
Coordenadoria do Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

Desenvolvimento de um aplicativo Android utilizando geolocalização

Luciano da Silva Oliveira

Msc. Diogo Fernando Trevisan (Orientador)

Novembro de 2014

Desenvolvimento de um aplicativo Android utilizando geolocalização

Luciano da Silva Oliveira

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Luciano da Silva Oliveira e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 7 de novembro de 2014.

Prof. Msc Diogo Fernando Trevisan
(Orientador)

Agradecimentos

Quero agradecer, em primeiro lugar, a Deus, pela força e coragem durante toda esta longa caminhada.

Aos meus pais, Maria Lucia da Silva Oliveira e Pedro Batista de Oliveira, pelo apoio e incentivo que deram durante a minha graduação.

Aos meus amigos e familiares que sempre acreditaram e me incentivaram para minha formação.

Ao Professor Diogo Fernando Trevisan pela paciência, compreensão e orientação na realização deste trabalho.

Às pessoas que contribuíram diretamente ou indiretamente a esse trabalho, o meu muito obrigado.

Luciano da Silva Oliveira

Resumo

Hoje em dia muitas pessoas estão aprendendo ou já sabem um segundo idioma, porém, deixam de praticar e acabam esquecendo o idioma que aprenderam.

Diversas empresas multinacionais estão contratando pessoas que saibam entender e falar fluentemente um segundo idioma. Muitas pessoas pela falta de prática não conseguem obter esses cargos, fazendo com que outra pessoa com mais experiência e com a fluência na língua estrangeira tenha mais chances de conseguir.

Com o avanço da tecnologia e a facilidade de acesso a dispositivos móveis podemos suprir esta necessidade. Com base nessas informações foi criado um aplicativo móvel que mostrará as pessoas que estejam próximas e cadastradas no banco de dados da aplicação possibilitando visualizar quais idiomas falam e trocar mensagens para se encontrarem para um diálogo. Por fim, é apresentado o desenvolvimento de uma aplicação, de forma detalhada, para o sistema operacional *Android* que utiliza os conceitos de geolocalização para permitir a localização de pessoas.

Palavras-chaves: *Android, Fluência, idioma, Computação Móvel, Localização.*

Abstract

Nowadays many people are learning or already know a second language, however, cease to practice and end up forgetting the language they have learned.

Several multinational companies are hiring people who can understand and speak a second language fluently. Many people with the lack of practice are unable to obtain these positions, causing another person with less experience and fluency in a foreign language have more chances of getting the job.

With the advancement of technology and the ease of access to mobile devices we can meet this need. Based on this information was created a mobile application that will show people that are close to and registered in the database of the application making it possible to view which languages they speak and exchange messages to meet up for a dialogue. Finally, it is presented the development of an application in detail, for Android operating system which uses the concepts of geolocation to enable the location of people.

Key-words: *Android, Fluency, Language, Mobile Computing, Location.*

Sumário

Agradecimentos	v
Resumo	vii
Abstract.....	ix
1 Introdução	17
1.1 Objetivo Geral:	17
1.1.1 Objetivos Específicos:.....	17
1.2 Justificativa e motivação.....	18
1.3 Metodologia	18
1.4 Organização do texto	18
2 Android: Geolocalização	21
2.1 Android	21
2.1.1 Arquitetura	21
2.2 Geolocalização.....	23
3 Desenvolvimento	25
3.1 Funcionamento da aplicação.....	25
3.1.1 API Google Maps.....	25
3.1.2 API Facebook.....	27
3.1.3 Chat	27
3.2 Descrição do aplicativo.....	28
3.3 Classes da implementação do aplicativo	31
3.3.1 Modulo Aplicação	32
3.3.2 Diagrama de caso de uso.....	33
3.3.3 Diagrama de Atividade.....	33
4 Resultados e Discussões	35
4.1 Detalhes dos testes	35
5 Conclusão	43
5.1 Trabalhos futuros	43
Referências Bibliográficas.....	45
Apêndice A: Instalação e configuração do Eclipse	47
Apêndice B: Instalação das APIs.....	49
B.1 Facebook API	49
B.2 Google Maps API	52

