartilhar com outras aplicações. Isso limita o aplicativo a utilizar apenas os recursos declarados restringindo o acesso ao sistema operacional, aumentando a segurança.

▪ **Comunicação entre processos**

Os processos podem se comunicar utilizando qualquer um dos mecanismos do UNIX. Exemplos incluem o sistema de arquivos ou bases locais. No entanto, as permissões do Linux ainda se aplicam.

▪ **Assinatura de aplicativos**

Todos os aplicativos do *Android*® (arquivos . Apk) devem ser assinados com um certificado cuja chave privada é mantida por seus desenvolvedores. Este certificado identifica o autor. O certificado não precisa ser assinado por uma autoridade de certificação, é perfeitamente admissível, e típico, aplicações *Android*® para utilizarem certificados auto assinados. Esse recurso possibilita que o aplicativo seja único e que tenha acesso aos recursos do sistema operacional. Aplicativos que tentam ser instalados sem serem assinados serão

rejeitados pelo dispositivo Android, e o Google play também não aceitará o aplicativo.

- **Permissões de recursos**

Um aplicativo *Android*® não tem permissões associados, o que significa que não pode fazer nada que possa afetar negativamente a experiência do usuário ou todos os dados do dispositivo. No momento da instalação do aplicativo, as permissões solicitadas pelo aplicativo são concedidas a ele pelo instalador de pacotes.

Nenhuma verificação de permissão é feita diretamente com o usuário, caso a permissão não seja concedida e qualquer tentativa do aplicativo utilizar um recurso que necessite de permissão irá falhar.

3. DESENVOLVIMENTO NA PLATAFORMA ANDROID®

Foi implementado um estudo de caso baseado na apostila cujo nome é Apostila de *Android®* - Programando passo a passo - 4ª Edição, com a autoria de Luciano Alves da Silva disponível no site www.apostilaandroid.ueuo.com. No momento do desenvolvimento do foram inseridas melhorias no aplicativo exemplo abstraído da apostila, pois havia bugs e também para melhorar a usabilidade e acessibilidade dos componentes *Android®* utilizados.

3.1. Ferramentas para o Desenvolvimento

3.1.1. IDE ECLIPSE

A aplicação desenvolvida foi um Cadastro de Pessoas no IDE ECLIPSE versão INDIGO no qual o usuário insere nome, profissão e idade e logo após efetua-se o cadastro apertando o botão Cadastrar Pessoa e o sistema volta para o menu principal. O usuário pode fazer de um único a vários cadastros. Para a visualização dos cadastros efetuados existira a opção Listar Pessoas que listara as pessoas na ordem em que foram cadastradas no sistema.

Desenvolvimento das telas da Aplicação Cadastro de Pessoas:

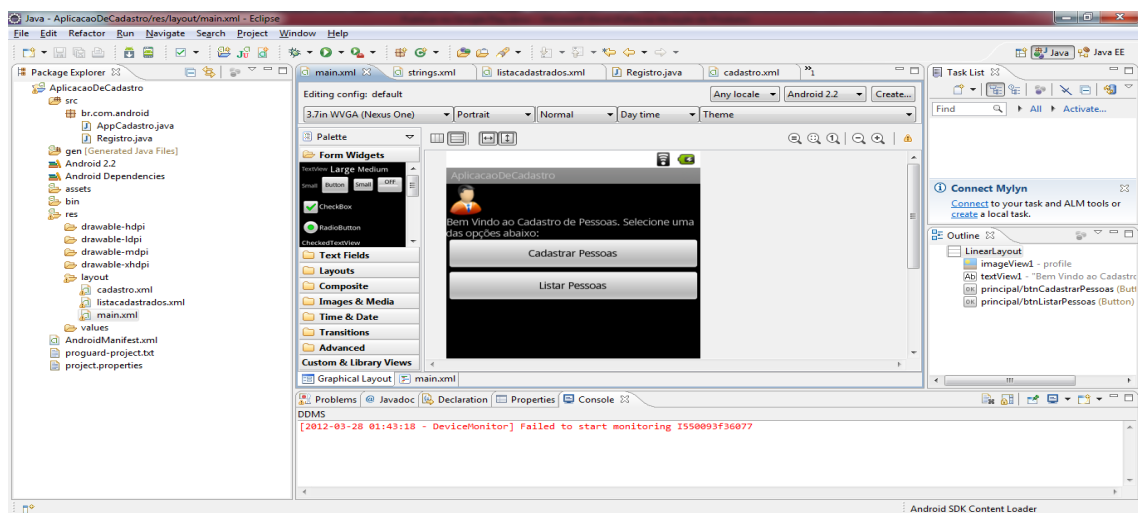


Figura 26 - Implementação do Main.xml.

Fonte: (Próprio Autor, 2012).

A figura 27 ilustra o desenvolvimento da lista do cadastro de pessoas:

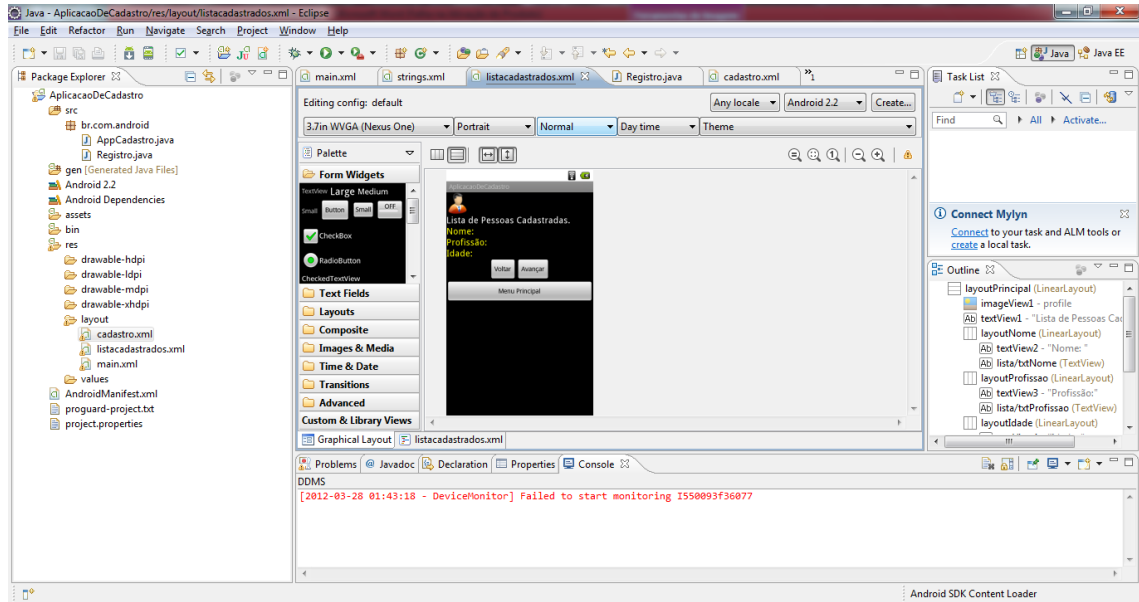


Figura 27 - Implementação da tela ListaCadastrados.xml.

Fonte: (Próprio Autor, 2012).

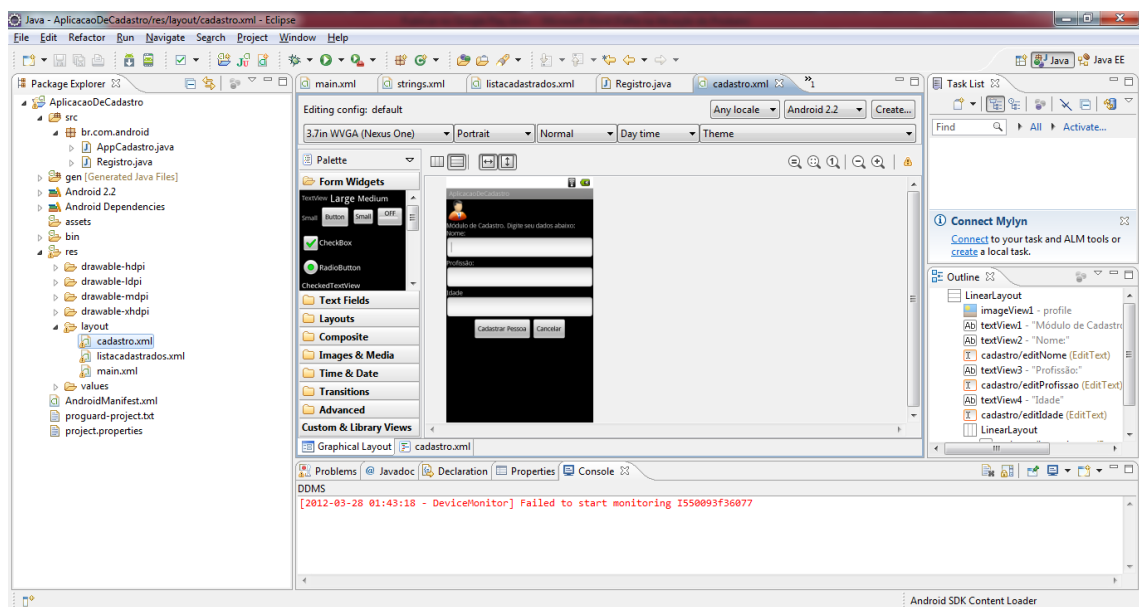


Figura 28 - Implementação da tela cadastro.xml.

Fonte: (Próprio Autor, 2012).

▪ Fluxo de telas e Funcionamento do sistema

O diagrama de atividades abaixo representa o fluxo de telas do sistema desenvolvido:

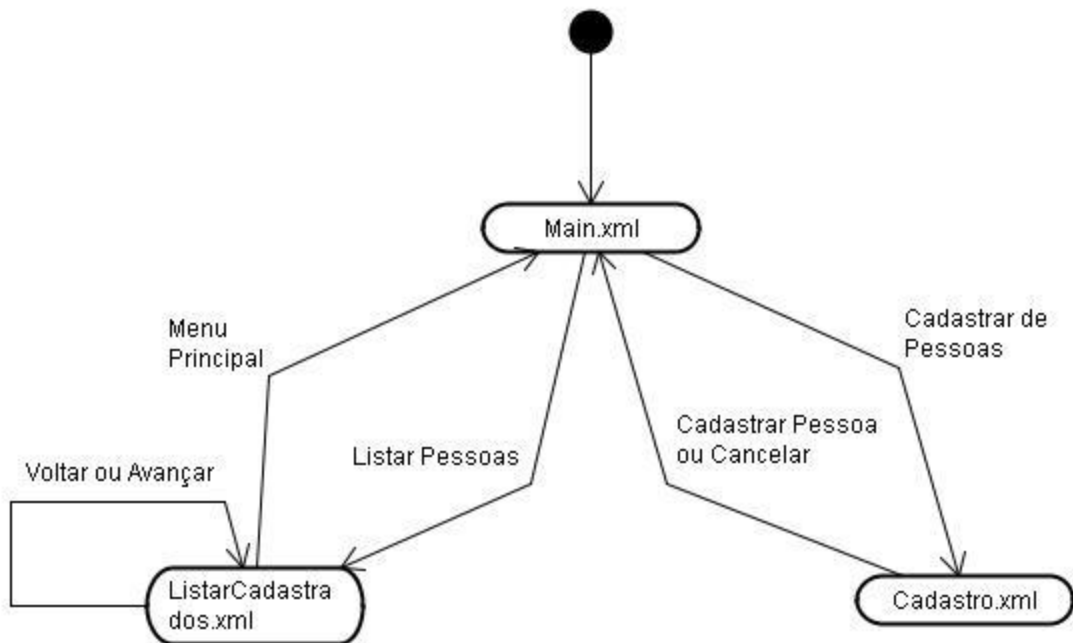


Diagrama 1 – Fluxo de Telas da Aplicação do Cadastro de Pessoas.

Fonte: (Próprio Autor, 2012).

▪ Funcionamento do Aplicativo Desenvolvido no Eclipse

O cadastro de pessoas se resume em cadastrar as pessoas e listas em seguida, isso apenas para mostrar o aplicativo desenvolvido.



Figura 29 - Escolhida a opção Cadastrar Pessoa.

Fonte: (Próprio Autor, 2012).

Agora serão inseridas duas pessoas, primeiro cadastro:

Nome: Rodolpho

Profissão: Programador

Idade: 24

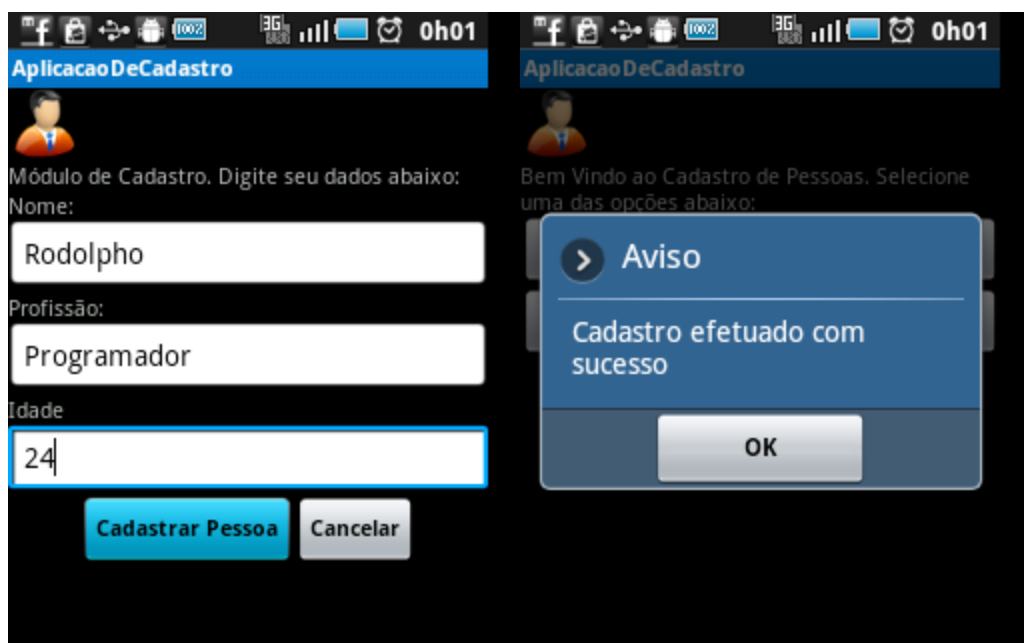


Figura 30 - Cadastro e Confirmação de cadastro no Layout Vertical.

Fonte: (Próprio Autor, 2012).

Segunda pessoa cadastrada:

Nome: Messi

Profissão: Boleiro

Idade: 23




Figura 31 – Cadastrando o boleiro Messi no layout horizontal.

Fonte: (Próprio Autor, 2012).

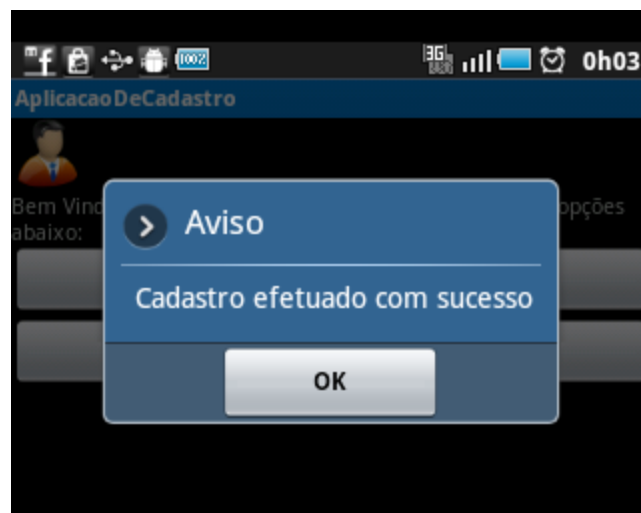


Figura 32 - Confirmação no Layout Horizontal.

Fonte: (Próprio Autor, 2012).

Cadastrados os usuários agora será feita a listagem das Pessoas:



Figura 33 - Escolhida a Opção Listar Pessoas.

Fonte: (Próprio Autor, 2012).

Ao acessar a lista já é mostrada a primeira pessoa das cadastrada:

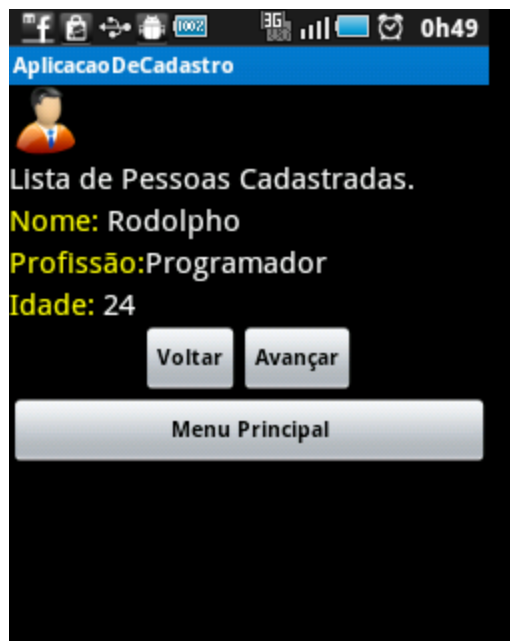


Figura 34 - Visualização da Primeira pessoa cadastrada, Rodolpho.

Fonte: (Próprio Autor, 2012).

E na sequência ao apertar Avançar já mostra a segunda pessoa cadastrada:

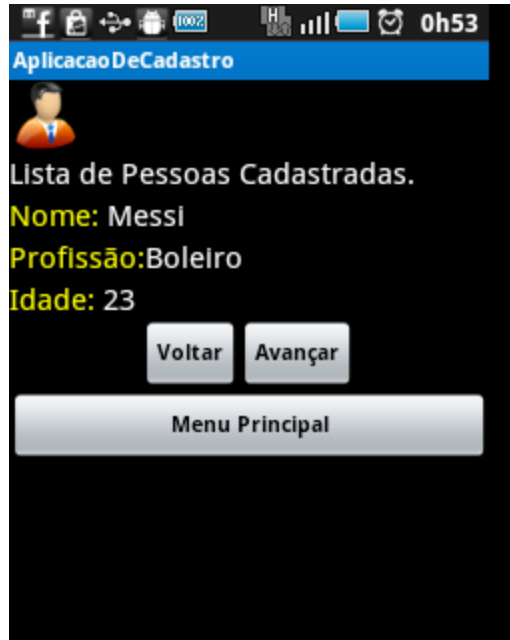


Figura 35 - Visualização da segunda pessoa cadastrada, Messi.

Fonte: (Próprio Autor, 2012).

Foram desenvolvidos dois exemplos de aplicações que utilizam o *GOOGLE MAPS API* que será explicado nos próximos tópicos, à primeira com a finalidade de estudo da ferramenta, aquisição de conhecimento, e obter as ferramentas e práticas ideais para o desenvolvimento de aplicações de Mapas para o *Android®* utilizando o *GOOGLE MAPS API*. A segunda aplicação foi desenvolvida para demonstrar tanto a localização atual exata como o deslocamento realizado utilizando o GPS do *Android®*.

3.1.2. Motodev

O *MotoDev Studio* é uma ferramenta desenvolvida pela *Motorola*®, baseado no Eclipse permite o desenvolvedor começar e ajuda os experientes a desenvolver o trabalho rapidamente, o decorrer do projeto será desenvolvido no *Motodev*.

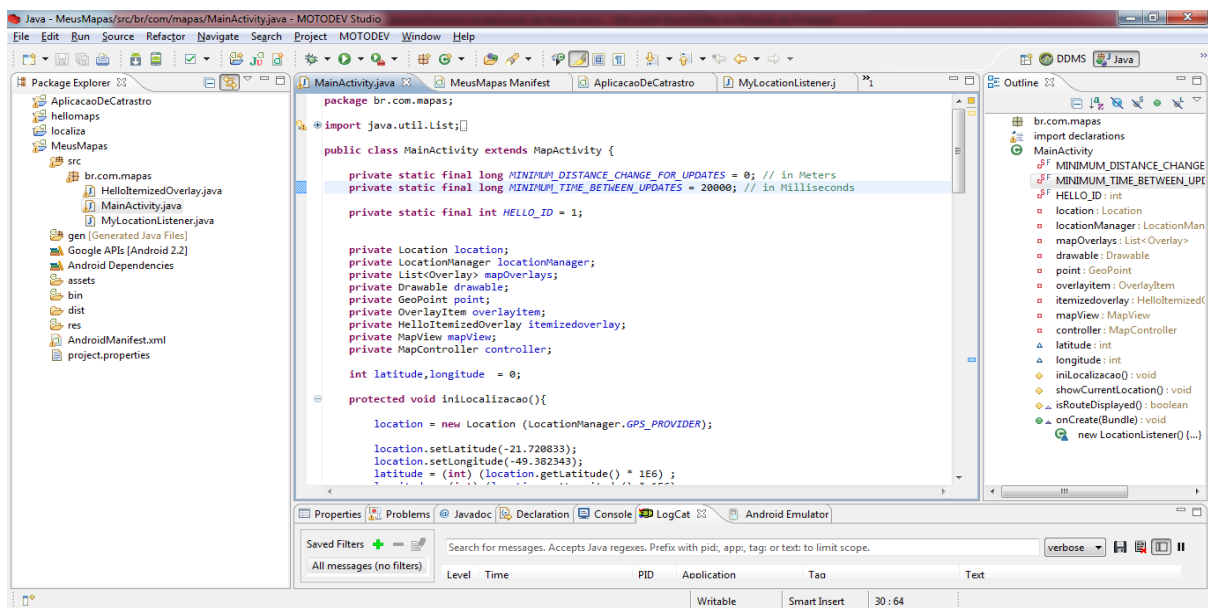


Figura 36 - IDE do Eclipse adaptado pela Motorola®, cujo Motodev.