Lista de Tabelas

2.1.1 Versões do Android.....	22
3.3.1 Classes do modulo da aplicação.....	32

Lista de Figuras

2.1. 1: Arquitetura do Android.....	21
3.1.1: Tipos de mapas: normal, satélite e terreno.....	26
3.1.2: Representação Objeto.....	28
3.1.3: Representação Array.....	28
3.2.1: Representação do login.....	29
3.2.2: Representação do painel principal.....	29
3.2.3: Representação do painel mensagens.....	30
3.2.4: Representação do mapa.....	30
3.2.5: Representação do chat.....	31
3.3.1: Visualização dos dados pelo phpMyAdmin.....	31
3.3.2: Diagrama de caso de uso do aplicativo.....	33
3.3.3: Diagrama de Atividade.....	34
4.1.1: Tela inicial da aplicação.....	35
4.1.2: Tela solicitando permissão.....	36
4.1.3: Tela principal da aplicação.....	36
4.1.4: Tela mapa com marcador.....	37
4.1.5: Tela mapa com menu.....	37
4.1.6: Tela mapa com janela no marcador.....	38
4.1.7: Tela de informações do usuário selecionado.....	38
4.1.8: Tela de Chat.....	39
4.1.9: Tela de notificação informando nova mensagem.....	39
4.1.10: Tela painel de mensagens.....	40
4.1.11: Tela de chat recebendo e enviando mensagens.....	40
4.1.12: Tela usuário recebendo mensagem.....	41
4.1.13: Menu com tela de mensagens.....	41
4.1.14: Tela de Painel de Mensagens.....	42
4.1.15: Painel de mensagem, deletando usuário.....	42
A. 1: Pagina oficial do Eclipse.....	47
B.1.1: Pagina oficial Facebook Developers.....	49
B.1.2: Tela de Import do Eclipse.....	49
B.1.3: Import API.....	50
B.1.4: Criando um aplicativo no site do Facebook.....	50
B.1.5: Dados para criar um aplicativo.....	51
B.1.6: Código para adquirir chave Hash.....	51

B.1.7: Inserindo chave Hash.	52
B.2.1: Baixando Google Play Services.	52
B.2.2: Obtendo chave SHA-1.....	53
B.2.3 Habilitando Google Maps Android API V2.	53
B.2.4: Inserir chave SHA-1.....	54
B.2.5: Obtendo API Key.	54

Capítulo 1

Introdução

Com a grande evolução da tecnologia, surgiu a computação móvel, que pode ser considerada como um novo paradigma computacional [7].

Com a diminuição dos custos dos dispositivos móveis, a computação móvel tornou-se viável não somente para empresas, como também para as pessoas de forma geral. A computação móvel tem como objetivo um conjunto de serviços comparáveis aos existentes em um sistema distribuído que permite a mobilidade [22].

A computação móvel permite que os usuários acessem informações e se comuniquem a qualquer instante e em qualquer lugar. Com isso os dispositivos móveis conseguiram um grande espaço no mundo se tornando fundamentais para nosso dia-a-dia [22].

A cada dia, novos aplicativos são desenvolvidos a partir da tecnologia atual. Um fator importante que foi criado a partir dessa evolução foi a conexão com a internet em diferentes pontos de acesso. Muitos aplicativos utilizam a internet para fazer atualizações, acessar páginas específicas entre outros. Entretanto existem aplicativos que não tem a necessidade de acessar a internet [22].

Os dispositivos móveis mais atuais já possuem *Global Positioning System* (GPS) embutido e as empresas de telefonia vendem ao consumidor final com preços acessíveis. O GPS permite localizar qualquer dispositivo na superfície terrestre podendo ser utilizado para finalidades pessoais ou profissionais [8].

1.1 Objetivo Geral:

Este trabalho tem como objetivo geral um estudo sobre sistema operacional *Android*, Geolocalização e aplicações utilizando *Application Programming Interface* (API) do Google Maps e Facebook.

1.1.1 Objetivos Específicos:

Os objetivos específicos são:

- Pesquisar o uso da API do Google Maps;
- A Geolocalização utilizando dispositivo *Android*.
- Implementar um aplicativo móvel que realiza a localização de pessoas próximas ao dispositivo móvel;
- Testar o software desenvolvido com o intuito de verificar o correto funcionamento de sua execução.