Fonte: (Próprio Autor, 2012).

▪ Principais recursos oferecidos pelo Motodev segundo a Motorola®

APP Validator: Detecta e repara condições inadequadas no código fonte, tais como permissões ausentes e configurações que conflitam com as especificações do dispositivo.



Figura 37 - APP Validator.

Fonte: (Próprio Autor, 2012).

Code Snippets: Disponibiliza exemplos de códigos dos componentes que serão utilizados na programação.

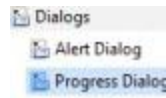


Figura 38 - Code Snippets.

Fonte: (Próprio Autor, 2012).

Code Generation Wizards: Dá a opção ao desenvolvedor de iniciar a criação de um projeto com base em *templates* com exemplos de aplicações.



Figura 39 - Code Generation Wizards.

Fonte: (Próprio Autor, 2012).

Database Management: É um gerenciador de banco de dados SQLite. Permite implantar e gerenciar bases de dados nos aplicativos *Android®*.

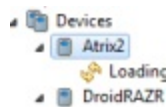


Figura 40 - Database Management.

Fonte: (Próprio Autor, 2012).

4. PROCESSAMENTO DE MAPAS EM DISPOSITIVOS MÓVEIS

4.1. Google Maps API

São as bibliotecas na qual são oferecidos recursos de mapeamento para o desenvolvimento de aplicações, na qual é fornecida para a plataforma *Android*® a biblioteca *com.google.Android.maps*. Essa biblioteca oferece download, renderização, e cache de Mapas, bem como um variedade de opções de exibição e controles.

A classe chave na biblioteca Maps é a *MapView*, subclasse da *ViewGroup* na biblioteca padrão *Android*®. O *MapView* exibe um mapa com os dados obtidos a partir do serviço Google Maps. Ele também fornece todos os elementos necessários para os utilizadores para controlar o mapa.

Em geral, a classe *MapView* fornece um ambiente em torno da API do Google Maps que permite ao aplicativo manipular dados do *Google Maps API* através de métodos de classe, e que permite trabalhar com mapas de dados.

Para que a classe *MapView* tenha acesso a dados do *Google Maps API*, é necessário se registrar com o serviço Google Maps e concordar com os Termos de Serviço aplicáveis para que a classe *MapView* seja capaz de obter dados do Google Maps, é necessário obter uma API KEY.

4.1.1. Obtendo uma MAP API KEY

O registro de um API do Google Maps é simples, gratuito:

Deve-se registrar a impressão digital MD5 do certificado que será utilizado para assinar o aplicativo e acessar o serviço de registo de mapas (<https://developers.google.com/maps/documentation/Android@/maps-api-signup?hl=pt-BR>) e inserir a impressão digital MD5 da chave privada do aplicativo, em seguida, será fornecida uma MAPS API Key, que está associada com o certificado do aplicativo compilado. No nos exemplos desenvolvidos a seguir o repositório de chaves se localizavam na pasta *.android* cujo nome do repositório era *debug.keystore*, no pacote Java existe uma ferramenta chamada *Keytool* no qual é necessário para extrair a impressão digital MD5 da chave privado do repositório que sairá no formato E3:65:3D:1A... em um conjunto de 16 pares de caracteres

separados por 2 pontos(:). Ao extrair os caracteres o próximo passo será acessar a página do Serviço de Registro de Mapas para registrar a aplicação.

A chave gerada pelo site do *Google Maps Api* deve ser inserida em **Android®:apiKey="VALOR DA CHAVE "**. O site irá gerar o XML correspondente ao *MapView* com a chave de utilização da API, conforme mostrado no trecho de código abaixo.

```
< com.google.Android®.maps.MapView
    xmlns:Android®="http://schemas.Android®.com/apk/res/Android®"
    Android®:apiKey="VALOR DA CHAVE "
    Android®:id="@+id/mapview"
    Android®:layout_width="fill_parent"
    Android®:layout_height="fill_parent"
    Android®:clickable="true"/>
```

Como podemos ver acima na *tag com.google.android.maps.MapView* o item *android:apiKey* se refere a onde deve ser inserida a chave gerada pelo site que a aplicação possa obter acesso total aos recursos do Google Maps. Foram desenvolvidos dois exemplos de aplicações para mostrar o funcionamento da API.

4.1.2. Exemplo 1 - Fixar Marcadores em diversos pontos do Mapa

Foi desenvolvido um aplicativo para marcar diversas localizações em determinadas coordenadas geográficas, os marcadores são imagens (Robô do *Android*®) que destacam o ponto geográfico, as informações sobre latitude e longitude disponibilizadas abaixo.



Figura 41 – A aplicação está destacando os Pontos Geográficos instanciados via código.

Fonte: (Próprio Autor, 2012).

Os pontos geográficos escolhidos foram o Brasil, Japão e México, ao clicar nos marcadores é mostrada uma mensagem.

1.) Fortaleza – Brasil

Latitude → -3733030

Longitude → -38498932



Figura 42 – Aproximação do ponto do Brasil – Fortaleza.

Fonte: (Próprio Autor, 2012).

Ao tocar no ícone do robô irá aparecer uma mensagem referente a localização, conforma a figura 43.



Figura 43 – Mensagem ao “clicar” no marcador.

Fonte: (Próprio Autor, 2012).

2.) Japão – Yokohama

Latitude → 35410000

Longitude → 139460000



Figura 44 – Aproximação do ponto Yokohama – Japão.

Fonte: (Próprio Autor, 2012).

Ao tocar no ícone do robô irá aparecer uma mensagem referente a localização, conforma a figura 45.



Figura 45 – Mensagem ao clicar no marcador.

Fonte: (Próprio Autor, 2012).

3.) Cidade do México – México

Latitude → 20692592

Longitude → -99580077



Figura 46 – Aproximação do ponto Ciudad de México.

Fonte: (Próprio Autor, 2012).

Ao tocar no ícone do robô irá aparecer uma mensagem referente a localização.



Figura 47 – Mensagem do Marcador.

Fonte: (Próprio Autor, 2012).

4.1.3. Exemplo 2 – Localização Atual colhida via GPS do dispositivo

Foi desenvolvido um aplicativo para efetuar a leitura do GPS do Smartphone e mostrar a localização atual do indivíduo conforme mostra a figura 48. O novo aplicativo foi baseado no primeiro mapa desenvolvido. Houve a necessidade de fazer uma pesquisa sobre os componentes necessários para que tal aplicativo fosse capaz de fazer o procedimento.

O *Android*® dá seu acesso às aplicações de serviços de localização suportados pelo dispositivo através das classes no pacote *android.location*.

O *LocationManager* é componente central da estrutura local e o *LocationManager*, que fornece APIs para determinar a localização.

O método *LocationListeter* é responsável por atualizar as coordenadas buscadas pelo GPS automaticamente, tanto por determinado tempo ou distancia percorrida. Abaixo o funcionamento do aplicativo desenvolvido.



Figura 48 – Localização Inicial – Trevo de Pongai – SP, Latitude: -21.720833
e Longitude: -49.382343.

Fonte: (Próprio Autor, 2012).

A figura 49 ilustra o funcionamento da aplicação com o GPS que foi habilitado no Univem e o marcador foi gerado conforme o caminho percorrido ate o Xerox do bola. A cada leitura do GPS é impresso um robozinho.



Figura 49 – Leitura de GPS e os pontos que a pessoa esteve localizada recentemente.

Fonte: (Próprio Autor, 2012).

Ao mesmo tempo em que a localização é marcada com robô, a barra de notificações mostra o valor com a latitude e a longitude colhidas pelo dispositivo.



Figura 50 – Notificação da Coordenada atual.

Fonte: (Próprio Autor, 2012).

5. SISTEMA DE APOIO A EVENTOS

5.1. Introdução

“[...] Com a finalidade de ampliar a esfera de seus relacionamentos inerentes ao convívio em família, no trabalho, escola ou no lazer e de quebrar a rotina dos afazeres diuturnos o homem cria, organiza e participa de reuniões que genericamente são chamadas de eventos[...]”, (GIACAGLIA, 2004).

Ou seja, o evento tem como objetivo reunir pessoas por um determinado período específico para um acontecimento, discussão assunto específico para um público alvo ou ocasiões dos mais variados tipos.

Os eventos podem ser denominados dentro de uma tipologia específica e variados, conforme a tabela 2:

Almoço	Debate	Painel
Assembléia	Desfile	Palestra
Bazar	Encontro	Pedra fundamental
Brainstorming	Entrevista coletiva	Performance
Brunch	Excursão	Pré-estréia
Café da manhã	Exposição	Posse
Campanha	Feira	Premiação
Campeonato	Festa	Regata
Carnaval	Festival	Retrospectiva
Casamento	Formatura	Reunião
Chá da tarde	Fórum	Roadshow
Churrasco	Gincana	Rodada de negócios
Coffee-break	Happy hour	Rodeio
Coletiva de imprensa	Inauguração	Salão
Colóquio	Jantar	Sarau
Comemoração	Jornada	Semana
Competição	Lançamento de livro	Seminário
Comício	Lançamento de produto	Show
Concerto	Leilão	Showcasing
Concilio	Mesa-redonda	Solenidade
Conclave	Micareta	Sorteio
Concurso	Missa	Teleconferência
Conferência	Mostra	Torneio
Confraternizações	Noite de autógrafos	Vernissage
Congresso	Oficina	Videoconferência
Convenção	Olimpíada	Visita
Coquetel	Open day	Workshop
Coral	Opera	Etc.
Curso	Outorga de títulos	

Tabela 2 – Tipologia de Eventos.

Para que o evento consiga atingir os seus objetivos é necessário planejamento adequado e toda uma estrutura profissional com gente qualificada para fazer com que tudo corra dentro dos conformes para que tanto os participantes quanto os organizadores fiquem satisfeitos com o resultado dessa integração de pessoas.

Os organizadores precisam se atentar a questões importantes como providenciar estrutura adequada ao evento proposto, prover informações ao público relativas ao evento, agenda do evento, informações sobre a cidade como locais úteis como rede bancária, hoteleira, restaurantes, pontos de vendas de ingresso, rede de transporte e etc. É necessária a difusão pontual das informações para que público fique informado sobre o andamento do evento.

Pensando nisso foi desenvolvido um software chamado Eloc, nome baseado na junção das palavras Evento e Localização, ele desenvolvido para o *Android*® com o objetivo de apoiar os organizadores na difusão das informações do evento para o público alvo.



Figura 51 – Logotipo do Eloc.

Fonte: (Próprio Autor, 2012).

5.2. Requisitos

Os organizadores do evento têm como objetivo de prover as seguintes informações:

- Informar dados e informações sobre o evento
- Informar a agenda do evento
- Disponibilizar informações sobre a organização do evento
- Informar aos participantes do evento lugares úteis como o local do evento, hotéis, restaurantes, bancos e etc, conforme a necessidade.

5.3. Diagrama de Classes

5.3.1. Classe MainActivity.java

A classe *MainActivity* é responsável por iniciar as tarefas do sistema, estendendo a classe *Activity* que diz para o *Android*® que essa classe que ira interagir com o usuário. A classe terá a função de verificar se existe conexão com a internet, carregar as *tabs* com os componentes *WebView* que irão posteriormente carregar as páginas referentes ao evento e os botões para a consulta das localizações.

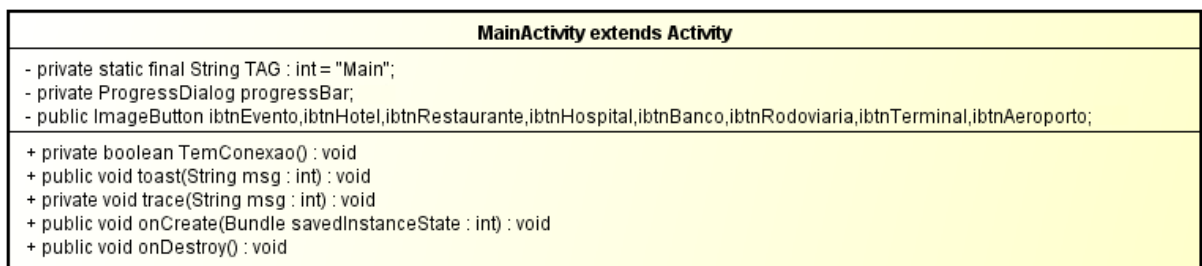


Diagrama 2 – Diagrama de Classe da classe MainActivity.

Fonte: (Próprio Autor, 2012).

5.3.2. Classe MyWebViewClient.java

A classe *MyWebViewCliente* manipula o componente *WebView* carregado na classe *MainActivity*. Por exemplo, no sistema desenvolvido no momento em que o sistema era iniciado, ao invés das paginas serem carregadas nas abas com o *WebView*, o navegador do *Android*® era chamado, e isso não era funcional para o sistema. Com essa classe instanciada com a método *shouldOverrideUrlLoading* esse problema não ocorre.



Diagrama 3 – Diagrama de Classes da classe MyWebViewClient.

Fonte: (Próprio Autor, 2012).

5.3.3. Classe Mapa.java

A classe Mapa é uma atividade de mapa estendendo a classe *MapActivity* que é responsável por renderizar o mapa com as marcadores conforme a opção escolhida na classe *MainActivity*. A classe terá métodos para conexão com o banco de dados online *Mysql* para buscar as informações de latitude e longitude e conseqüentemente marcar os pontos no mapa conforme a quantidade encontrada no banco de dados. Haverá também métodos para busca da posição atual do aparelho via GPS.

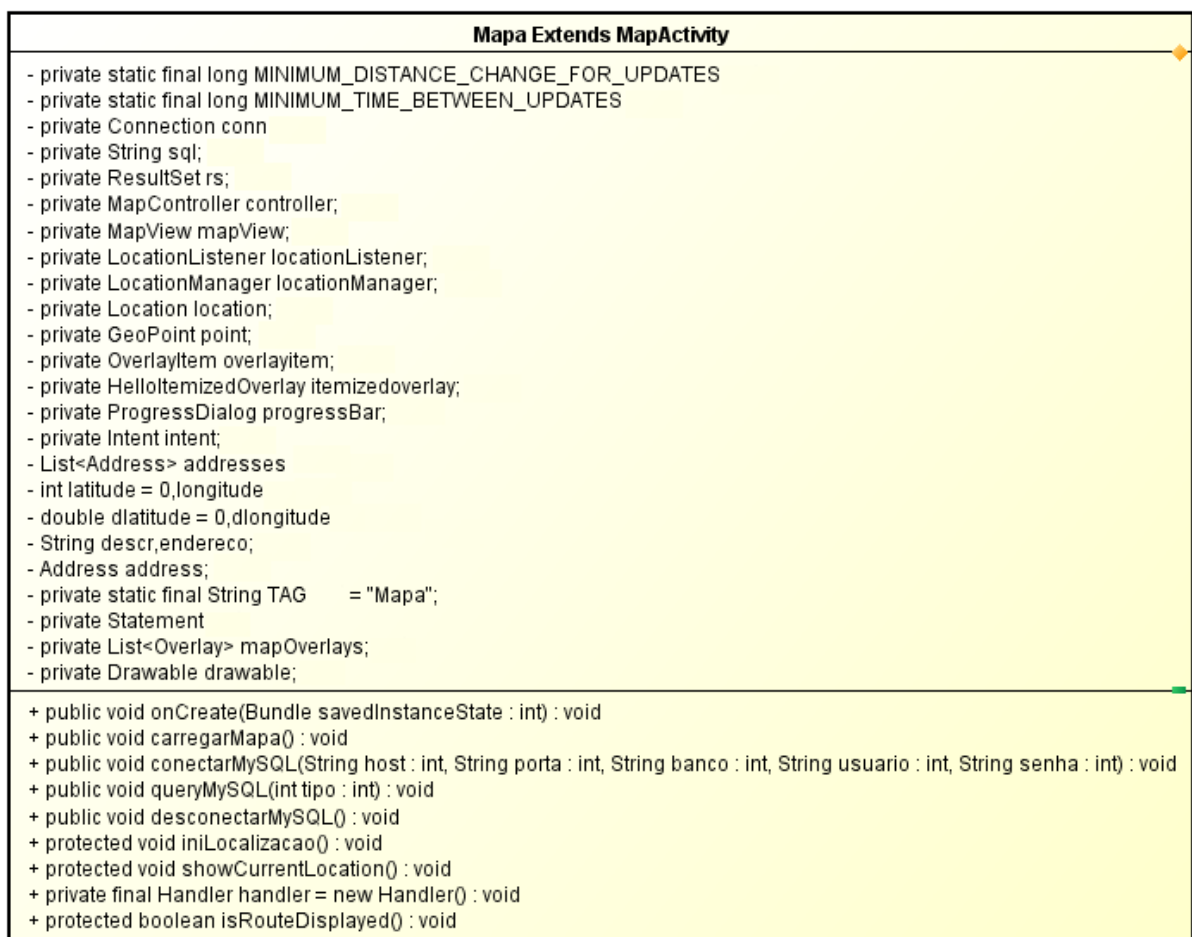


Diagrama 4 – Diagrama de Classe da classe Mapa.

Fonte: (Próprio Autor, 2012).

5.3.4. Classe HelloItemizedOverlay

A classe *HelloItemizedOverlay* é responsável por carregar, manipular o mapa na quando instanciada na classe Mapa.

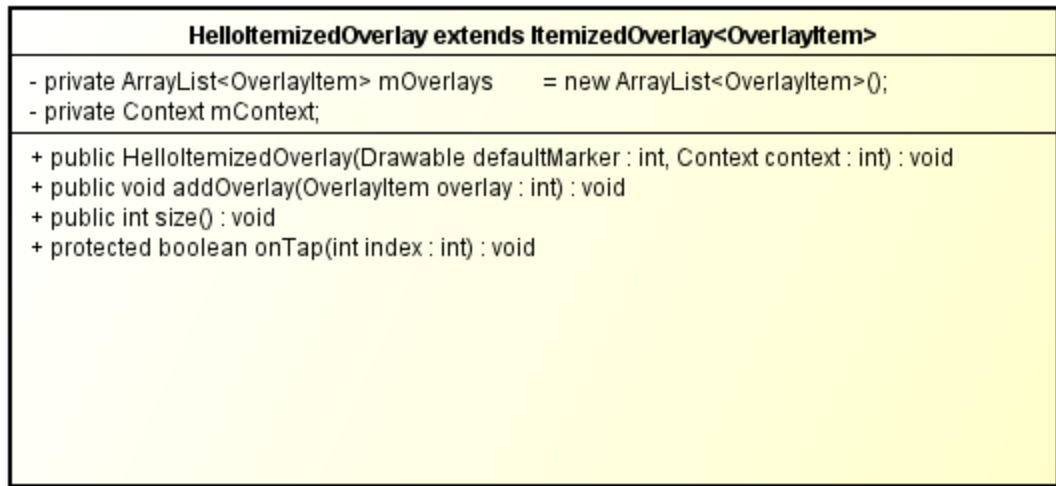


Diagrama 5 – Diagrama de Classe da classe HelloItemizedOverlay.

Fonte: (Próprio Autor, 2012).

5.3.5. Diagrama de classes relacionadas

No diagrama 6 é possível visualizar as classes relacionadas:

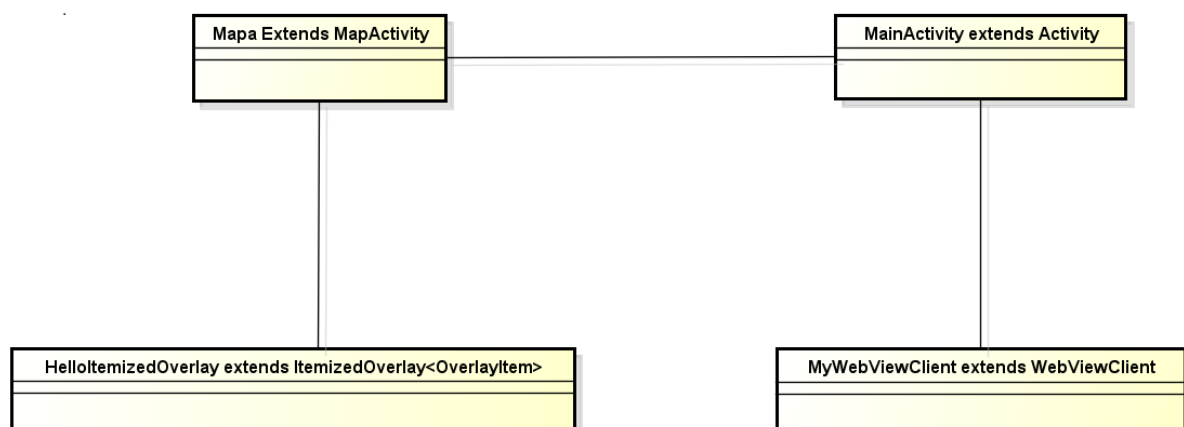


Diagrama 6 – Diagrama de Classes do sistema.

Fonte: (Próprio Autor, 2012).

5.4. Procedimento para uso do Google Maps API.

O procedimento utilizado para ao projeto é o mesmo que foi descrito no item 3.1, a diferença é que foi necessária a criação de um novo repositório de chave privada, pois *debug.keystore* funciona apenas para versões de teste e não para versões release. Assim o sistema criado não conseguiu se conectar ao serviço do Google e conseqüentemente acarretando no não aparecimento dos componentes do mapa na aplicação.