1.2 Justificativa e motivação

Com a grande quantidade de pessoas estudando um segundo idioma, o domínio ainda deixa a desejar por não praticar a fluência [17]. A motivação deste trabalho é aproximar as pessoas que queiram praticar algum idioma através do uso de seus dispositivos móveis.

1.3 Metodologia

Para a realização do estudo sobre Sistema Operacional *Android*, Geolocalização e aplicações utilizando as APIs foram consultados livros, revistas, artigos, teses, e dissertações existentes em bibliotecas digitais.

A localização de um dispositivo móvel é realizada através de um sistema de geolocalização utilizando triangulação de torres de telefonia e acesso à internet.

Para o desenvolvimento do aplicativo móvel foi escolhido o sistema operacional *Android* 4.4 Kitkat com API 19, pelo fato de ser um sistema aberto (*open source*) e possuir diversas documentações sobre sua utilização. Para desenvolvimento foram utilizados os softwares *Android* SDK e Eclipse IDE. A conexão com o banco de dados é online através de um servidor em PHP e o Sistema de Gerenciamento de banco de dados MySQL.

Na realização dos testes com o aplicativo foram utilizados dois celulares Modelo Moto G, processador 1.2 GHz Quad Core.

1.4 Organização do texto

Capítulo 1

Introdução

Neste capítulo é apresentada a introdução deste trabalho bem como os seus objetivos gerais e específicos, a justificativa e como será a metodologia do trabalho.

Capítulo 2

Android: Geolocalização

No capítulo 2 apresenta-se um estudo sobre a arquitetura do sistema operacional, versões e tipos de localizações de dispositivos.

Capítulo 3

Desenvolvimento

No Capítulo 3 é apresentado desenvolvimento da aplicação.

Capítulo 4

Resultado e Discussões

No capítulo 4 são apresentados os resultados obtidos e as discussões.

Capítulo 5

Considerações Finais

No capítulo 5 apresenta-se a conclusão do trabalho e os trabalhos futuros.

Apêndice A**Instalação e configuração do Eclipse IDE**

No apêndice A apresenta-se a instalação e configuração do Eclipse IDE no sistema operacional Windows 8.

Apêndice B**Instalação das APIs**

No apêndice B apresenta-se a instalação das APIs Facebook e Google Maps.

Capítulo 2

Android: Geolocalização

2.1 Android

O Google anunciou o sistema *Android*, o qual foi desenvolvido através de uma aliança multinacional de tecnologia e da indústria móvel chamada *Open Handset Alliance* (OHA), tendo como objetivo promover uma plataforma *Open Source* amigável para desenvolvedores e flexibilidade para desenvolver produtos [19].

O *Android* começou em 2005 quando o Google adquiriu o controle de uma pequena empresa ainda em fase inicial. Em meio a especulações do mercado, começaram o desenvolvimento do novo *Android* baseado em *Linux*, sistema operacional gratuito para computadores [2].

A *Open Handset alliance* é um grupo que foi criado em 5 de novembro de 2007, formado por mais de trinta empresas que se uniram para acelerar e inovar dispositivos *mobile*, sendo a primeira plataforma móvel livre e aberta. O *Android* é constituído com base no *Kernel Linux* e utiliza uma máquina virtual para otimizar memória e recursos [20].

2.1.1 Arquitetura

O sistema *Android* é uma pilha de 5 camadas diferentes e cada camada é um grupo de componentes de software. A Figura 2.1.1 mostra a arquitetura do Sistema Operacional *Android*. Todo sistema operacional *Android* utiliza como base o *Kernel Linux* 2.6, porém esse kernel foi melhorado para que possa atender às necessidades e as limitações dos dispositivos móveis [4].