A ferramenta *Motodev* oferece opção para a criação de um novo repositório de chaves privadas chamada *Signing and Keys*.

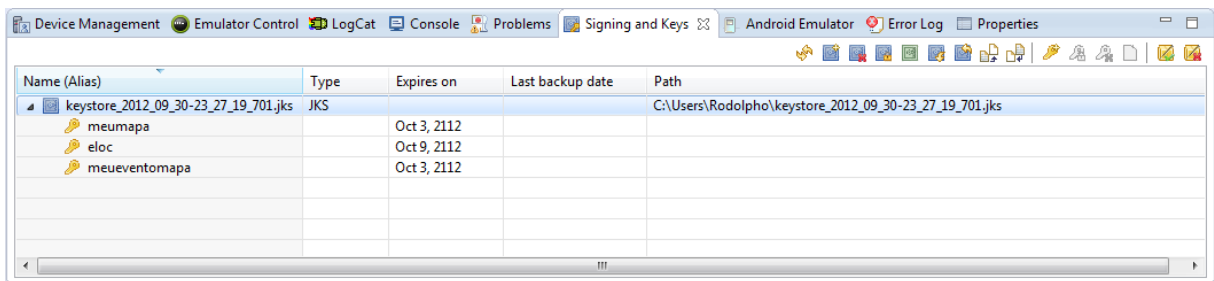


Figura 52 – Ferramenta para a criação do repositório de chaves.

Fonte: (Próprio Autor, 2012).

Na imagem pode se visualizar o repositório de chaves que está localizado na pasta C:\Users\Rodolpho\keystore_2012_09_30-23_27_19_701. jks com as chaves privadas criadas no caso meumapa, eloc e meueventomapa,. Para extrair o MD5 será necessário acessar o aplicativo Keytool na pasta *bin* do Java via comando:

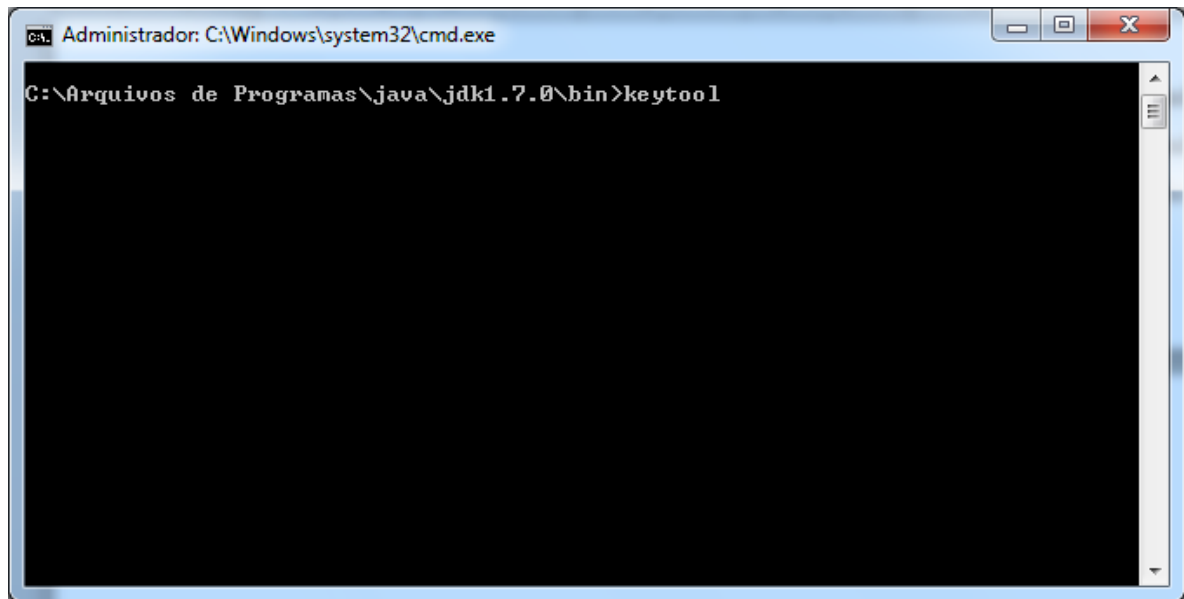


Figura 53 – Caminho do aplicativo Keytool.

Fonte: (Próprio Autor, 2012).

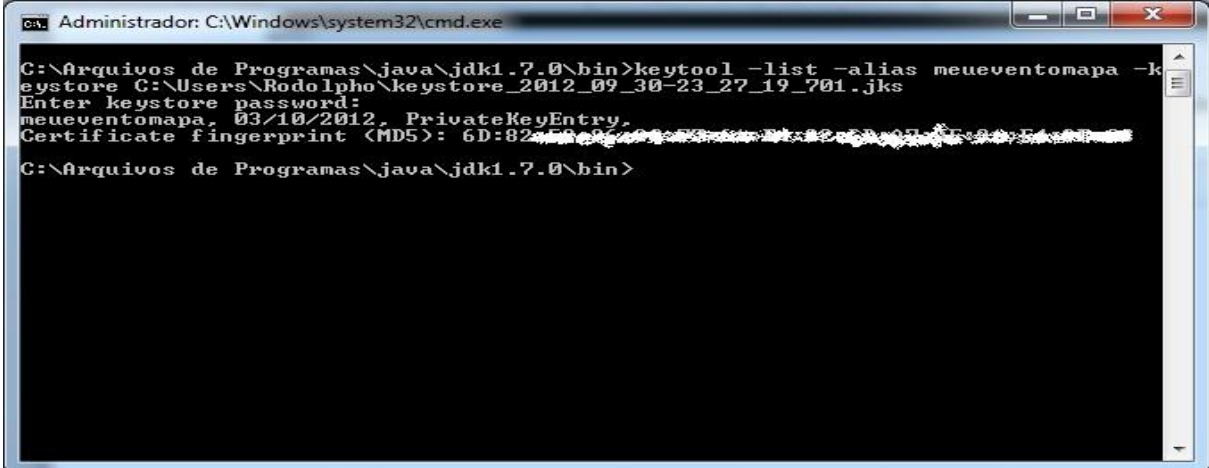
A frente do comando *keytool* deve-se inserir os comandos na sintaxe abaixo:

```
keytool -list -alias alias_name -keystore my-release-key.keystore
```

Onde:

- *list*: Comando para listar as chaves
- *alias*: nome da chave a ser aberta
- *keystore*: Caminho do repositório de chaves

Logo após os dados inseridos e será solicitada a senha do repositório de chaves e posteriormente gerada impressão digital MD5 da chave.



```
Administrador: C:\Windows\system32\cmd.exe
C:\Arquivos de Programas\java\jdk1.7.0\bin>keytool -list -alias meueventomapa -keystore C:\Users\Rodolpho\keystore_2012_09_30-23_27_19_701.jks
Enter keystore password:
meueventomapa, 03/10/2012, PrivateKeyEntry,
Certificate fingerprint (MD5): 6D:82:4F:0A:02:0A:02:0A:02:0A:02:0A:02:0A:02:0A
C:\Arquivos de Programas\java\jdk1.7.0\bin>
```

Figura 54 – Gerando a impressão digital MD5 da chave privada

Fonte: (Próprio Autor, 2012).

Feita a impressão digital da chave MD5, acessar a pagina <https://developers.google.com/maps/documentation/Android@/maps-api-signup?hl=pt-BR> do serviço *Google Maps Api* e inserir a chave gerada e Clicar em GENERATE API KEY.

O serviço irá gerar o código XML já com a chave do *Android Maps*, bastando apenas utilizar na aplicação.

The screenshot shows the Google Maps API registration page. At the top left is the Google Maps BETA logo. The main heading is "API do Google Maps". Below it, there is a breadcrumb trail: "Página inicial do Google Code > API do Google Maps > Inscrição na API do Google Maps". A yellow banner reads "Obrigado por se inscrever para uma chave de API do Android Maps!". Below this, it says "Sua chave é:" followed by a text box containing the API key: "Oz8Q_eFMY8YeHK1_eJWU1_b1TVVHr_VGd11kZ1E4-z1-z3". Below that, it says "Esta chave é válida para todos os aplicativos assinados com seu certificado cuja digital é:" followed by a text box containing the SHA-1 fingerprint: "6D:82_53_86_88_5B_64_54_88_8B_1B_1B_61_54_8F_86". Finally, it says "Aqui está um exemplo de layout xml para ajudá-lo a começar a usar os recursos de mapeamento:" followed by a code block containing the XML snippet:


```
<com.google.android.maps.MapView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:apiKey="Oz8Q_eFMY8YeHK1_eJWU1_b1TVVHr_VGd11kZ1E4-z1-z3"
  />
```

Figura 55 – Gerando o layout xml para a utilização dos recursos de mapeamento.

Fonte: (Próprio Autor, 2012).

5.5. Aplicação Web

5.5.1. Introdução

Para o cadastramento dos dados relativos ao evento e as localizações relacionadas foi desenvolvido um site em linguagem PHP com banco de dados Mysql, para ambos não há nenhum custo de uso. O site está hospedado no domínio www.meumapa.kinghost.net, onde foram criados cadastros para a alimentação das informações referentes ao evento.

Nesse site será onde serão alimentadas as informações referentes ao evento e um cadastro de localizações úteis onde os organizadores poderão categorizar as localizações por tipo inserindo valores como latitude e a longitude de uma ou mais localidades para a categoria, o site possuirá os seguintes cadastros.

- *Login* de usuário
- *Menu* Principal
- Cadastro de Dados do Evento
- Cadastro de Agenda
- Cadastro de Dados dos Organizadores
- Cadastro de Localizações
- Gestão de Usuários do *Site*

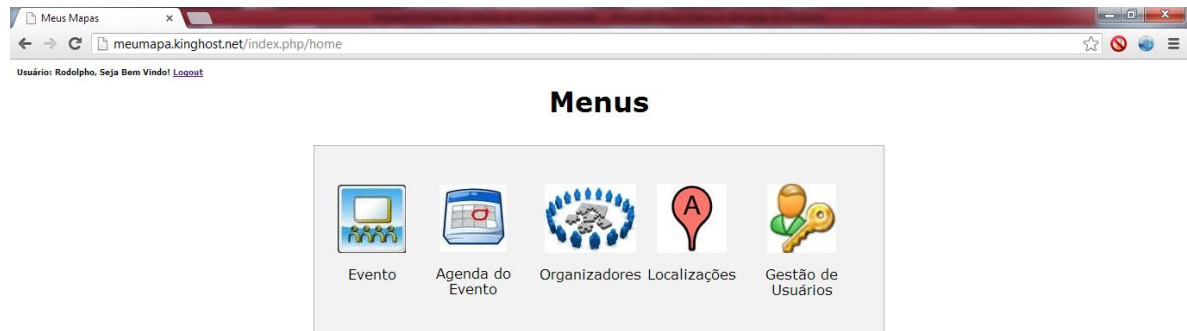


Figura 56 - Menu principal do site.

Fonte: (Próprio Autor, 2012).

5.5.2. Tecnologias Utilizadas

- **PHP**

O PHP é uma linguagem de *script* amplamente utilizada de propósito geral que é especialmente adequado para desenvolvimento *Web* e pode ser incorporado em HTML. PHP (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral, muito utilizada e especialmente guarnecida para o desenvolvimento de aplicações *Web* dentro do HTML, (PHP, 2012).

O que distingue o PHP de algo como *Javascript* no lado do cliente é que o código é executado no servidor, gerando HTML que é então enviado para o cliente. O cliente receberia os resultados da execução desse *script*, mas não saberia como é o código fonte. Você pode inclusive configurar seu servidor para processar todos os seus arquivos HTML como PHP, e então não haverá nenhum modo dos usuários descobrirem que se você usa essa linguagem ou não. O php extremamente simples para um iniciante, e ao mesmo tempo oferece muitos recursos para programadores profissionais, (PHP, 2012).

Para o desenvolvimento local será necessário instalar um servidor web, geralmente o Apache incorpora o php e o banco de dados *Mysql*. Existem alguns programas disponíveis na internet que já instalam tudo de uma vez, casos como dos *Xampp* ou o *Zend Server*. Existem também Frameworks que auxiliam no desenvolvimento em php, deixando o desenvolvimento e aprendizado mais rápido e tornando também o código, limpo, estruturado e de fácil entendimento. Ferramenta utilizada para o desenvolvimento do site foi o Framework Codeigniter é baseado em camada M.V.C.

▪ Codeigniter

CodeIgniter é um kit de ferramentas para construção de aplicações web usando PHP. O objetivo é permitir o desenvolvimento de projetos mais rapidamente, através de um conjunto de bibliotecas para as tarefas mais comuns, bem como uma interface simples e uma estrutura lógica para acessar essas bibliotecas. O framework é licenciado em código aberto portanto há a liberdade para fazer o que quiser com a plataforma, é um framework leve e rápido, as *url's* são limpas definidas como *Url* amigável ou seja baseada em seguimentos. Com o funcionamento baseado em nas camadas *Model-View-Controller*, existe a separação dos códigos lógicos e códigos da camada de interface. Incluem também bibliotecas para a conexão com o banco de dados, manipulação de imagens e entre outras, (CodeIgniter, 2012).

▪ Fluxo de funcionamento do Codeigniter

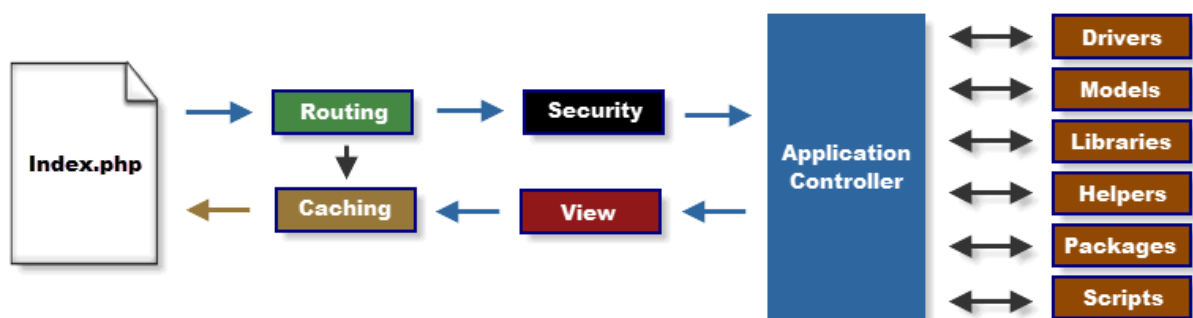


Figura 57 – Fluxo de funcionamento do Codeigniter.

Fonte: (Codeigniter, 2012.)

O *index* serve como controlador de frente e inicializa os recursos básicos para executar o *codeigniter*. O *routing* examina o pedido http e determina o que de ser feito com ele. Se um arquivo de cache existe, ele é enviado ao navegador. Antes de o *Application Controller* ser carregado ele passa pela filtragem de segurança pelo item *Security*, depois disso o *Application Controller* carrega os modelos, e bibliotecas para que a solicitação HTTP seja atendida. Logo após a *View* é carregada com os dados processados e seguida ela envia os dados ao navegador. Se o *caching* estiver ativado a *view* será armazenada para no caso de futuras solicitações

▪ MVC

O MVC (*Model-View-Controller*) é uma metodologia de desenvolvimento que faz a separação da parte lógica e da apresentação:

Model: São as classes de persistência que são responsáveis por recuperar, incluir, alterar e excluir os registros do banco de dados.

View: é a camada responsável pela renderização dos dados ou usuário, ou seja, é a interface ao usuário com a aplicação.

Controller: Funciona como uma camada intermediária entre o *model* e a *view*, ela gerencia as ações e faz a ligação entre a interface e a camada de persistência ou qualquer outro recurso necessário para processar a requisição HTTP e gerar um pagina da Web.

- **CKEditor**

O *CKEditor* é um editor de textos para ser usado dentro de páginas da web, trazendo para a web características comuns de edição encontradas em aplicações desktop, como o *Microsoft Word* e *OpenOffice*, (CKEDITOR, 2012).

O componente será utilizado na inserção dos dados do evento, agenda do evento e dos dados dos organizadores.

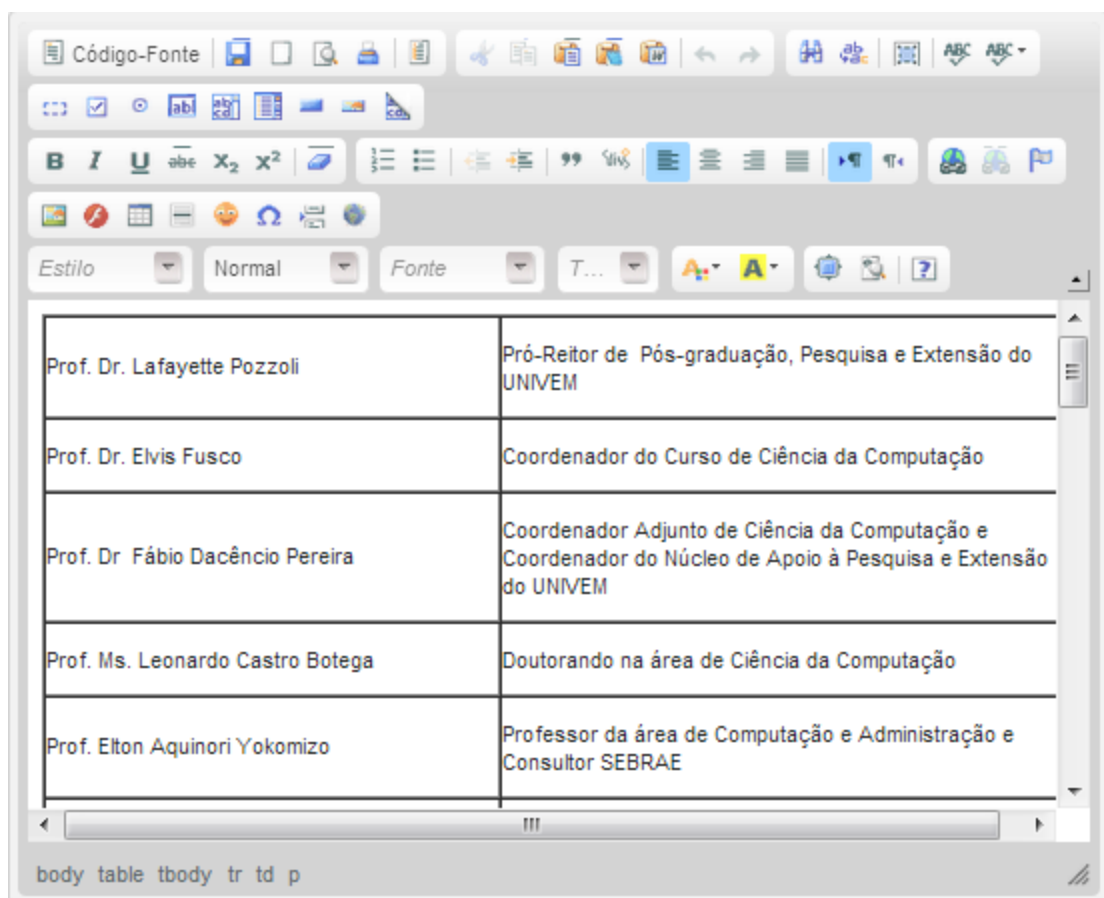
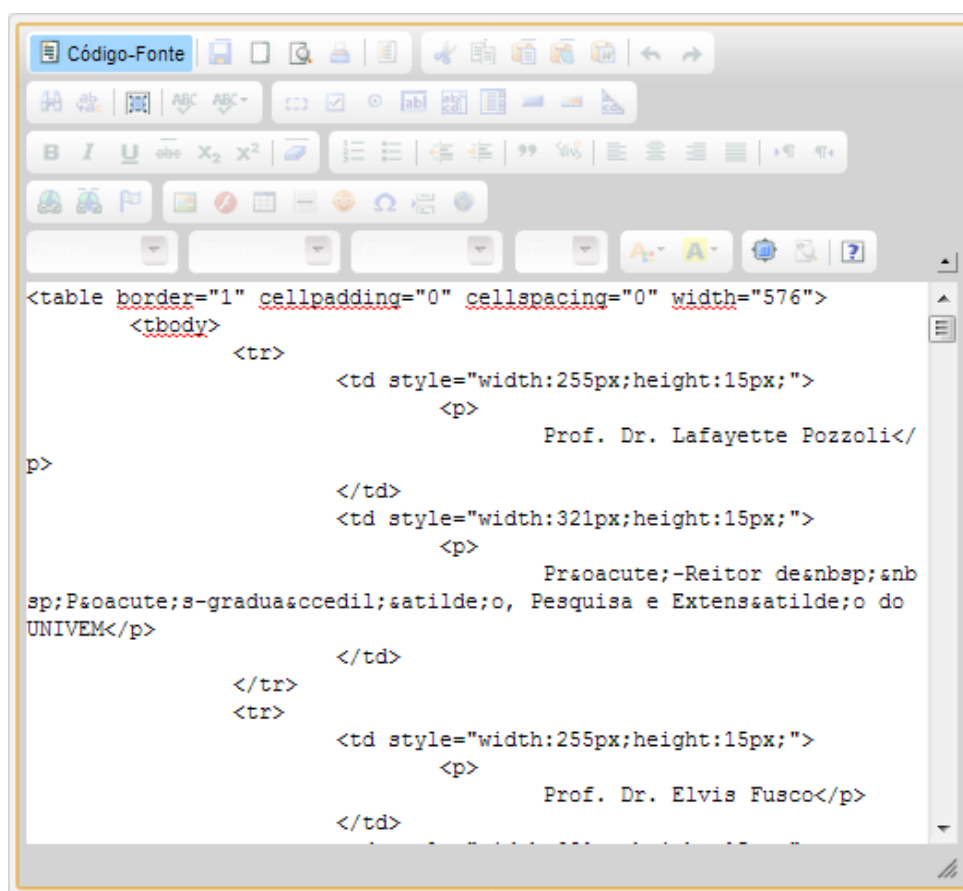


Figura 58 – Editor de Texto para a internet CKEditor..

Fonte: (Próprio Autor, 2012).

Ao mesmo tempo em que o usuário está digitando e formatando o texto, “por traz” ele monta um script HTML com o texto e a formatação correspondente, assim o *script* pode ser salva em uma variável ou em um banco de dados e depois podendo se usar o *script* gerado para criar uma nova pagina ou utilizar o HTML gerado para complementar alguma outra pagina com a informação criada. Tudo transparente ao usuário.



```

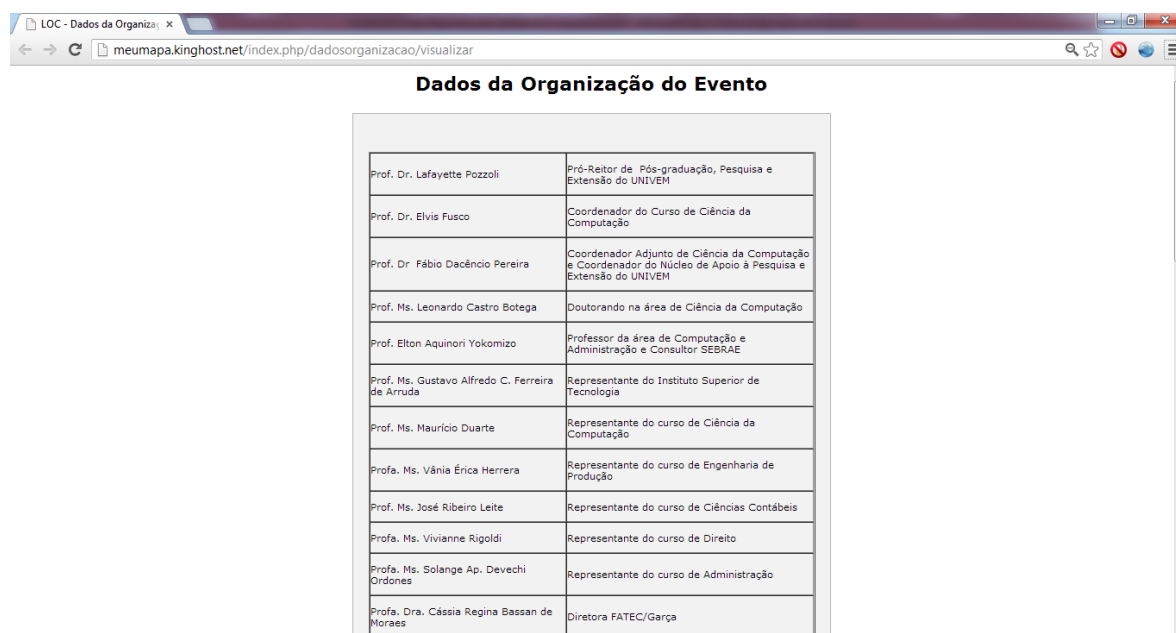
<table border="1" cellpadding="0" cellspacing="0" width="576">
  <tbody>
    <tr>
      <td style="width:255px;height:15px;">
        <p>
          Prof. Dr. Lafayette Pozzoli</p>
      </td>
      <td style="width:321px;height:15px;">
        <p>
          Pr&ocute;-Reitor de&nbsp;sp;P&ocute;s-gradua&cedil;satilde;o, Pesquisa e Extens&atilde;o do UNIVEM</p>
      </td>
    </tr>
    <tr>
      <td style="width:255px;height:15px;">
        <p>
          Prof. Dr. Elvis Fusco</p>
      </td>
    </tr>
  </tbody>
</table>

```

Figura 59 – Script HTML gerado dos textos criados pelo usuário.

Fonte: (Próprio Autor, 2012).

A figura 60 mostra a pagina gerada após a inserção dos dados no editor:



The screenshot shows a web browser window with the address bar displaying 'meumapa.kinghost.net/index.php/dadosorganizacao/visualizar'. The main content area is titled 'Dados da Organização do Evento' and contains a table with the following data:

Prof. Dr. Lafayette Pozzoli	Pró-Reitor de Pós-graduação, Pesquisa e Extensão do UNIVEM
Prof. Dr. Elvis Fusco	Coordenador do Curso de Ciência da Computação
Prof. Dr. Fábio Dacêncio Pereira	Coordenador Adjunto de Ciência da Computação e Coordenador do Núcleo de Apoio à Pesquisa e Extensão do UNIVEM
Prof. Ms. Leonardo Castro Botega	Doutorando na área de Ciência da Computação
Prof. Elton Aquinori Yokomizo	Professor da área de Computação e Administração e Consultor SEBRAE
Prof. Ms. Gustavo Alfredo C. Ferreira de Arruda	Representante do Instituto Superior de Tecnologia
Prof. Ms. Maurício Duarte	Representante do curso de Ciência da Computação
Profa. Ms. Vânia Érica Herrera	Representante do curso de Engenharia de Produção
Prof. Ms. José Ribeiro Leite	Representante do curso de Ciências Contábeis
Profa. Ms. Vivianne Rigoldi	Representante do curso de Direito
Profa. Ms. Solange Ap. Devechi Ordones	Representante do curso de Administração
Profa. Dra. Cássia Regina Bassan de Moraes	Diretora FATEC/Garça

Figura 60 – Pagina php gerada com os dados inseridos no CKeditor buscadas no MYSQL.

Fonte: (Próprio Autor, 2012).

▪ MYSQL

O MySQL é um sistema gerenciador de banco de dados que tornou-se o mais popular do mundo com código aberto por causa de seu alto desempenho, alta confiabilidade e facilidade de uso. Hoje pertencente a *ORACLE* é também o banco de dados de escolha para uma nova geração de aplicativos criados para a *LAMP (Linux, Apache, MySQL, PHP / Perl / Python)*. Muitas das organizações maiores e de mais rápido crescimento do mundo, incluindo *Facebook, Google, Adobe, Alcatel Lucent e Zappos* contam com o MySQL para economizar tempo e dinheiro com seus *Websites* de alto volume, sistemas e pacotes de software. É executado em mais de 20 plataformas, incluindo *Linux, Windows, Mac OS, Solaris, IBM AIX*. É oferecida uma gama abrangente de ferramentas de banco de dados, serviços de treinamento, suporte e consultoria. A versão recente do MySQL é a 5.5.28

5.6. Integração Aplicação Android® e Aplicação Web

5.6.1. Integração pelo banco de dados online - Mysql

Integração do banco de dados online entre ambas as aplicações, em que o Sistema Web é onde são feitos os cadastros dos dados do evento e das localizações por categoria com a geolocalização correspondente inserindo os valores de latitude e a longitude. O Sistema Android® o ELoc buscará a informação da localização solicitada pelo usuário conectando-se ao banco de dados online. Para que isso integração da biblioteca *MySQL Connector/J* ao projeto que está disponível em <http://dev.mysql.com/downloads/connector/j/>. A partir daí a aplicação estará apta a realizar as consultas e buscar a informação.



Figura 61 – Tela de locais úteis é integrado com a aplicação web.

Fonte: (Próprio Autor, 2012).

5.6.2. Integração entre as páginas php geradas pelo Sistema Web - php

O Software desenvolvido em *Android*® o ELoc buscará as páginas php geradas pelo Sistema Web e integra-las as 3 abas referentes ao evento, tais como:

- Evento:
<http://meumapa.kinghost.net/index.php/evento/visualizar>
- Agenda do Evento:
<http://meumapa.kinghost.net/index.php/agenda/visualizar>
- Organizadores:

<http://meumapa.kinghost.net/index.php/dadosorganizacao/visualizar>

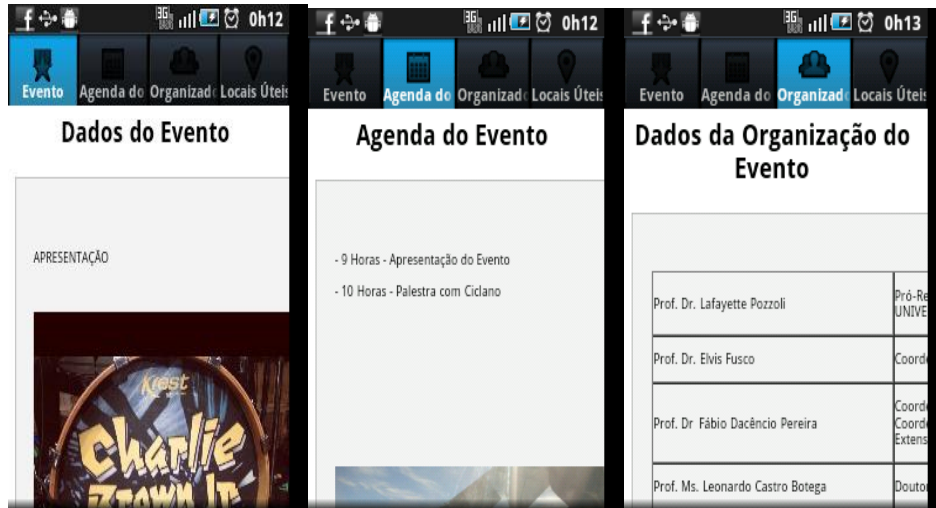


Figura 62 – Telas do Sistema Android® das abas Evento, Agenda do Evento e Organizadores.

Fonte: (Próprio Autor, 2012).

5.7. Comparação de aplicativos da mesma categoria no Google Play

5.7.1. Eventbrite

Segundo o desenvolvedor dá a qualquer pessoa a possibilidade de criar, promover e vender bilhetes para todos os eventos possíveis e imagináveis; também ajuda a descobrir e partilhar eventos que respondem às suas paixões. Quer seja uma festa de bairro ou um concerto esgotado num estádio. Oferece também a busca de eventos locais que ocorram no dia, e a opção de compartilhamento do evento com os amigos, obtenção de bilhetes e suporte a vários idiomas inclusive o português brasileiro.

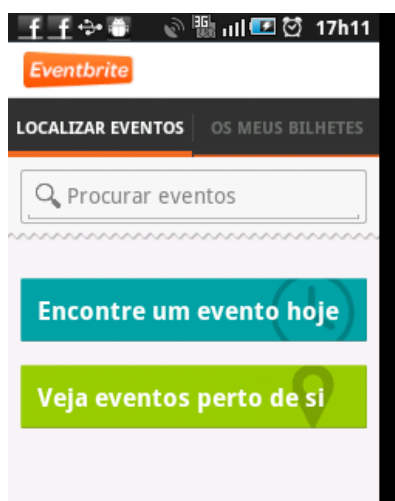


Figura 63 –Tela inicial do sistema Eventbrite.

Fonte: (Próprio Autor, 2012).

Como ao clicar em ENCONTRE UM EVENTO HOJE e VEJA EVENTOS PERTO DE SI não trouxe nada, talvez por se basear na localidade digitada e na data específica:



Figura 64 – Busca de eventos.

Fonte: (Próprio Autor, 2012).

Então se digitou na busca Procurar Eventos no nome da cidade de São Paulo:

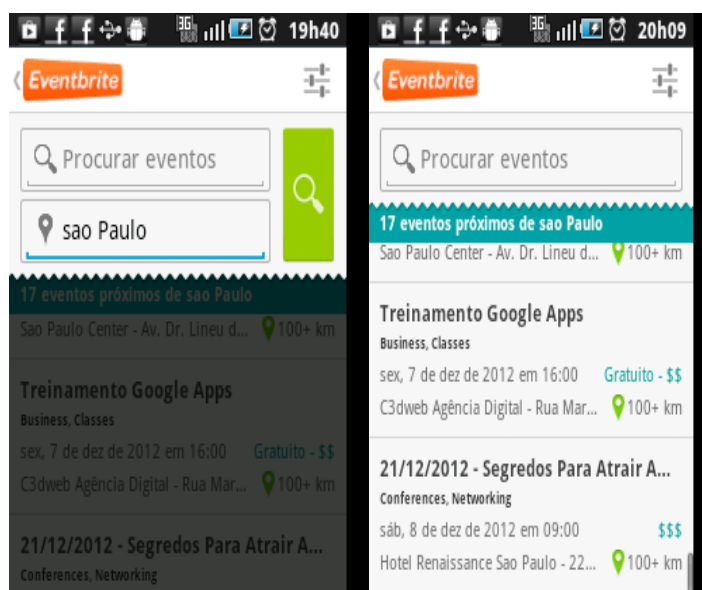


Figura 65 – Na busca efetuada foram listados 17 eventos.

Fonte: (Próprio Autor, 2012).

Ao clicar no evento *Treinamento Google Apps* aparece a tela com os dados do evento e o mapa onde se localiza o evento traçando a rota de como chegar e posteriormente ao clicar em no botão *regista*, dai o sistema encaminha a uma pagina com a cobrança da entrada do evento.

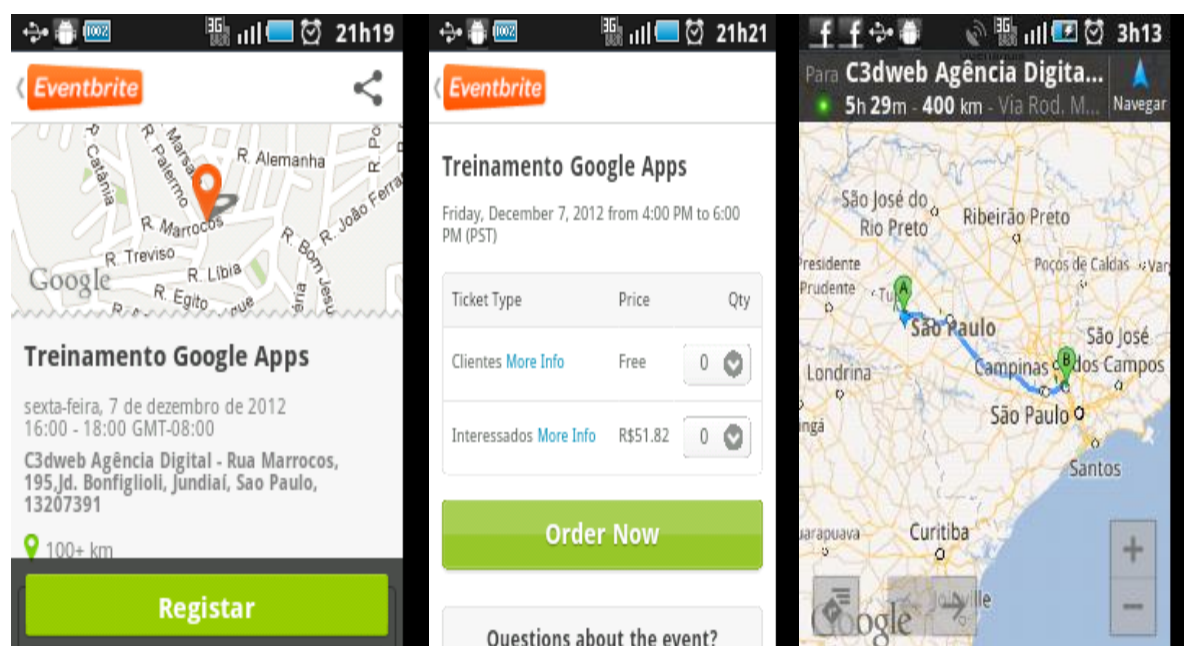


Figura 66 – Imagens da descrição do evento e da cobrança da entrada.

Fonte: (Próprio Autor, 2012).

O Aplicativo para *Android*® pode ser encontrado no link https://play.google.com/store/apps/details?id=com.eventbrite.attendee&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS55dmVudGJyaXRILmF0dGVuZGVVIII0. O sistema funciona para dispositivos *Android*® 2.2 ou superior.

5.7.2. WhatsOn

Segundo o desenvolvedor o sistema tem o objetivo de encontrar os eventos mais recentes, e mostrar todos os seus detalhes e obter a sua localização.

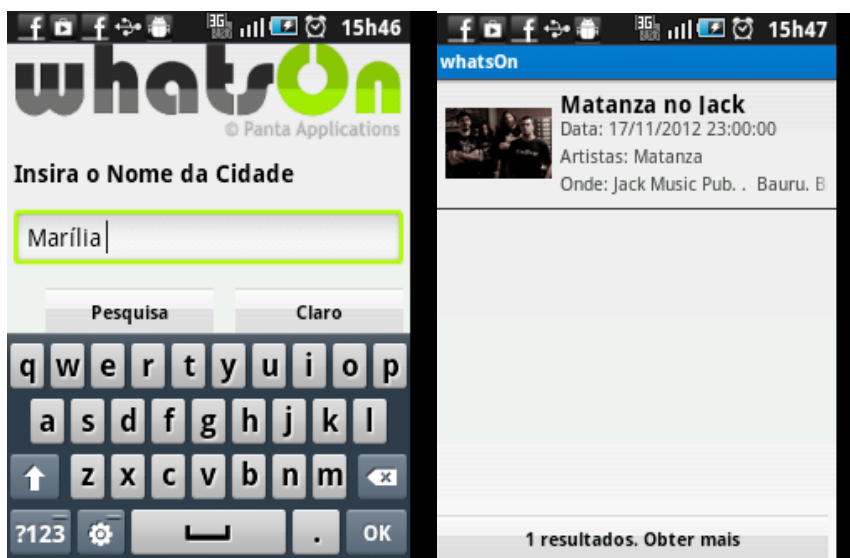


Figura 67 – WhatsOn - Tela Inicial de pesquisa de localidade.

Fonte: (Próprio Autor, 2012).

O aplicativo possui uma busca por localidade onde se quer saber se há eventos, caso não haja na localidade ele buscará na localidade mais perto e mostrará os resultados.

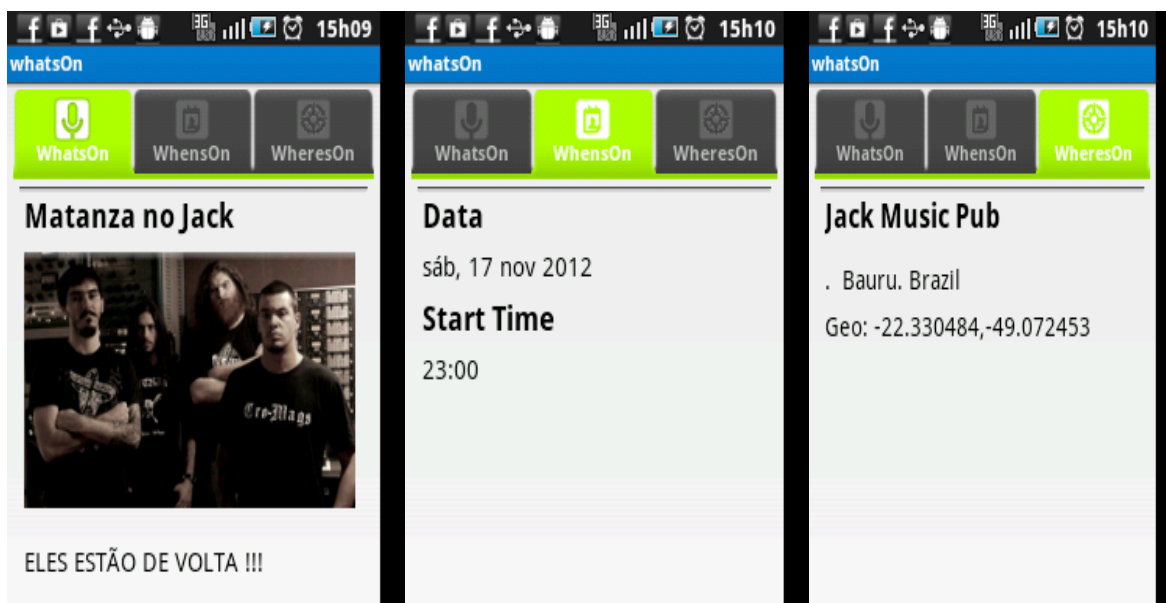


Figura 68 – WhatsOn Abas sobre o Evento.

Fonte: (Próprio Autor, 2012).

As abas *WhatsOn* que mostra os dados do evento, *WhensOn* que descreve quando é o evento e a *WheresOn* que mostra o local do evento apenas com os valores com a latitude e a longitude do evento.

No botão de configurações do aparelho existe uma funcionalidade chamada Local do Evento que deveria trazer a localidade do evento, mas não funcionou nos testes efetuados no *Android*® versão 2.2 *Froyo*. O sistema funciona para dispositivos *Android*® 2.2 ou superior.

5.7.3. Comparações

Os aplicativos *Evenbrite* e *WhatsOn* são boas opções para quem quer saber quais eventos estão acontecendo em determinada localidade, caso haja eventos os dois aplicativos farão uma listagem de todos, podendo os eventos ser de diversos tipos. No *WhatsOn* o serviço referente a mapas para trazer a localização do evento não funcionou, mas é intuitivo no manuseio das informações do evento. O *Eloc* foi desenvolvido para um evento especificamente, o serviço traz o local do evento e mais um conjunto de opções de locais úteis e serviço de GPS para trazer o local atual e logradouro da localidade onde o usuário foi localizado. O *Evenbrite* bem completo, pois além de trazer os dados referentes ao evento e um mapa do local com a opção de traçar a trajetória de “como chegar”, ele oferece opção para comprar bilhetes ou ingressos do evento.