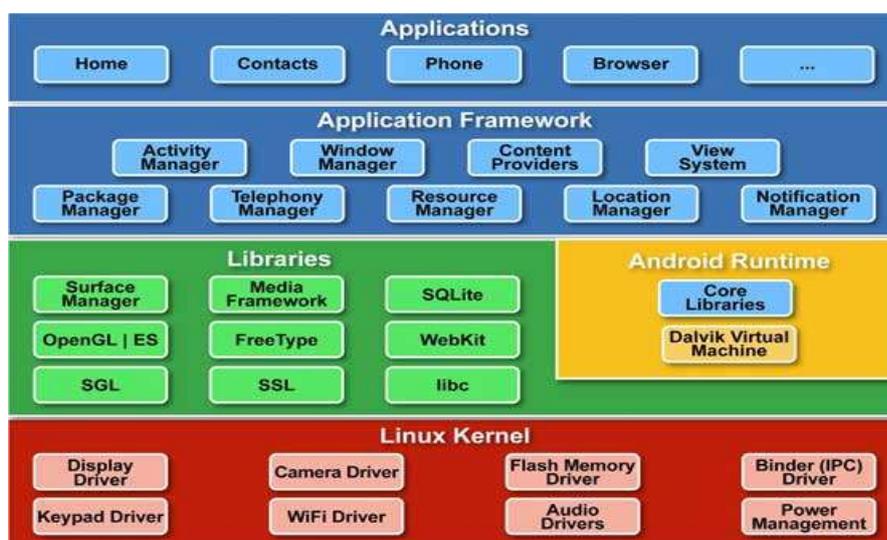


Figura 2.1. 1: Arquitetura do Android. Fonte: [23]

O *kernel do Linux* age como uma camada de abstração entre o hardware e outras camadas de software. O *Android* usa o *Linux* para todas as funcionalidades como gerenciamento de memória, gerenciamento de processador, rede entre outros.[23]

A próxima camada é *Libraires*, que são bibliotecas nativas do *Android*, as quais utilizam diversos componentes do sistema. A implementação do sistema é através da biblioteca padrão do C (*libc*), mas com licença *Berkeley Software Distribution* (BSD) e otimizada para dispositivos *mobile*. O *Android* possui também bibliotecas para suporte de imagem, áudio e vídeo, oferecendo suporte para ambientes gráficos em 2D e 3D. Uma biblioteca nativa permite utilizar o banco de dados relacional *SQLite*, e um *render* de fontes bitmap e vetoriais [1].

A camada *Android RunTime* oferece suporte para que as aplicações sejam executadas. Os componentes desta camada são o *core libraries* que executa aplicações nativas do *Android* e o *Dalvik Virtual Machine* [25].

A *Dalvik Virtual Machine* usa recursos do núcleo do *Linux*, como gerenciamento de memória a baixo nível e *multi-threading*. Com isto cada aplicativo roda seu próprio processo, sendo possível rodar diversos aplicativos simultaneamente [24].

A camada *Application Framework* disponibiliza as funcionalidades para construção de aplicativos através das bibliotecas nativas. Os desenvolvedores utilizam todos os recursos necessários que o *Android* pode oferecer. Existem quatro tipos de componentes na camada: *services, activities, broadcast receivers e content providers*, existem também recursos que são necessários para utilização: *layouts, strings*, estilos e imagens e o arquivo manifesto que é utilizado para declaração de todos os componentes da aplicação e recursos [1].

A camada *Applications* possui uma grande quantidade de aplicativos embutidos no sistema, como aplicativos nativos e desenvolvidos por terceiros, adquiridos através do Google Play [9].

Para facilitar a distribuição de aplicativos *Android* a empresa Google desenvolveu o Google Play, que tem como objetivo organizar os aplicativos. O site conta com esquema de comentários, classificação e qualificação. Assim os desenvolvedores podem saber quais são os pontos fracos e fortes dos seus aplicativos baseado na experiência dos usuários [3].

O *Android* é uma plataforma aberta de desenvolvimento para fabricantes e desenvolvedores que queiram melhorar suas ideias. Desde a criação do Sistema Operacional *Android*, ele passou por diversas modificações em curto espaço de tempo. Na Tabela 2.1.1 podemos ver a evolução do *Android* 1.0 até a versão 4.4 [1].

Tabela de Versões do Android.

Codename	Versões
(Nenhum)	1.0 – 1.1
Cupcake	1.5
Donut	1.6
Eclair	2.0 – 2.1
Froyo	2.2.x
Gingerbread	2.3 – 2.3.7
Honeycomb	3.0 – 3.2.x
Ice Cream Sandwich	4.0.0 – 4.0.4
Jelly Bean	4.1.x – 4.3
KitKat	4.4

Tabela 2.1.1: Versões do Android Fonte: [1]

A cada versão, o *Android* corrige as falhas da versão anterior e a melhora. A partir da versão 1.5, os nomes são atribuídos como nomes de doces e podemos analisar que sempre é em ordem alfabética. Até hoje a Google não disponibilizou o nome das duas primeiras versões do sistema com letras “A” e “B” [26].