6. RESULTADOS ALCANÇADOS

6.1. Cenário de Testes

6.1.1. Sobre o Evento

FLISOL 2012

O evento de testes será o FLISOL que é Festival Latino-americano de Instalação de Software Livre que é um evento organizado pelos professores e coordenadores dos cursos de Ciência da Computação e Sistemas de Informação do Centro Universitário Eurípides de Marília o UNIVEM e evento será realizado na própria instituição. O objetivo do evento é promover o uso de software livre, apresentando sua filosofia, seu alcance, avanços e desenvolvimento ao público em geral.



Figura 69 – Logotipo do Evento FLISOL.

Fonte: Evento Flisol - Marília, 2012.

Com esta finalidade, diversas comunidades locais de software livre (em cada país, em cada cidade/localidade), organizam simultaneamente eventos em que se instala gratuitamente e totalmente legal, software livre nos computadores levados pelos participantes. Também, paralelamente, são oferecidas apresentações, palestras e oficinas, sobre temas locais, nacionais e latino-americanos sobre Software Livre, com toda sua variedade de expressões: artística, acadêmica, empresarial e social.

6.1.2. Programação Local do Evento

O horário do evento em Marília será das 09:00 às 13:00

Palestra: "Nem só de *nerds* e *geeks* vive o Software-Livre"

Palestrante: Danilo Magrini

Local: Auditório Sunji Nishimura

Horário: 9h00

Resumo: "Ao contrário do que se imagina, não é somente codificando que se contribui com projetos de código-aberto. O objetivo dessa palestra é identificar as diversas formas de colaborar com projetos livres, bem como identificar as vantagens que tal ideal proporciona".

Install Fest: Instalação e Configuração de Softwares Livres (Sistemas Operacionais, Plataformas de Desenvolvimento, Aplicativos em geral)

Local: Computing and Information Systems Research Lab (COMPSI-UNIVEM)

Horário: 10h00 às 13h00

Oficina: Introdução ao Moodle (plataforma de EAD)

Instrutor: Ivan Sakai

Local: Computing and Information Systems Research Lab (COMPSI-UNIVEM)

Horário: 10h00

Oficina: Instalação e Configuração e Personalização da ferramenta Wordpress

Instrutor: Claudio Rosse

Local: Computing and Information Systems Research Lab (COMPSI-UNIVEM)

Horário: 10h00

Minicurso: Introdução ao JQuery

Instrutor: Rafael Santana

Local: Laboratório 6

Horário: 10h00

Minicurso: Criação ágil de aplicativos WEB utilizando tecnologias Java

Instrutor: Saulo B. Tupinambá

Local: Laboratório 1

Horário: 10h00

6.1.3. Organizadores do Evento

Coordenador Geral do Evento

Prof. Leonardo Botega - botega@univem.edu.br

Coordenadores

Prof. Elvis Fusco - fusco@univem.edu.br

Prof. Rodolfo Barros Chiamonte - chiamonte@univem.edu.br

Prof. Fábio Dacêncio - dacencio@univem.edu.br

Realização: *Computing and Information Systems Research Lab – COMPSI/Univem*

6.1.4. Parceiros “Fictícios” do Evento e locais úteis

O por se tratar de um evento aberto e conseqüentemente divulgado em varias cidades, a comissão organizadora se preocupou em dar estrutura aos participantes provenientes de outras cidades e fez parcerias com os principais hotéis e restaurantes da cidade, além de fornecer informações sobre o serviço de transporte como a localização dos terminais metropolitanos, rodoviários e aeroportos da cidade, e também outros serviços como bancos e hospitais.

Local do Evento

- Univem
 - Latitude: -22.231935°
 - Longitude: -49.967902°

Hotéis

- Hotel 1
 - Latitude: -22.222267°
 - Longitude: -49.948110°
- Hotel 2
 - Latitude: -22.218314°
 - Longitude: -49.946763°
- Hotel 3
 - Latitude: -22.222219°
 - Longitude: -49.948501°
- Hotel 4
 - Latitude: -22.227461°
 - Longitude: -49.958788°
- Hotel 5
 - Latitude: -22.236126°
 - Longitude: -49.974615°
- Hotel 6
 - Latitude: -22.230937°
 - Longitude: -49.968559°

Restaurantes

- Restaurante 1
 - Latitude: -22.229241°
 - Longitude: -49.931198°
- Restaurante 2
 - Latitude: -22.191924°
 - Longitude: -49.959662°

- Restaurante 3
 - Latitude: -22.210462°
 - Longitude: -49.953209°
- Restaurante 4
 - Latitude: -22.232742°
 - Longitude: -49.969709°
- Restaurante 5
 - Latitude: -22.236312°
 - Longitude: -49.969433°

Bancos

- Banco 1
 - Latitude: -22.223285°
 - Longitude: -49.957237°

 - Latitude: -22.218254°
 - Longitude: -49.948464°

 - Latitude: -22.215711°
 - Longitude: -49.949051°

 - Latitude: -22.204817°
 - Longitude: -49.955496°
- Banco 2
 - Latitude: -22.232883°
 - Longitude: -49.965957°

 - Latitude: -22.232564°
 - Longitude: -49.968537°

- Latitude: -22.221406°
 - Longitude: -49.951132°

 - Latitude: -22.218062°
 - Longitude: -49.948509°

 - Latitude: -22.217672°
 - Longitude: -49.948584°

 - Latitude: -22.215069°
 - Longitude: -49.949386°

- Banco 3
 - Latitude: -22.222357°
 - Longitude: -49.954555°

 - Latitude: -22.216423°
 - Longitude: -49.948855°

 - Latitude: -22.216095°
 - Longitude: -49.948944°

- Banco 4
 - Latitude: -22.216827°
 - Longitude: -49.948973°

 - Latitude: -22.224073°
 - Longitude: -49.958884°

- Banco 5

- Latitude: -22.218903°
- Longitude: -49.948308°

- Latitude: -22.213326°
- Longitude: -49.948925°

- Banco 6
 - Latitude: -22.217811°
 - Longitude: -49.948775°

Hospitais

- Hospital 1
 - Latitude: -22.238074°
 - Longitude: -49.966346°

- Hospital 2
 - Latitude: -22.213135°
 - Longitude: -49.950309°

- Hospital 3
 - Latitude: -22.226921°
 - Longitude: -49.935605°

- Hospital 4
 - Latitude: -22.212691°
 - Longitude: -49.939156°

Terminal Metropolitano de Ônibus

- Latitude: -22.215804°
- Longitude: -49.9467288°

Terminal Rodoviário

- Latitude: -22.230409°
- Longitude: -49.923213°

Aeroporto de Marília

- Latitude: -22.196991°
- Longitude: -49.929798°

6.2. Usabilidade

6.2.1. Sistema Web

▪ Login

Depois de coletas as informações referentes ao evento, as mesmas devem ser cadastradas no site www.meumapa.kinghost.net, onde o usuário devera inserir o nome de usuário e senha pois o acesso é restrito, a senhas são criptografadas para aumentar a segurança.

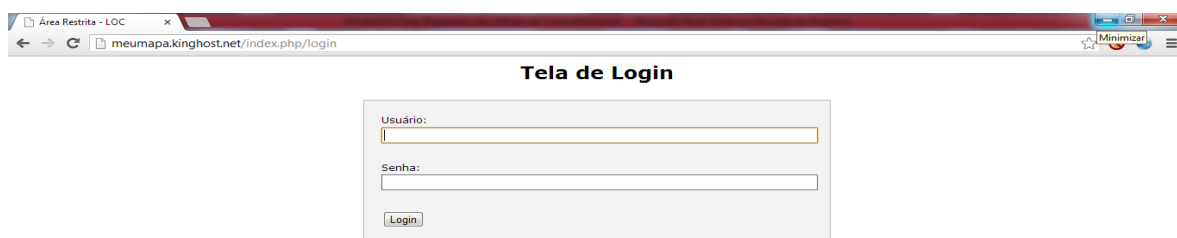


Figura 70 – Acessar o Sistema online para a inserção das informações.

Fonte: (Próprio Autor, 2012).

▪ Menu Principal

Ao acessar o sistema o usuário será levado ao Menu principal, com várias opções, Primeiramente serão inseridos os dados do evento nas opções Evento, Agenda do Evento e Dados dos Organizadores, em todos os cadastros há a opção de inserir uma imagem. No canto superior direito é mostrado o nome usuário com opção de *logout*.



Figura 71 – Menu principal do Sistema Web.

Fonte: (Próprio Autor, 2012).

6.2.2. Cadastros do Sistema Web

▪ Cadastro do Evento

Ao clicar no botão Evento poderá inserir os dados sobre o evento a ser realizado, com a opção para inserção de uma imagem.



Figura 72 - Inserindo os dados do evento (esquerda), ao clicar em visualizar pagina será possível ver o resultado (direita).

Fonte: (Próprio Autor, 2012).

▪ Cadastro da Agenda do Evento

Ao clicar no botão Agenda do Evento será possível inserir a agenda como dadas de horários relacionados ao evento, com opção para inserção de uma imagem:

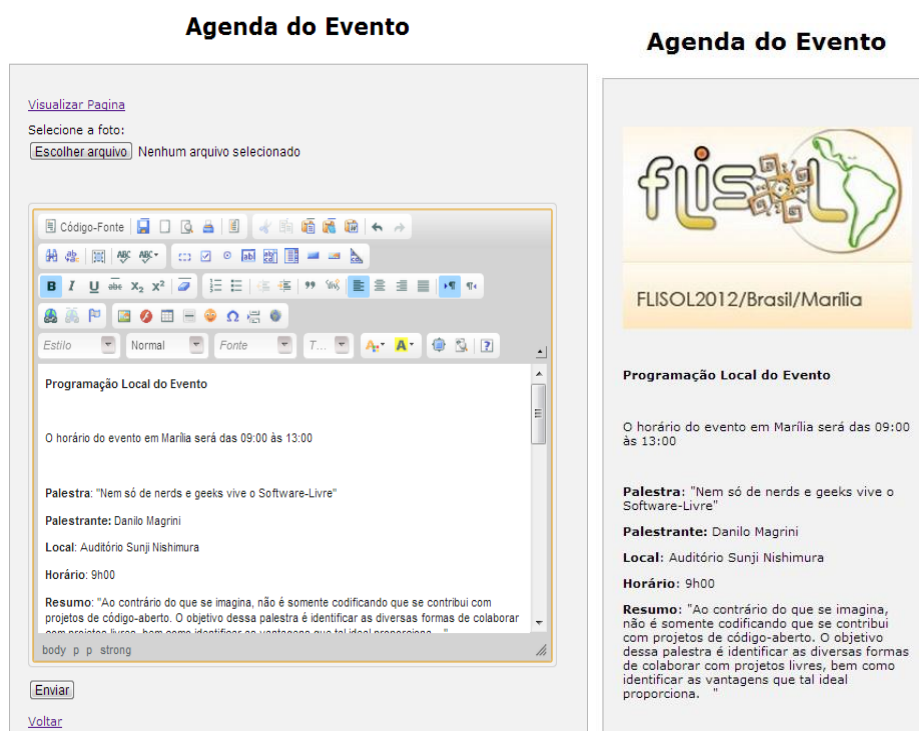


Figura 73 - Inserindo a agenda do evento (esquerda), ao clicar em visualizar pagina será possível ver o resultado (direita).

Fonte: (Próprio Autor, 2012).

▪ Cadastro dos Dados da Organização do Evento

Ao clicar no botão Organizadores será possível inserir informações referentes aos organizadores do evento, com opção para inserção de uma imagem:



Figura 74 - Inserindo a agenda do evento (esquerda), ao clicar em visualizar página será possível ver o resultado (direita).

Fonte: (Próprio Autor, 2012).

▪ Cadastro de Localizações

Após a inserção dos dados do evento, o usuário poderá inserir as localizações clicando em Localizações.

O sistema possui 8 tipos de localização pré-definidos em que o ELoc poderá mostrar as localizações em Locais Úteis: Evento, Hotéis, Restaurantes, Bancos, Hospital, Terminal Urbano, Terminal Rodoviário e Aeroporto.

Cadastro de Localizações



[Criar Tipo de Localizacao](#)

Código	Descrição do Tipo	Ações
1	Evento	Editar Deletar
2	Hotel	Editar Deletar
3	Restaurante	Editar Deletar
4	Hospital	Editar Deletar
5	Banco	Editar Deletar
6	Rodoviaria	Editar Deletar
7	Terminal Urbano	Editar Deletar
8	Aeroporto	Editar Deletar

[Voltar](#)

Figura 75 – Tipo de localizações disponíveis no sistema.

Fonte: (Próprio Autor, 2012).

Em cadastro de localizações é onde poderão se incluir as localizações e vincular um tipo em clicando em “Criar Localização”.

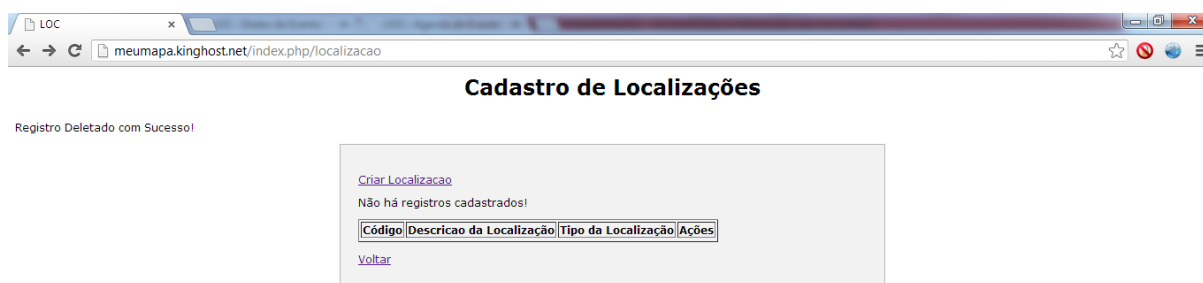
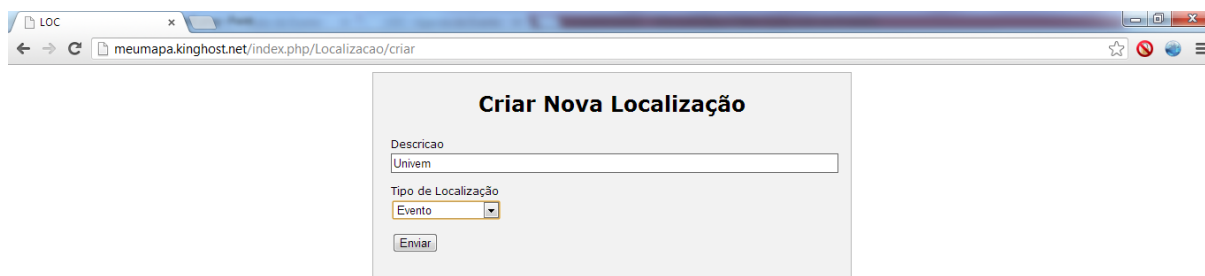


Figura 76 – Cadastro de Localizações.

Fonte: (Próprio Autor, 2012).

Será cadastrada a localização do evento principal o Univem que será do tipo Evento.



The image shows a web browser window with the address bar displaying "meumapa.kinghost.net/index.php/Localizacao/criar". The main content area contains a form titled "Criar Nova Localização". The form has two input fields: "Descricao" with the value "Univem" and "Tipo de Localização" with a dropdown menu set to "Evento". Below the fields is an "Enviar" button.

Criar Nova Localização	
Descricao	<input type="text" value="Univem"/>
Tipo de Localização	<input type="text" value="Evento"/>
<input type="button" value="Enviar"/>	

Figura 77 - Criando uma nova localização.

Fonte: (Próprio Autor, 2012).

Após criada a localização ela será listada na *grid* de Localizações, Na Grid em “Ações” poderá se Editar, Deletar ou Gerar uma geolocalização para que uma ou mais localizações apareçam no mapa do sistema *Android*® o que será feito no próximo passo.

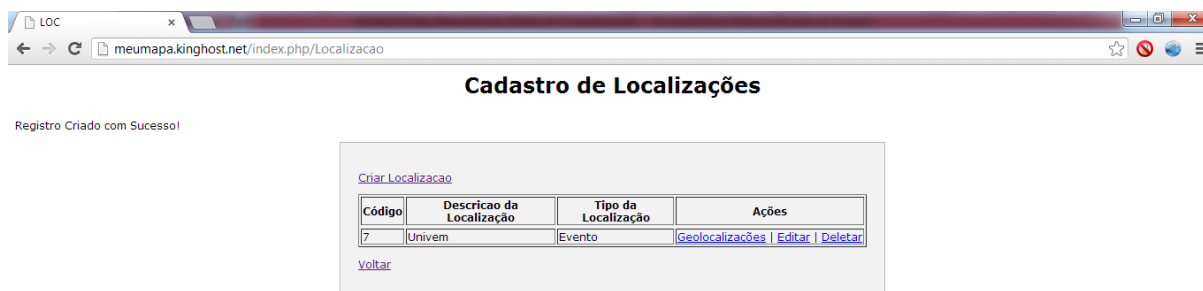


Figura 78 – Grid de localizações com a localização criada.

Fonte: (Próprio Autor, 2012).

▪ Cadastrando a Geolocalização

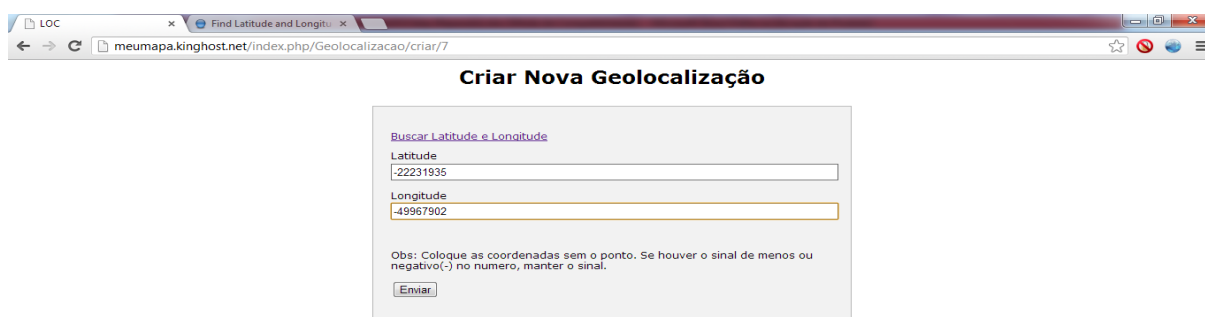
Ao clicar no em Geolocalizações, será mostrada a grid de todas as geolocalizações criadas para a localização, caso não haja alguma, há a opção de criação em “Criar Localização”.



Figura 79 – Grid de Geolocalizações da localização Univem.

Fonte: (Próprio Autor, 2012).

Ao clicar em “Criar Geolocalização” o usuário irá inserir a latitude e a longitude da localização, caso o usuário não saiba a geolocalização ele poderá fazer a pesquisa clicando em Buscar latitude e longitude onde será aberta uma nova página de navegador redirecionando o usuário ao domínio <http://www.findlatitudeandlongitude.com>, onde será possível buscar a latitude e a longitude de qualquer lugar do mundo. Ao inserir os dados de com a latitude e a longitude haverá as recomendações de como inserir os dados corretamente.



The image shows a web browser window with the address bar displaying `meumapa.kinghost.net/index.php/Geolocalizacao/criar/7`. The page title is "Criar Nova Geolocalização". Below the title is a form titled "Buscar Latitude e Longitude". The form contains two input fields: "Latitude" with the value "-22231935" and "Longitude" with the value "-49967902". Below the input fields is a note: "Obs: Coloque as coordenadas sem o ponto. Se houver o sinal de menos ou negativo(-) no numero, manter o sinal." and an "Enviar" button.

Figura 80 – Criação da Geolocalização inserindo a latitude e a longitude.

Fonte: (Próprio Autor, 2012).

No mapa abaixo a circunferência em azul representa o ponto onde se quer saber a Latitude e Longitude, a informação é mostrada nos quadros.

The screenshot shows a web browser window with the URL www.findlatitudeandlongitude.com. The page title is "Find Latitude and Longitude". The main content is a satellite map of a city with a blue circle highlighting a specific location. Below the map, there are several data boxes:

- Load Location:** Fields for Latitude, Longitude, Location, and Post Code.
- Map Coordinates of Selected Location:**
 - Latitude: $9^{\circ}22'13''S$ (9.37028°)
 - Longitude: $48^{\circ}58'2.6423''W$ (-48.96981°)
 - Location: Miraflores, Miraflores - São Paulo, 11265-000, República Federativa do Brasil
 - Latitude: -22.371917
 - Longitude: -48.969817
- Selected Location (Approximate):**
 - Address: Miraflores, Miraflores - São Paulo, 11265-000, República Federativa do Brasil
 - Latitude: -22.371917
 - Longitude: -48.969817
- Map Coordinates of Mouse:**
 - Latitude: $9^{\circ}22'13''S$ (9.37028°)
 - Longitude: $48^{\circ}58'2.6423''W$ (-48.96981°)
 - Latitude: -22.371917
 - Longitude: -48.969817
- Map Parameters:**
 - Zoom: 19
 - Type: Satellite
 - Ctrl: 180° 60° pixels
 - Reset map: [button]
 - Current link: [button]
- Menu:** Home, Login to address, address to Latitude/Longitude, Batch Geocode, Batch Reverse Geocode, Location Search, Reverse Geocodes, Antipode Map (Turning 180°), GP's Coordinates Converter, Facebook, Record Lat-Longs.
- Contact information:** Developer: Carlos B. Zwiets/hofer, Email: weomaster@gmail.com
- VVCC Valid:** W3C XHTML 1.0, W3C CSS 2.1

Figura 81 – Site Find Latitude and Longitude que auxiliara na busca da geolocalização desejada.