2.2 Geolocalização

O Sistema Baseado em Localização (SBL) foi desenvolvido pelo Departamento de Defesa dos Estados Unidos na década de 70, o qual implantou o *Global Positioning System* (GPS), mas somente na década de 90 o sistema ganhou destaque devido a uma legislação americana em 1996, na qual as operadoras foram obrigadas a disponibilizar a localização das chamadas realizadas. Com isto elas tiveram que se adaptar e investir mais em infraestrutura para permitir realizar esse tipo de localização [5].

As principais aplicações do SBL são:

- Serviço de informação: é chamado de “páginas amarelas” no qual o objetivo é consultar locais, eventos e atrações [21].
- O serviço de rastreamento: é utilizado geralmente pelo governo ou empresas privadas para auxiliar equipes na localização de cargas, veículos e pessoas em situações de emergência [21].
- Gerenciamento de recurso: é utilizado em um grande volume de dados e processamento, geralmente são gerenciados por frota de ônibus, transportadoras ou cooperativas de taxis para alocação de recurso em determinadas regiões [21].
- O sistema de navegação: auxilia para que uma pessoa possa se deslocar de um ponto A até um ponto B [21].

Existem diversas tecnologias para geolocalização. Pode-se citar o *Global Positioning System* (GPS), *Assisted Global Positioning System* (A-GPS), *Cell Identification*, *Angle of Arrival* (AOA) e *Time of Arrival* (TOA) [21].

O GPS é baseado em sinais que são transmitidos a partir de 24 satélites. Para poder realizar a localização com o GPS deve-se estar em contato com quatro satélites, e ao mesmo tempo ser capaz de determinar latitude, longitude e altitude. A desvantagem desse sistema é a falta de recepção de sinais GPS no interior de edifícios e casas [12].

A-GPS é a evolução do GPS, que utiliza as informações da rede para determinar a posição dos quatro satélites com os quais precisa sincronizar, poupando energia da bateria. O posicionamento é feito inicialmente no cliente, que envia os dados para o servidor de localização e executa os cálculos e tem uma precisão maior no cálculo do posicionamento [21].

Um dos primeiros métodos adotados pelo SBL é o *Cell Identification*, uma tecnologia barata e com pouca precisão que utiliza Estações de Rádio Base (ERBs). Neste método a localização é estimada através da antena que o dispositivo está conectado. O dispositivo sempre conecta-se a antena mais próxima, ou seja, que possua o sinal mais forte [14].

No método AOA, uma antena especial na ERBs obtém a localização do dispositivo a partir dos cálculos da direção e ângulo em que o sinal chega às antenas. Este método tem alta precisão mas pode sofrer interferências e possui um custo elevado, tornando-se inviável [21;11]

A TOA monitora o tempo de chegada do sinal transmitido pela estação móvel e recebidos pelo ERBS. Esse método é chamado de triangulação de ERBS, e é utilizado na

maioria dos sistemas celulares atuais. Para receber a localização esse sistema tem um custo elevado devido aos temporizadores em cada estação base [21].

Hoje em dia os dispositivos móveis utilizam o A-GPS para obter a localização, porém o sistema ainda não é perfeito e está sujeito a *bugs* e falhas. Entretanto o A-GPS torna-se a utilização do GPS bem mais rápida, evitando que se gaste tempo para encontrar a localização do dispositivo móvel [18].

Capítulo 3

Desenvolvimento

Este capítulo está dividido em 3 seções, A seção 3.1 aborda o funcionamento e estrutura do aplicativo móvel. Na seção 3.2 será descrito a aplicativo. Na seção 3.3 são descritas as classes da aplicação e os diagramas de caso de uso e atividades.

3.1 Funcionamento da aplicação

O aplicativo desenvolvido para o sistema operacional *Android* tem como objetivo localizar dispositivos móveis próximos e disponibilizar uma forma de comunicação utilizando um chat.

A forma de comunicação com a base de dados online é feita através da internet e para encontrar a localização é utilizada a API do Google Maps.

Para efetuar o cadastro do usuário na aplicação será utilizada a API do Facebook que tem como objetivo obter informações da conta do usuário no Facebook.

3.1.1 API Google Maps

A API do Google Maps para *Android* permite exibir um mapa do Google em uma aplicação. Além da funcionalidade de mapeamento ela também suporta uma grande quantidade de iterações que são consistentes com o modelo de interface *Android*. A classe utilizada como um objeto do mapa é a *GoogleMap*, dentro de sua interface será representado um *MapFragment* ou *MapView* [10].

O Google Maps fornece diversas operações automaticamente, como, conectar ao serviço do Google Maps, exibir vários controles como *pan* e *zoom* e responde a *pan* e *zoom* com gestos para *zoom in* ou *out*. Além disso, podemos controlar o comportamento de mapas através dos objetos que representam os mapas (instancias do *GoogleMap*) [10].

O *MapFragment* é uma subclasse da classe *Android Fragment* que permite que você coloque um mapa em um Fragmento Android. Os objetos do *MapFragment* atuam como recipientes para o mapa e fornecem acesso ao objeto *GoogleMap*. Para se utilizar a API do Google Maps o *Android* requer API nível 12 ou superior para obter suporte ao objeto *MapFragment*. [10].

O *MapView* é uma subclasse do *Android View* (classe que representa a construção de componentes de interface do usuário), e permite colocar um mapa no *Android View*. O *MapView* é muito parecido com o *MapFragment*. Ele atua como um recipiente para o mapa e disponibiliza funcionalidades de mapa através do objeto *GoogleMap* [10].

Existem cinco tipos de mapas disponíveis na API do Google Maps para *Android* [10]:

- Normal – Estradas, recursos naturais, etiqueta de estradas e de recursos.
- Híbrido – Dados fotográficos de satélite com mapas rodoviários. Etiquetas de estradas e de recursos também são visíveis.
- Satélite – Dados fotográficos de satélite. Etiquetas de estradas e recursos não são visíveis.
- Terreno – Dados topográficos. O mapa inclui cores, curvas de nível, etiquetas, perspectiva e sombreamento. Algumas estradas e rótulos são visíveis.
- Default – O mapa será processado com uma grade vazia e os detalhes das imagens não serão carregadas.

Para definir um tipo de mapa, deve-se que fazer a chamada do método `setMapType()` do objeto *GoogleMap* passando uma constante definida pela classe *GoogleMap*. Por exemplo:

```
GoogleMap map; // Atribuindo o tipo do mapa para "híbrido"
map.setMapType (GoogleMap. MAP_TYPE_SATELLITE);
```

A Figura 3.1.1 mostra a comparação dos tipos normal, satélite e terreno para o mesmo local.

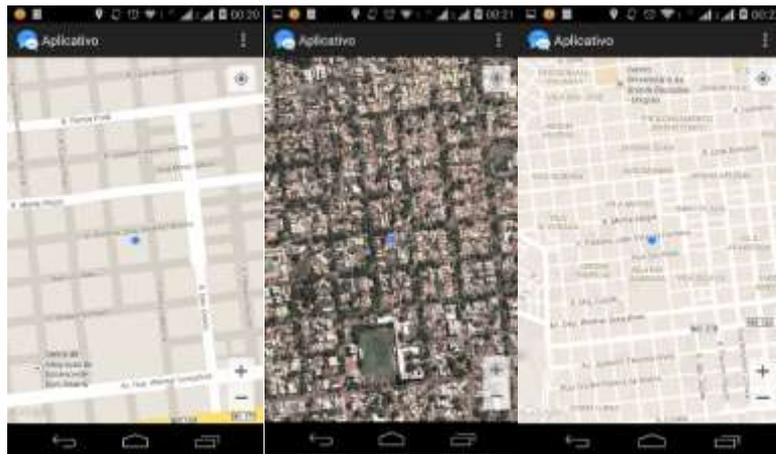


Figura 3.1.1: Tipos de mapas: normal, satélite e terreno.

A API do Google Maps permite configurar uma visualização do mapa que disponibiliza a posição da câmera, ferramentas para zoom, rolamento, inclinação, tipo de mapa, botões e bússola disponíveis na tela [10].

Um mapa do Google é projetado para preencher toda a região de um recipiente (layout onde será exibido o mapa), normalmente um *MapView* ou *MapFragment*. Vários aspectos do mapa são apresentados e são definidos pelas dimensões do recipiente [10]:

- Área de visualização da câmera.
- Controles de mapa estão posicionados à borda do mapa.
- Informações legais, tais como declarações de direitos autorais ou logotipo do Google aparecem ao longo da borda inferior do mapa.