Fonte: (Próprio Autor, 2012).

Após inserida a geolocalização o mapa do software *Android*® já poderá listar no mapa a localização recém, criada. O usuário poderá inserir mais de uma geolocalização para a localização, por exemplo na cidade podem haver mais de dois Bancos da mesma instituição mas em locais diferentes.

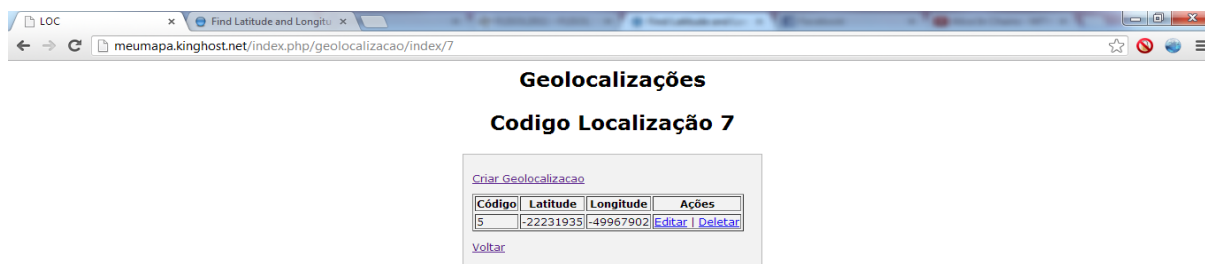


Figura 82 – Grid das geolocalizações cadastradas.

Fonte: (Próprio Autor, 2012).

O procedimento para inserir as localizações é o mesmo para todos os tipos de localizações. Após inserir os todos dados do evento e todas as localizações com as geolocalizações referentes, o ELoc estará apto para auxiliar os participantes e os organizadores do evento.

Cadastro de Localizações

[Criar Localizacao](#)

Código	Descrição da Localização	Tipo da Localização	Ações
7	Univem	Evento	Geolocalizações Editar Deletar
8	Hotel Tenda	Hotel	Geolocalizações Editar Deletar
9	Hotel Max Plaza Marília	Hotel	Geolocalizações Editar Deletar
10	Estoril Apart Hotel	Hotel	Geolocalizações Editar Deletar
11	Hotel Quality Sun Valley Marília	Hotel	Geolocalizações Editar Deletar
12	Apart Hotel Cavallari	Hotel	Geolocalizações Editar Deletar
13	Hotel Santa Monica	Hotel	Geolocalizações Editar Deletar
14	Churrascaria Kieza	Restaurante	Geolocalizações Editar Deletar
15	Churrascaria Alvorada	Restaurante	Geolocalizações Editar Deletar
16	Maninhos Grill	Restaurante	Geolocalizações Editar Deletar
17	Restaurante Tubarão	Restaurante	Geolocalizações Editar Deletar
18	Restaurante Rancho Brasileiro	Restaurante	Geolocalizações Editar Deletar
19	Banco do Brasil	Banco	Geolocalizações Editar Deletar
20	Banco Santander	Banco	Geolocalizações Editar Deletar
21	Banco Itau	Banco	Geolocalizações Editar Deletar

Figura 83 – Sistema Web com todas as Localizações cadastradas e as geolocalizações inseridas.

Fonte: (Próprio Autor, 2012).

▪ Gestão de Usuários

O sistema Web possui gestão de usuários no qual são cadastrados os usuários no qual poderão ter acesso ao sistema.

▪ Cadastro de Usuários

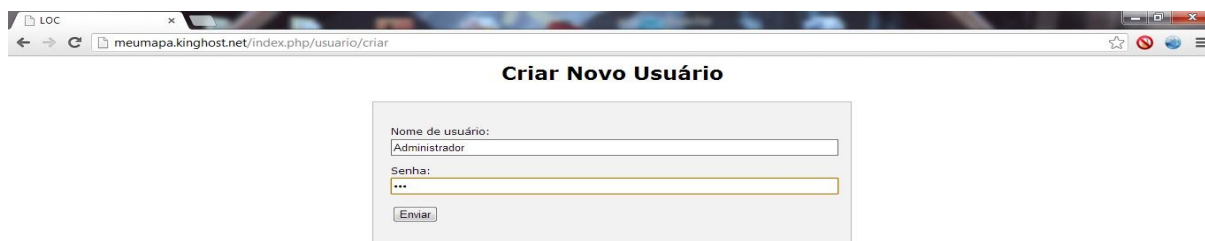
Em gestão de usuários o gestor do sistema poderá inserir os usuários conforme a necessidade.



Figura 84 – Grid do Cadastro de usuários.

Fonte: (Próprio Autor, 2012).

Ao clicar em “Criar Novo Usuário” aparecerão os campos para a inserção do nome do usuário e de uma senha pessoal.



Nome de usuário:
Administrador

Senha:

Enviar

Figura 85 – Inserindo o Usuário e a Senha.

Fonte: (Próprio Autor, 2012).

Ao clicar em enviar, o sistema terá mais um usuário que poderá acessar o sistema *Web*, no caso o Administrador.



Registro Criado com Sucesso!

Criar Novo Usuario

Código	Nome de Usuario	Ações
1	Rodolpho	Editar Deletar
6	Administrador	Editar Deletar

[Voltar](#)

Figura 86 – Grid de usuários com o usuário recém-cadastrado.

Fonte: (Próprio Autor, 2012).

- **Acesso ao site com o usuário criado**

Na tela de *login*, será testado o acesso com o usuário Administrador recém-criado.

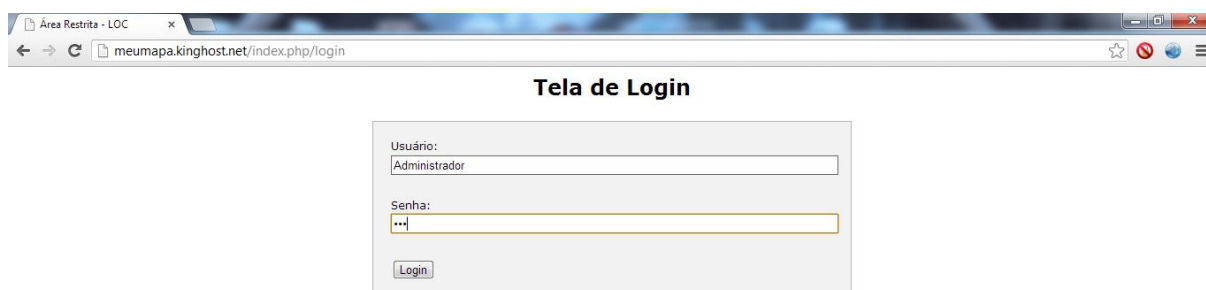


Figura 87 - Acessando o sistema com o usuário criado.

Fonte: (Próprio Autor, 2012).

Ao acessar o sistema, aparecerá no canto superior esquerdo o nome do usuário, no caso o Administrador.



Figura 88 – Menu Principal do Sistema Online.

Fonte: (Próprio Autor, 2012).

6.2.3. Sistema Android® - Eloc

Depois de inseridas as informações sobre o Flisol no Sistema *online*, o Eloc receberá as informações do evento nas abas Evento, Agenda do Evento e Organizadores e das localizações na aba Locais Úteis, ao abrir o Eloc as páginas são carregadas previamente no sistema.



Figura 89 – Carregando as Informações do Sistema Online.

Fonte: (Próprio Autor, 2012).

▪ Aba Evento

Ao selecionar a aba Evento será carregado do Sistema Online a página php referente aos dados do evento, conforme cadastrado posteriormente.

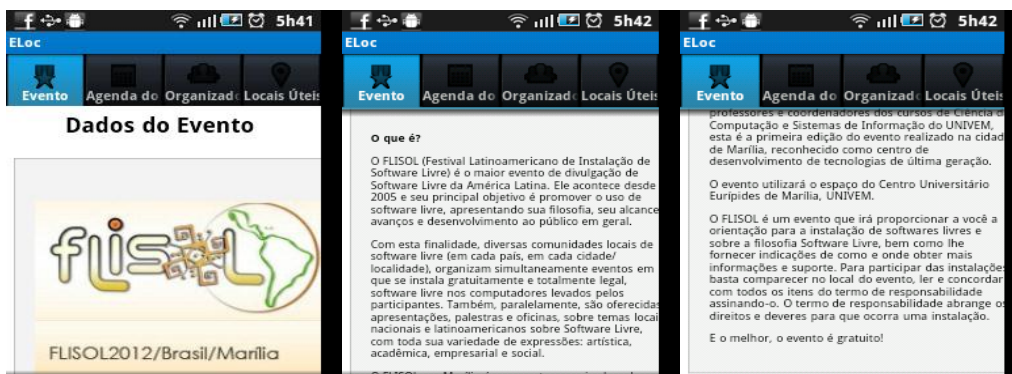


Figura 90 – Aba Evento do Eloc recebendo a página dos dados do evento Flisol cadastrado no sistema online.

Fonte: (Próprio Autor, 2012).

▪ **Aba Agenda do Evento**

Ao Selecionar a aba Agenda do Evento também será carregado do Sistema Online a página php referente a agenda do evento, conforme cadastrado posteriormente .

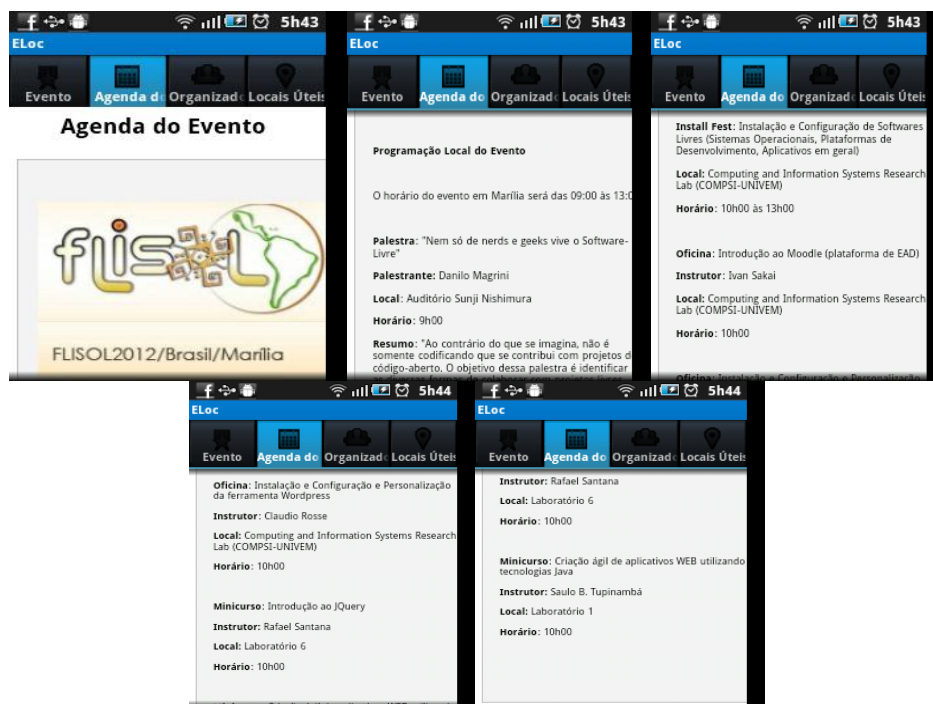


Figura 91 – Aba Agenda do Evento recebendo a página da agenda do evento Flisol cadastrado no sistema.

Fonte: (Próprio Autor, 2012).

▪ Aba Organizadores

Por fim, seguindo o mesmo procedimento das abas anteriores ao selecionar a aba Organizadores será carregado do Sistema Online a página php referente aos dados dos organizadores do evento, conforme cadastrado no Sistema *Online*.

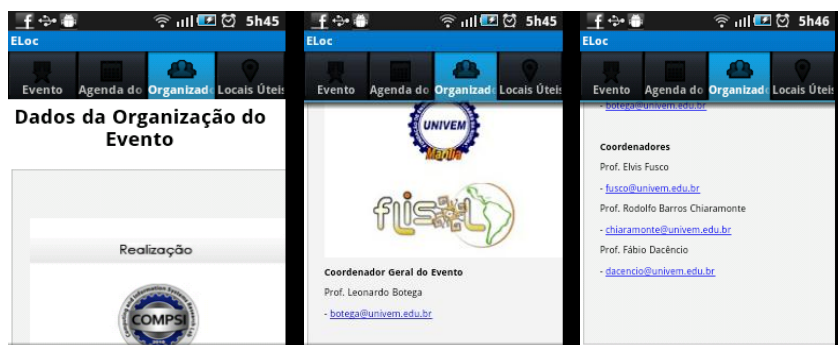


Figura 92 – Aba Organizadores recebendo os dados dos organizadores do evento Flisol cadastrado no sistema online.

Fonte: (Próprio Autor, 2012).

▪ Aba Locais Úteis

Na aba Locais úteis serão disponibilizadas as opções das localizações, onde depois de selecionada a opção serão renderizadas as geolocalizações cadastradas no Sistema *Online* para a cidade de Marília. Os Locais Úteis estão classificados entre Evento, Serviços e Utilidades e Transporte.



Figura 93 – Menu de Locais Úteis.

Fonte: (Próprio Autor, 2012).

- **Evento**
- **Localização do Evento**

A opção Evento trará a geolocalização cadastrada no sistema *online* relacionada o Local do Evento que no caso é o Univem, ao clicar no marcador o endereço é trazido automaticamente pelo serviço do *Android® Maps* pela classe *Geocode*.



Figura 94 – Geolocalização com o marcador mostrando Local do Evento e o endereço.

Fonte: (Próprio Autor, 2012).

- **Serviços e Utilidades**
- **Localização dos Hotéis**

A opção Hotéis trará todas as geolocalizações dos hotéis cadastrados, aparecerão 5 marcadores.

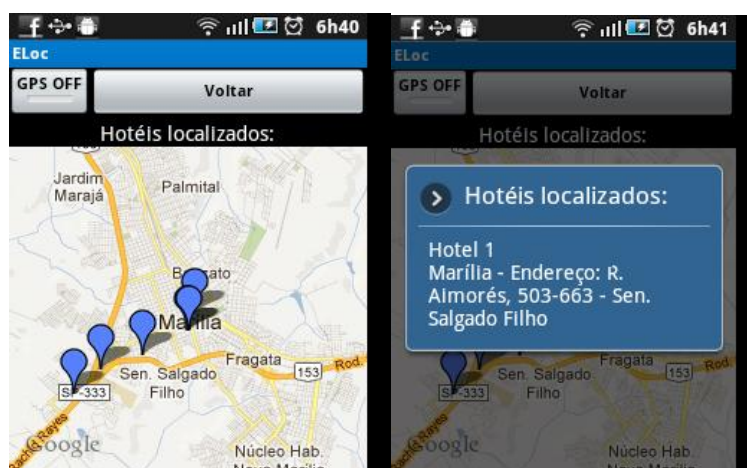


Figura 95 – Geolocalização com os marcadores dos hotéis cadastrados. No para exemplo foi selecionado o hotel 1.

Fonte: (Próprio Autor, 2012).

▪ Localização dos Restaurantes

A opção Restaurante trará todas as geolocalizações dos restaurantes cadastrados, aparecerão 5 marcadores.

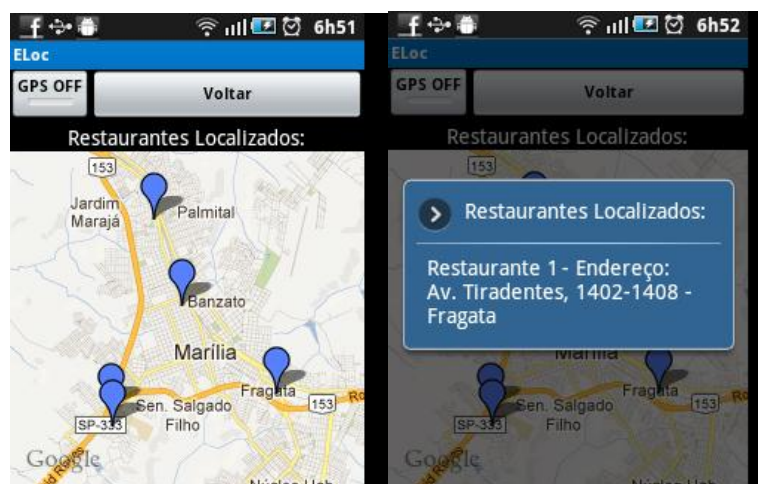


Figura 96 – Geolocalização com os marcadores dos restaurantes cadastrados. No para exemplo foi selecionada o restaurante 1.

Fonte: (Próprio Autor, 2012).

▪ Localização dos Bancos

A opção Bancos trará todas as geolocalizações dos bancos cadastrados.

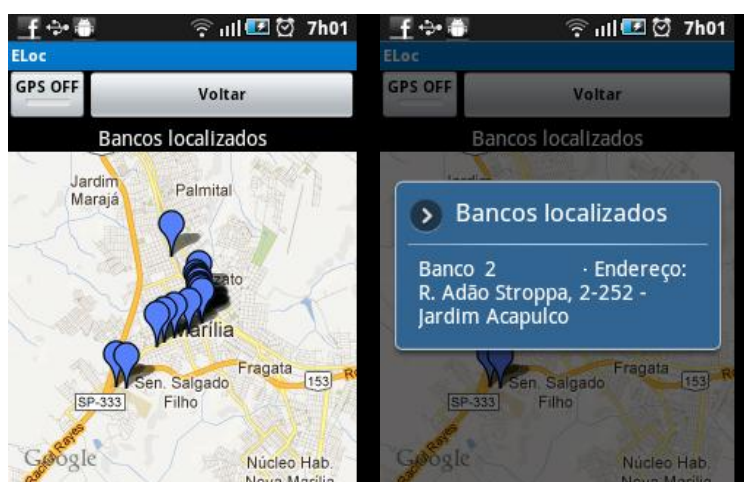


Figura 97 - Geolocalização com os marcadores dos bancos cadastrados. No para exemplo foi selecionado o banco 2.

Fonte: (Próprio Autor, 2012).

▪ Localização dos Hospitais

A opção Hospitais trará todas as geolocalizações dos hospitais cadastrados.

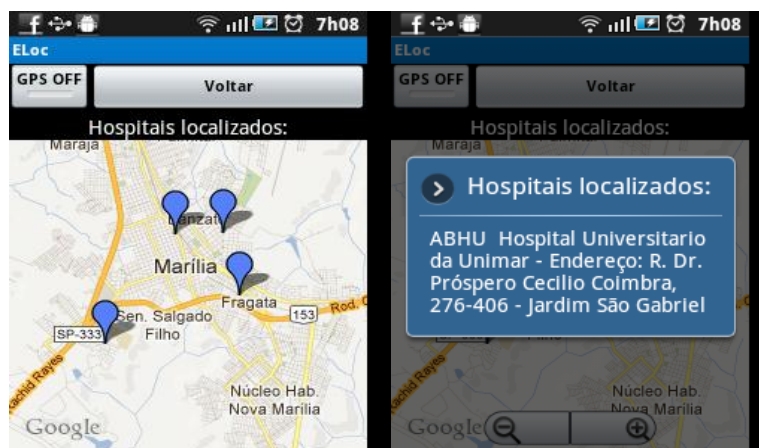


Figura 98 - Geolocalização com os marcadores dos hospitais cadastrados. No para exemplo foi selecionado o hospital.

Fonte: (Próprio Autor, 2012).

- Transporte
- Terminal Urbano

A opção terminal urbano trará a geolocalização do terminal urbano da cidade

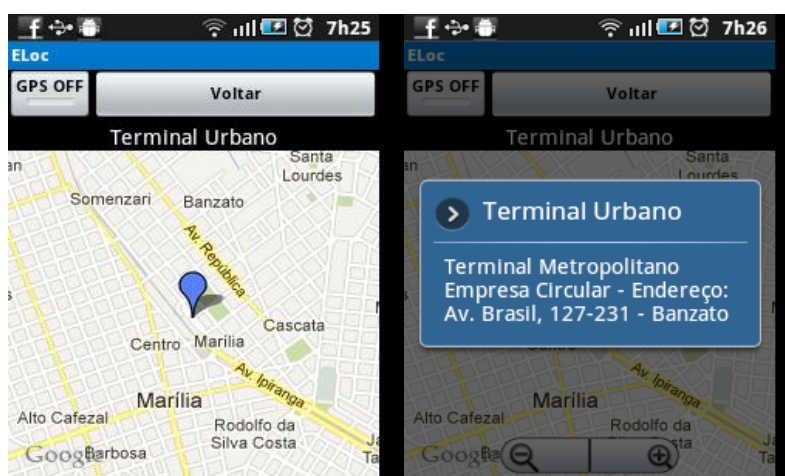


Figura 99 - Geolocalização com os marcadores do terminal urbano cadastrado.

Fonte: (Próprio Autor, 2012).

▪ Terminal Rodoviário

A opção terminal rodoviário trará a geolocalização do terminal rodoviário da cidade.