3.1.2 API Facebook

A API do Facebook possui diversas classes que permitem integrar o Facebook com uma aplicação *Android*. Ao aceitar as permissões, as informações do usuário serão compartilhadas com o aplicativo [16]. A API principal do Facebook é a API *Graph*. Esta permite gravar e ler dados. O mais recente paradigma da API utiliza objetos e os relacionamentos ou conexões entre eles. Com a API é possível excluir, recuperar, localizar, publicar objetos (fotos, postagens, notas e etc) e até mesmo filtrar resultados [6].

Para utilizar a API do Facebook algumas classes são necessárias. A classe principal Facebook implementa uma série de métodos *request()*, e oferece classes auxiliares para tornar as chamadas da API assíncronas. A classe Facebook *Dialog* possui um método para autenticar uma visualização padrão na WEB [6]

O *OAuth 2.0* é uma estrutura de autorização que permite um aplicativo obter acesso limitado a um serviço HTTP, sendo necessário a aprovação entre o proprietário do serviço HTTP e a aplicação [13].

Nesta abordagem, a API utiliza o diálogo de autenticação na Web do Facebook dentro de uma visualização Web. Com isto reduz a complexidade de implementação e ganha a confiança do usuário exibindo as permissões e login [6].

Para desenvolver uma aplicação que utiliza a API Facebook devemos cadastrar o desenvolvedor e o aplicativo no Facebook *Developers*. Ao realizar o cadastro um identificador será disponibilizado. Cada aplicação possui um identificador único (*APP_ID*) e cada desenvolvedor possui um código (*APP_Secret*) [6].

Para que o usuário disponibilize suas informações, devemos solicitar algumas permissões que serão guardadas em uma lista de *string* e posteriormente efetuar a chamada do método *ParseFacebookUtils.logIn()* que tem como objetivo informar ao usuário quais permissões a aplicação estará acessando [6].

O método *authorize()* pode gravar comentários e preferências, ler informações do usuário e dar as credencias de acesso. Esse método também cuida da solicitação de autenticação para o Facebook e retorna o *access_token* que é utilizado em toda a sessão do Facebook. Depois que o usuário aceitar as permissões o aplicativo poderá acessar o Facebook e realizar as ações solicitadas nas permissões [6].

Somente após todos os procedimentos de permissões e autorizações serem aceitas será possível utilizar a API para obter as informações do usuário.

3.1.3 Chat

Para que os usuários troquem mensagens foi implementado um chat. Para isso foi utilizada uma ferramenta chamada *JavaScript Object Notation (JSON)*, que é baseada em uma estrutura *javascript* [15].

O JSON é composto por duas estruturas:

- Uma coleção de pares de nome-valor. Em diversas linguagens é caracterizado como um objeto, *struct* ou *arrays* associativa.

- Uma lista ordenada de valores, que pode ser caracterizado com uma *array*, vetor, lista ou sequência.

Para representar o JSON utilizamos a seguinte forma: Um objeto é um conjunto desordenado de pares de nome/valor. Para começar um objeto utiliza-se “{” (chave de abertura) e termina-se com (chave de fechamento) “}”. Cada nome é seguido por “:” (dois pontos) e os pares de nome/valor são seguidos por “,” (virgula), assim como podemos visualizar na Figura 3.1.2 [15].

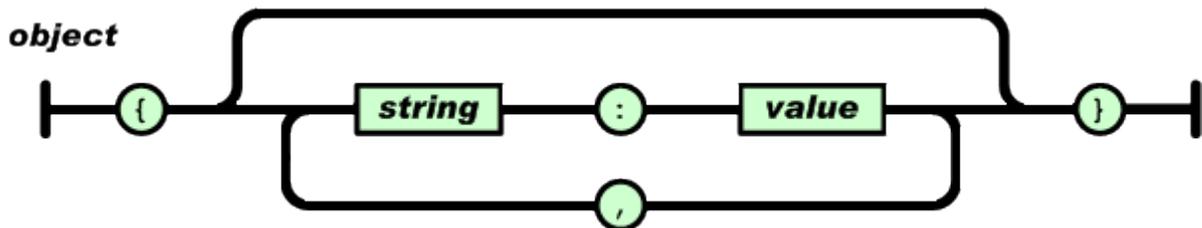


Figura 3.1.2: Representação Objeto. Fonte: [15].