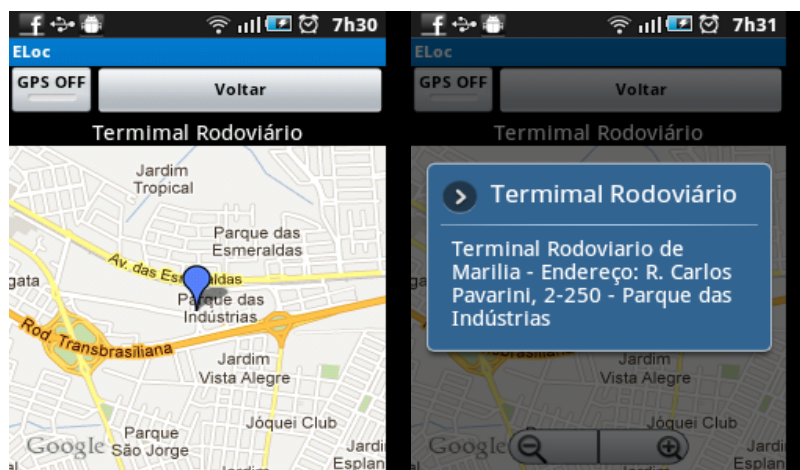


Figura 100 - Geolocalização com o marcador do terminal rodoviário cadastrado.

Fonte: (Próprio Autor, 2012).

▪ Aeroporto

A opção aeroporto trará a geolocalização do aeroporto da cidade



Figura 101 - Geolocalização com o marcador do aeroporto cadastrado.

Fonte: (Próprio Autor, 2012).

- GPS

O Eloc contém uma opção de GPS no qual ajuda ao usuário a encontrar a sua localização atual, sem perder o marcador da localização pesquisada também.

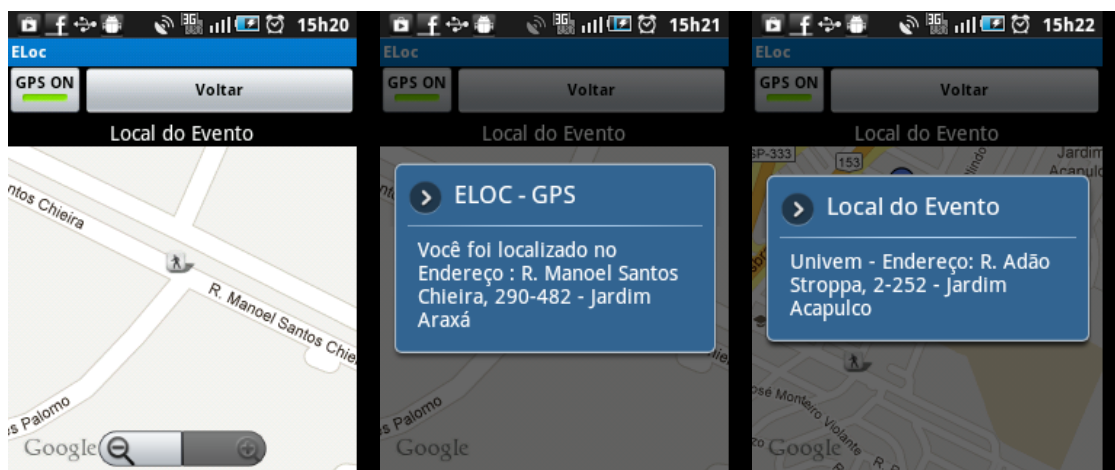


Figura 102 – Mostra o funcionamento do GPS mostrando inclusive o endereço onde se está no momento, sem perder o marcador da busca efetuada.

Fonte: (Próprio Autor, 2012).

▪ Verificações de recursos disponíveis

Caso não haja internet no dispositivo, ou no momento do uso do GPS ele esteja desativado no aparelho ou a localização solicitada para pesquisa não tem geolocalização cadastrada no sistema online o sistema faz a verificação e mostra a mensagem ao usuário em caso de erro.



Figura 103 – Verificações de recursos e dados disponíveis.

Fonte: (Próprio Autor, 2012).

6.3. Publicação no Google Play

Foi criada uma conta no Google Play, sendo que o aplicativo já foi publicado com versões de com correções e se encontra disponível no link: https://play.google.com/store/apps/details?id=br.com.meueventomapa&feature=search_result#?t=W251bGwsMSwxLDEsImJyLmNvbS5tZXVldmVudG9tYXBhI0. O aplicativo Eloc se encontra na versão 1.3. Para demonstração será enviada uma nova versão 1.4



Figura 104 – Pagina do aplicativo desenvolvido o ELoc com opção para download.

Fonte: (Próprio Autor, 2012).

6.3.1. Publicando o aplicativo (.apk)

Para publicar o aplicativo é necessário extrair o arquivo .apk gerado em conjunto com a chave privada do projeto, sempre deveser usada a mesma chave para a compilação de todas as versões posteriores.

Caso seja a primeira versão do sistema na ferramenta *Motodev* ir em no Menu *MOTODEV* → *Extract Android® Aplication*. Caso contrario, será necessário incrementar a tag *Android®:versionCode* no arquivo *manifest* do sistema antes de gerar o novo .apk por esse procedimento. No caso do ELoc será necessário incrementar o valor do *version code*.

Nesse lugar será onde deveser especificar o qual o projeto e a chave que deveser assinar o aplicativo.

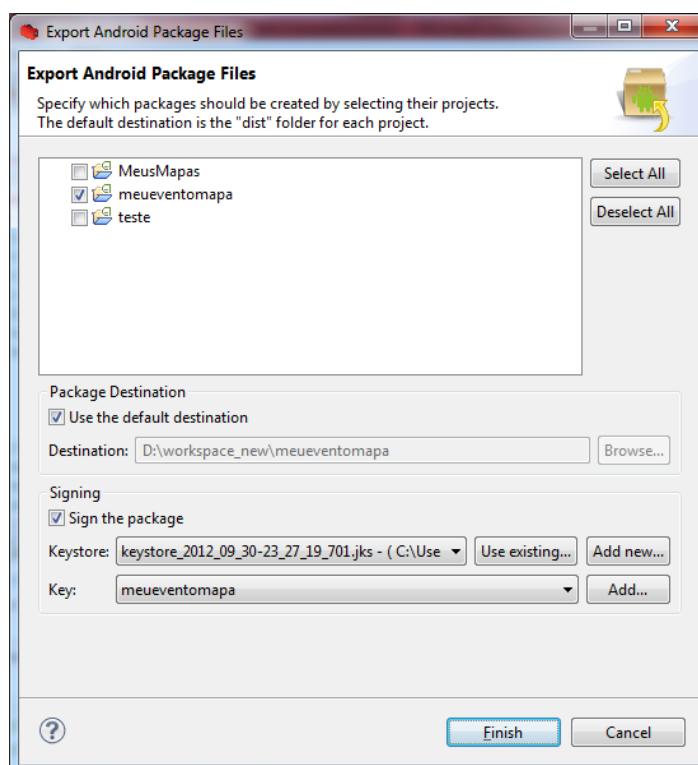


Figura 105 – Gerando o .apk, selecionando o projeto e a chave.

Ao confirmar o projeto e chave, no momento da compilação deverá se inserir a senha da chave privada.

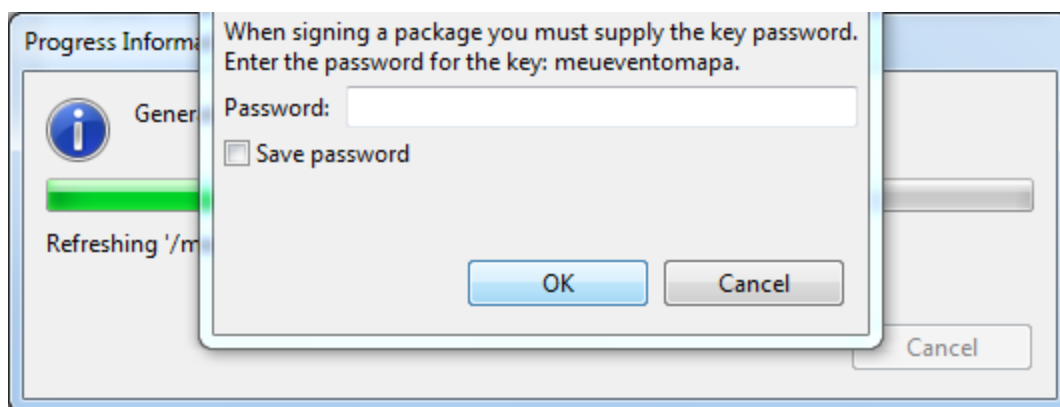


Figura 106 – Ao gerar o apk, foi solicitada a senha da chave.

Fonte: (Próprio Autor, 2012).

Depois de gerar o .apk, deve-se acessar o *Android® Developer Console* que é o ambiente para o envio dos aplicativos. Com o arquivo será atualizado será selecionado o repositório do ELoc.

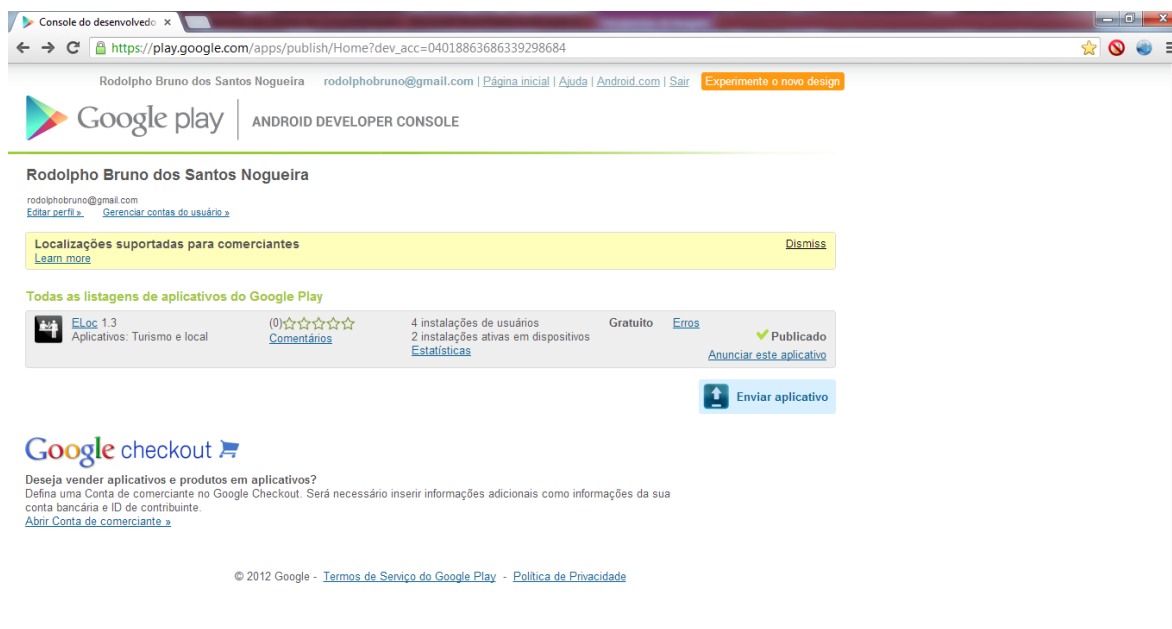


Figura 107– Ambiente para envio e atualização de Aplicativos Android®.

Fonte: (Próprio Autor, 2012).

Na aba Arquivos APK, “Novo Apk” enviara a nova versão do aplicativo Android® ao *Google Play*.

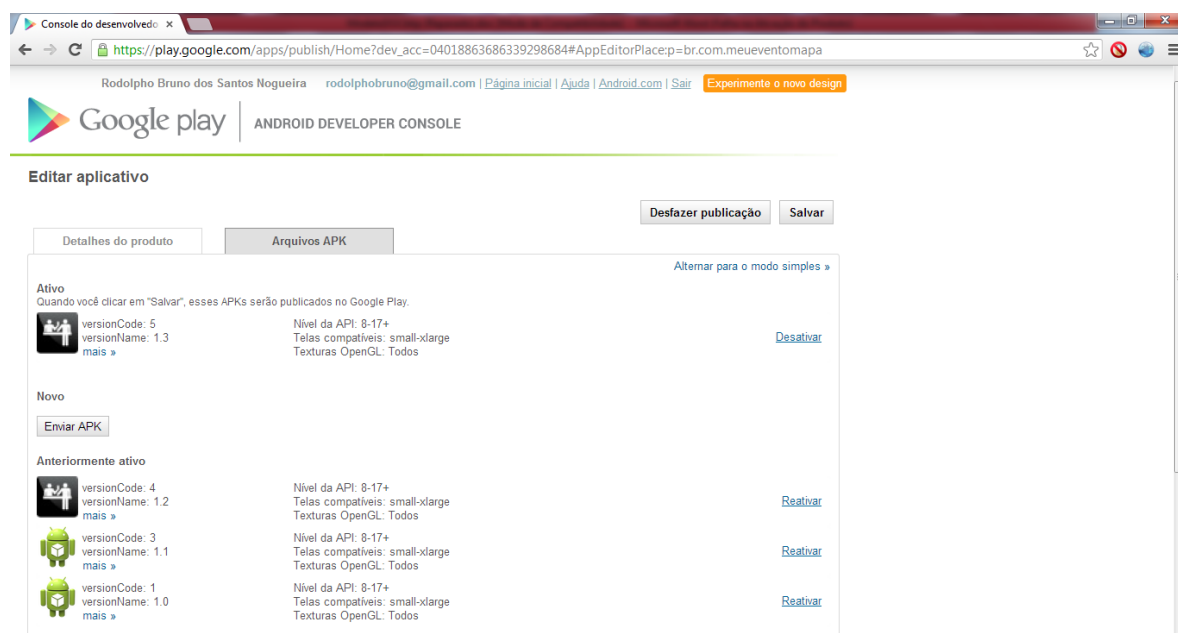


Figura 108 – Repositório de versões lançadas.

Fonte: (Próprio Autor, 2012).

Em seguida aparecerá a opção de escolher o arquivo, onde será selecionado o aplicativo gerado pelo *Motodev*.

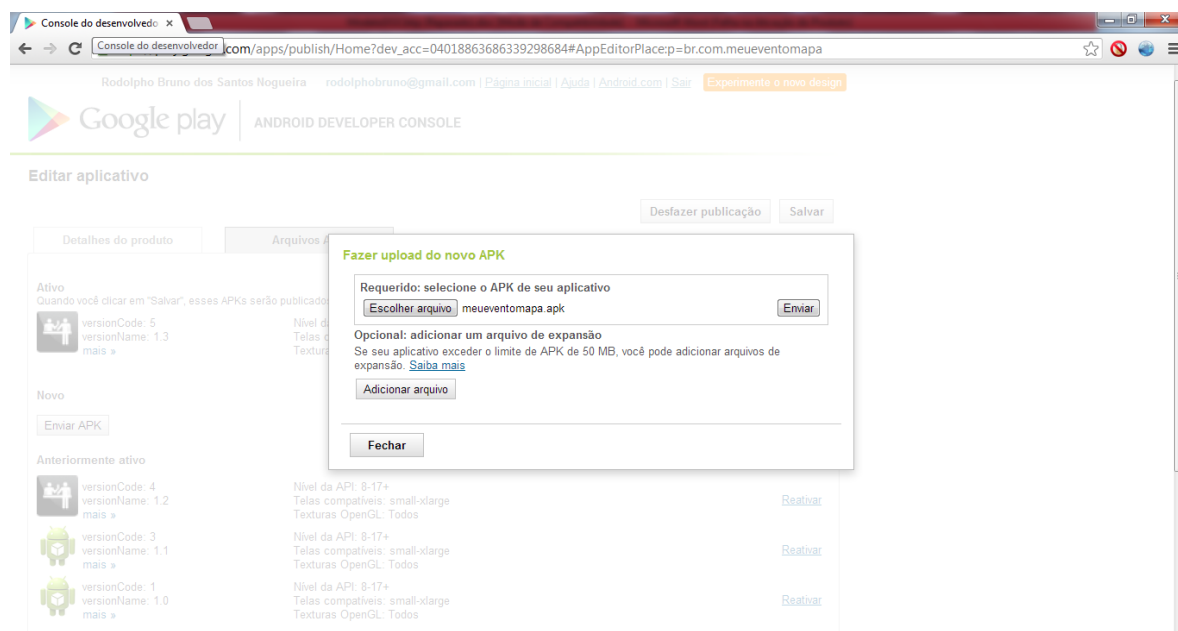


Figura 109 – Enviando uma nova versão.

Fonte: (Próprio Autor, 2012).

No momento em que o aplicativo é enviado as especificações são mostradas em uma caixa de confirmação, ao confirmar esta etapa o aplicativo estará disponível no *Google Play*.

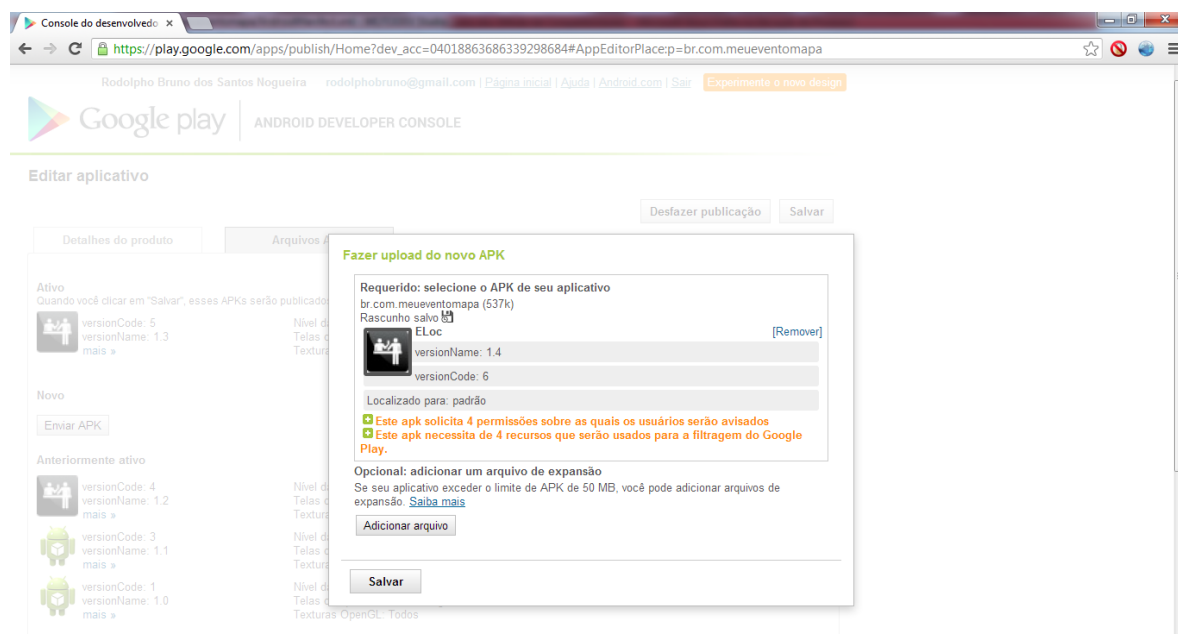


Figura 110 - Ao homologar o arquivo enviado o Google Play, será pedida confirmação de envio da versão 1.4.

Fonte: (Próprio Autor, 2012).

Existe um histórico de versões onde o desenvolvedor poderá deixar explícito ao usuário o que foi alterado ou ajustado na versão enviada do sistema, ao enviar uma versão o *Google Play* já detecta para quais versões da plataforma o sistema desenvolvido é compatível.

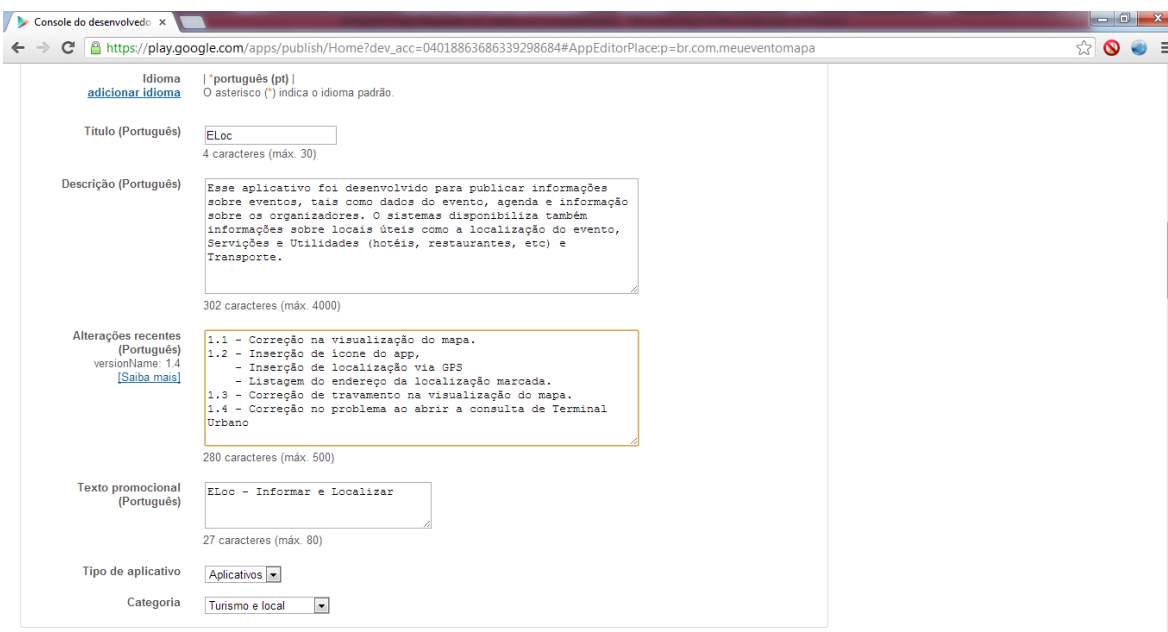


Figura 111– Informando ao usuário as alterações recentes.

Fonte: (Próprio Autor, 2012).

Para que a nova versão esteja disponível é necessário “desativar” a versão anterior 1.3 para que a nova 1.4 possa entrar em vigor, assim para ser atualizada em dispositivo que já possuem o aplicativo, mas em versão anterior, ou para que possa ser disponibilizado para download no *Google Play* com a versão mais atual. Para o novo aplicativo aparecer no *Google Play* pode demorar algum tempo.

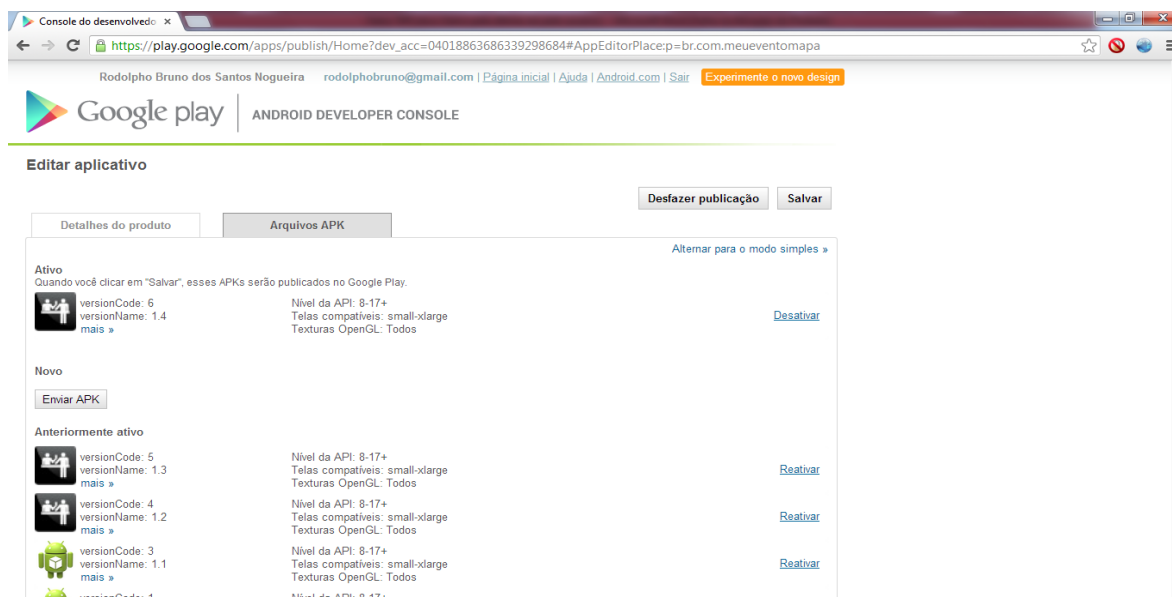


Figura 112 – Nova versão em vigor no Google Play.

Fonte: (Próprio Autor, 2012).

6.3.2. Instalando o aplicativo

- Fazer download utilizando o próprio dispositivo

Para instalar o aplicativo basta encontrar no *Android*® o programa de *downloads* do *Google Play* o *Play Store* e pesquisar pelo nome ELoc e confirmar o download conforme a imagem abaixo.

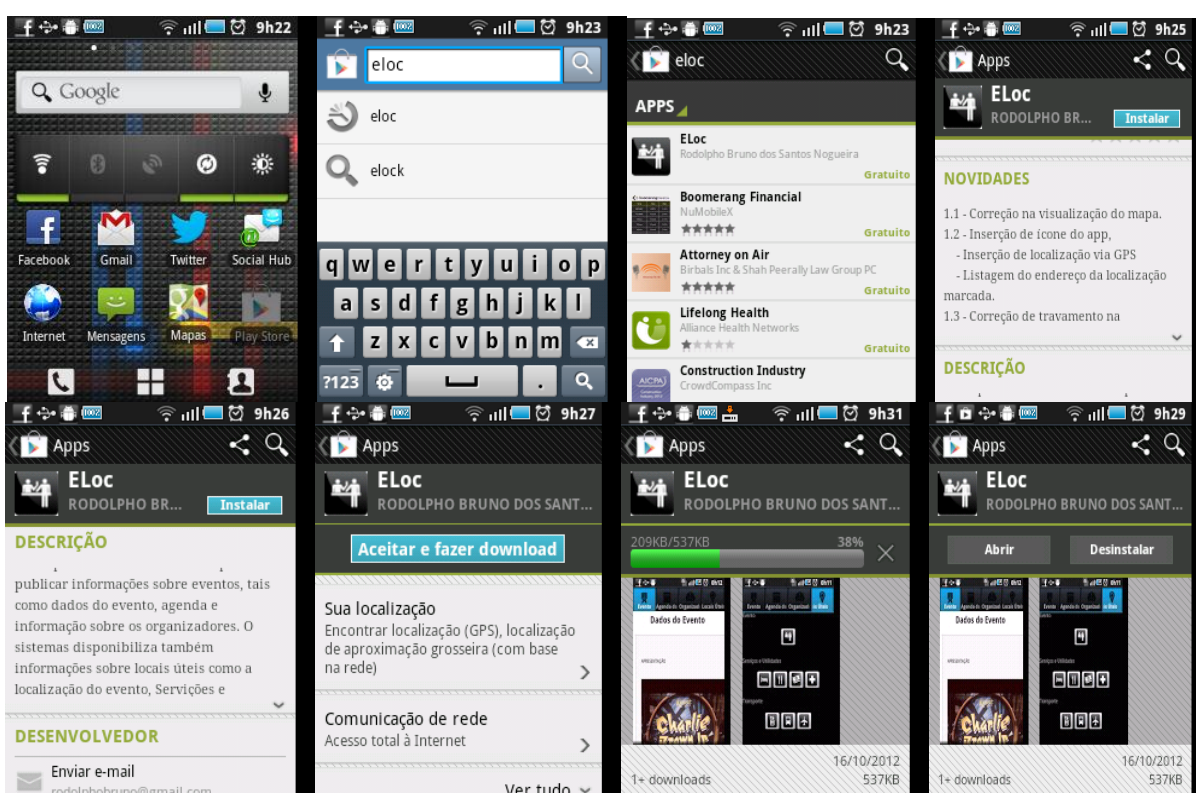


Figura 113 – Buscando o Aplicativo no Google Play Store e em seguida instalando o Eloc.

Fonte: (Próprio Autor, 2012).

Depois de instalado o aplicativo ele e estará pronto para uso!

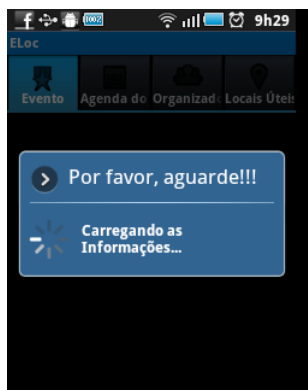


Figura 114 – Aplicativo ELoc baixado do Play Store e funcionando.

Fonte: (Próprio Autor, 2012).

▪ Fazer download utilizando a conta Google

É possível efetuar o download sem necessariamente utilizar o dispositivo, para isso é necessário configurar uma conta de sincronização no aparelho *Android*, no caso a conta *Google*.

Para configurar no dispositivo o usuário deverá acessar Configurações → Contas e Sincronização → Adicionar Conta → Google, no procedimento haverá a opção de criar uma conta ou apenas vincular uma existente, caso o usuário possua uma *Gmail* é o suficiente.



Figura 115 – Configurando a sincronização da conta Google.

Fonte: (Próprio Autor, 2012).

Ao configurar a conta de sincronização as especificações do aparelho configurado são enviadas para o *Google* e é vinculada a conta cadastrada, assim quando o usuário pesquisa por algum aplicativo no *Google play* e o mesmo é compatível com a versão do *Android* aparelho do usuário, aparecerá no site uma opção para download, se aceito para o dispositivo ira automaticamente para o dispositivo cadastrado.

Para fazer o download basta acessar <http://play.google.com/store> e pesquisar por Eloc.

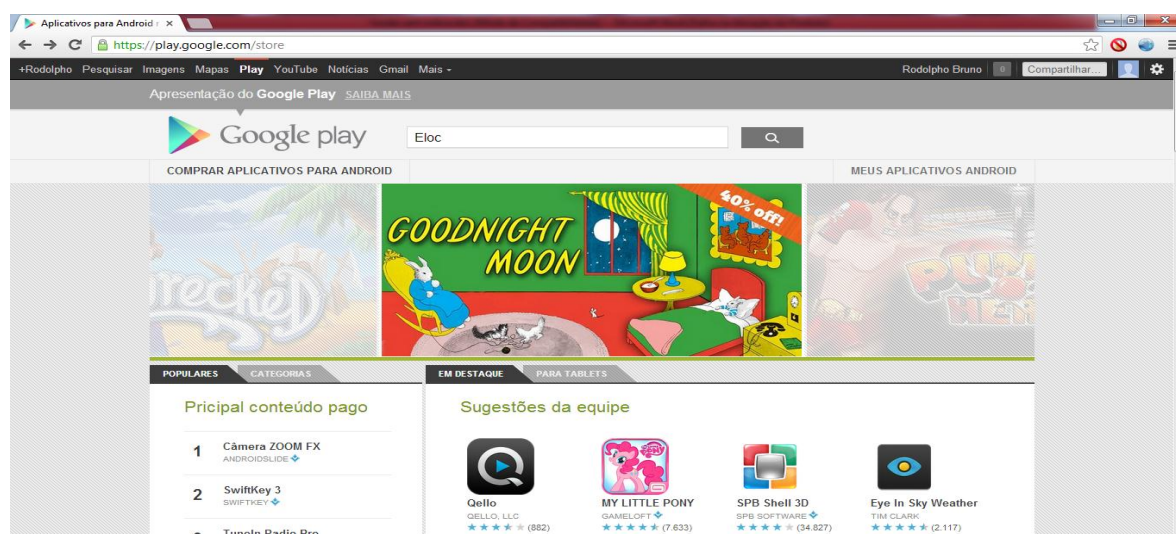


Figura 116 – Pesquisando pelo Aplicativo no Google play.

Fonte: (Próprio Autor, 2012).

Depois de efetuada a busca, O aplicativo ELoc foi listado e abaixo apareceu um botão “INSTALAR”, a partir desse momento o procedimento de instalação já poderá ser efetuado.

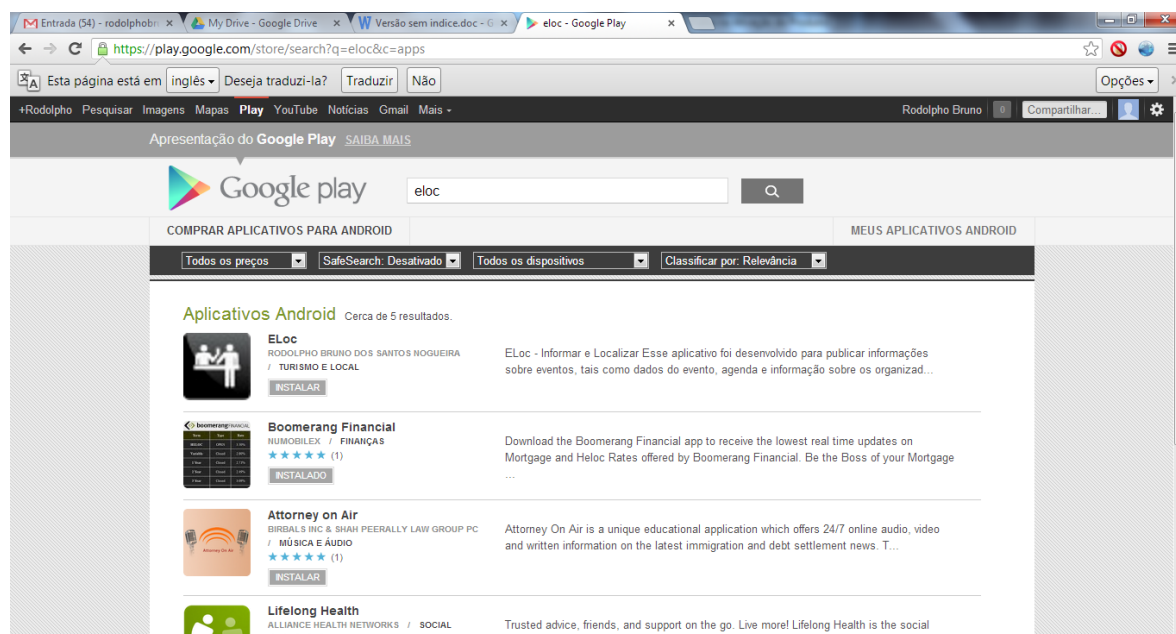


Figura 117 - Eloc Listado no Google Play.

Fonte: (Próprio Autor, 2012).

Mas para mostrar da funcionalidade completa, o cadastro com o ELoc será acessado. Ao acessar ao lado do ícone da aplicação ELoc no site aparecerá o botão “INSTALAR” onde será feita a operação de instalação. Abaixo haverá uma mensagem dizendo se o aplicativo é compatível com a versão do sistema operacional instalado no aparelho.



Figura 118 – Instalando o Aplicativo diretamente do site.

Fonte: (Próprio Autor, 2012).

Ao clicar em instalar, aparecerá uma mensagem de confirmação para qual dispositivo o aplicativo será instalado, depois de confirmado o aplicativo será baixado automaticamente para o aparelho selecionado.

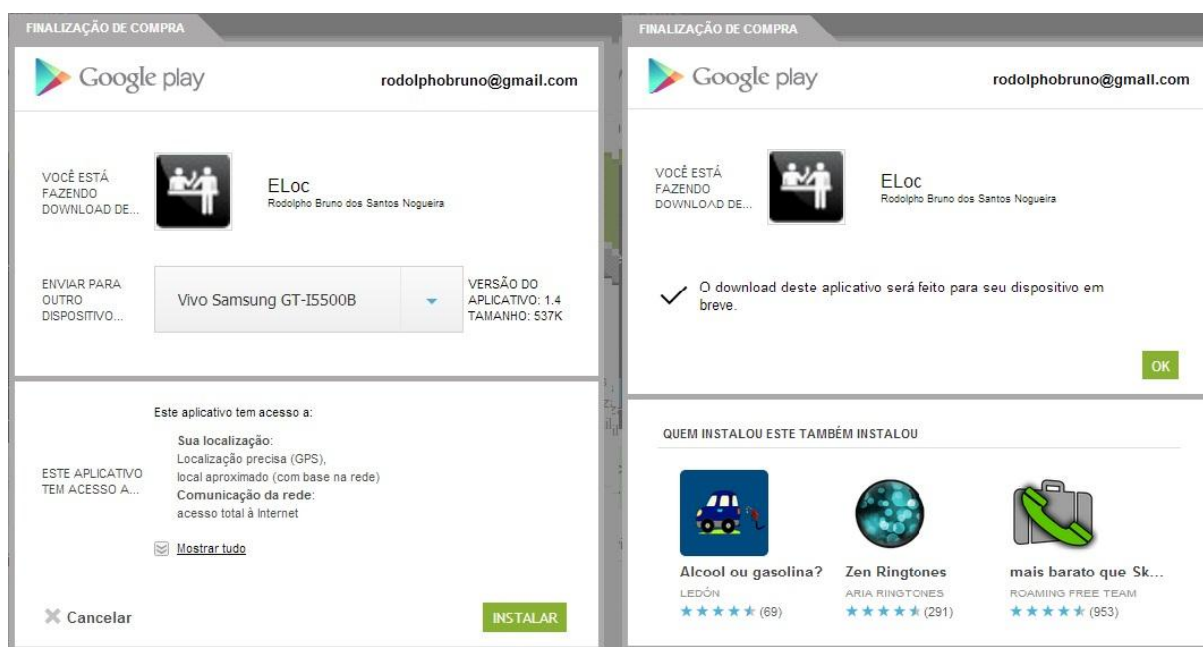


Figura 119 – Confirmação e download do ELoc

Fonte: (Próprio Autor, 2012).

Depois de Confirmado o download, o download automaticamente se iniciará.



Figura 120 – Baixando e Instalando o ELoc

Fonte: (Próprio Autor, 2012).

Depois de instalado o ELoc está pronto para uso.

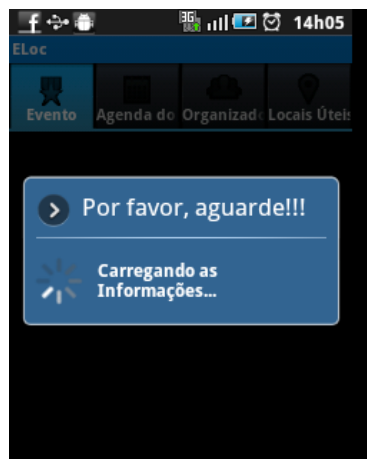


Figura 121 – Depois de baixado e instalado o ELoc está funcionando no Android.

Fonte: (Próprio Autor, 2012).

7. CONCLUSÕES

Tendo em vista o cenário promissor no mercado de aplicações móveis, este trabalho mostrou o grande potencial da plataforma de desenvolvimento *Android*®. Foi desenvolvida uma aplicação *Android*® integrada a tecnologias *OpenSource* (Mysql, PHP) e disponibilizada uma versão para download no *Google Play*, cumprindo assim todos os objetivos propostos no início deste trabalho. O desenvolvimento do aplicativo móvel foi rápido, graças a grande quantidade de material disponibilizado na forma de tutoriais e fóruns especializados com uma grande comunidade ativa. Assim, por meio de uma filosofia de liberdade e acessibilidade, o sistema operacional móvel *Android*® vai se consolidando como o principal do segmento, contribuindo de forma efetiva para o sucesso deste novo e promissor nicho de mercado.

8. REFERÊNCIAS BIBLIOGRÁFICAS

AOSP. Android Open Source Project **Info Tech**, 2012. Disponível em: <<http://source.android.com/tech/security/>>. Acessado em novembro/2012.

ANDROID. **Plataform Versions**, 2012. Disponível em: <<http://developer.android.com/about/dashboards/index.html>>. Acessado em outubro/2012.

ANDROID DEVELOPERS. **API GUIDES**, 2012. Disponível em <<http://developer.android.com/guide/components>>. Acessado em novembro/2012.

APPLE. **O que é o iOS ?**, 2012. Disponível em: <<http://www.apple.com/br/ios/what-is/>> Acessado em outubro/2012.

BADA; **Get a Smart Start for you mobile life**, 2012. Disponível em <<http://bada.com/badadevices.htm>>. Acessado em outubro/2012.

BLACKBERRY. **BlackBerry Brasil**, 2012. Disponível em: <<http://br.blackberry.com/>> Acessado em outubro/2012.

BRIGGS, J.; **Como funciona o Palm webOS, 2009**. Disponível em: <<http://eletronicos.hsw.uol.com.br/palm-webos.htm>>. Acessado em outubro/2012.

CKEDITOR. **What is CKEditor?**, 2012. Disponível em < <http://ckeditor.com/what-is-ckeditor> >. Acessado em novembro/2012.

CODEIGNITER. **Codeigniter at a Glance**, 2012 <http://codeigniter.com/user_guide/overview/at_a_glance.html>. Acessado em novembro/2012.

GIACAGLIA, M. C.; **Organização de Eventos: teoria e pratica**. Pioneira Thomson

Learning, 2004.

HADDAD, I.; **Happy 1 Year Anniversary, MeeGo!**, 2011. Disponível em: <https://meego.com/sites/all/files/users/u19961/meego_anniversary_article.pdf>. Acessado em outubro/2012.

JIPPING, J. J.; **Smartphone Operating Concepts With Symbian OS**. John Wiley & Sons, 2007.

MSDN. **Windows Mobile**, 2008. Disponível em: <<http://msdn.microsoft.com/pt-br/library/bb158486.aspx>>. Acessado em outubro/2012.

MYSQL. **Why Myslq ?**, 2012. Disponível em <<http://www.mysql.com/why-mysql/>>. Acessado em novembro/2012.

PHP. **What is PHP ?**, 2012. Disponível em <http://www.php.net/manual/pt_BR/intro-what-is.php>. Acessado em novembro/2012.

OLIVEIRA, R. C. de; **Dispositivos Moveis Portáteis: Tecnologias envolvidas no desenvolvimento de aplicativos para smartphones**, 2011. Disponível em: <<http://pt.scribd.com/doc/74798706/Monografia-DISPOSITIVOS-MOVEIS-PORTATEIS>>. Acessado em Outubro/2012.

SALES, J; **SYMBIAN OS INTERNALS: Real Time Kernel Programming**. John Wiley & Sons, 2006.

SURFACE; **SURFACE**, 2012. Disponível em <<http://www.microsoft.com/Surface/en-US>>. Acessado em novembro/2012.

VALENTINO, L.; SCHNEIDER, H.; ROBBIE, S.; **Aplicações Móveis. Arquitetura, projeto e desenvolvimento**. PEARSON Makron Books, 2005.

VALLIN, A.; **Brew MP, o chip que transforma um celular em smartphone**. 2010.

Disponível em: <<http://www.tecmundo.com.br/celular/6136-brew-mp-o-chip-que-transforma-um-celular-em-smartphone.htm#ixzz2BCGhh4mU>>. Acessado em outubro/2012.

WEBOS; **Palm Web Os**. 2011. Disponível em: < <http://webos.org/phones/>>. Acessado em novembro/2012.

WINDOWS PHONE 8; **Novidades no Windows Phone 8**. 2012. Disponível em:

<<http://www.windowsphone.com/pt-br/how-to/wp8/start/whats-new-in-windows-phone>>.

Acessado em novembro/2012.