Um *array* é um conjunto de valores ordenados. O *array* inicia com “[” (colchete de abertura) e termina com “]” (colchete de fechamento), os valores são separados por “,” (virgula), assim como mostra a Figura 3.1.3 [15].

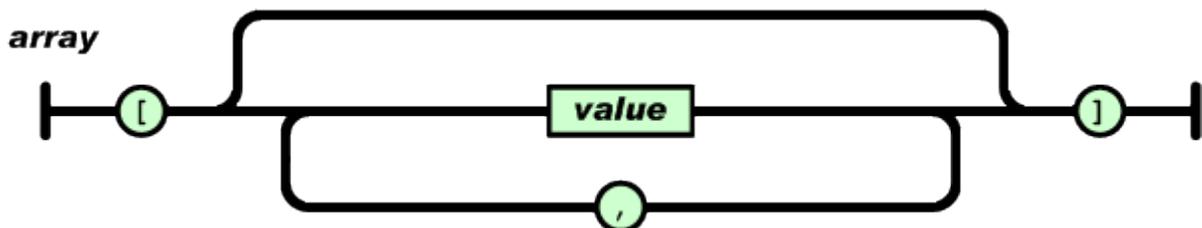


Figura 3.1.3: Representação Array. Fonte: [15].

Para que o aplicativo possa efetuar trocas de informações, foi necessário implementar um servidor PHP. O servidor recebe mensagens via JSON. Estas mensagens possuem um pedido para enviar, verificar ou receber mensagens de um celular.

Para enviar, receber e verificar mensagens foram criados três arquivos onde cada um tem uma função específica. Estes arquivos utilizam SQL para inserir informações no banco e o JSON para enviar informações ao dispositivo móvel e também para que o servidor receba informações do dispositivo.

3.2 Descrição do aplicativo

Ao abrir o aplicativo, o usuário entrará na tela de *login*, onde serão necessárias as informações de nome, e-mail, idioma, cidade e foto de perfil que serão obtidas do Facebook. Após efetuar o *login* a tela principal será aberta. Pode-se visualizar um exemplo na Figura 3.2.1.

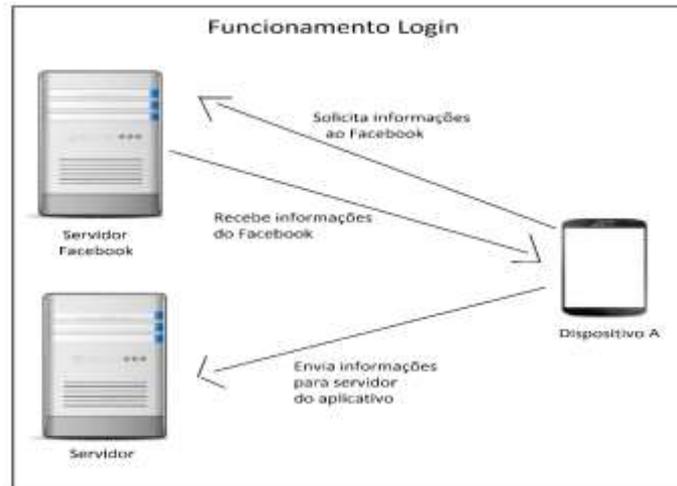


Figura 3.2.1: Representação do login

Na tela principal serão exibidas algumas informações obtidas do Facebook do usuário. Dentro da tela existem quatro opções. Na primeira é possível visualizar as mensagens não lidas e o usuário que enviou a mensagem. Na segunda podemos limitar a distância de busca e na terceira temos a opção de efetuar a busca. Por fim, na última opção de deslogar do aplicativo.

A tela principal ficará efetuando verificações no servidor constantemente para novas mensagens. Caso possua uma nova mensagem uma notificação será exibida. A Figura 3.2.2 mostra o funcionamento do painel principal.



Figura 3.2.2: Representação do painel principal.

Na tela de painel de mensagens serão exibidos apenas os usuários que enviaram mensagens e os usuários que receberam. É possível remover o usuário para que não apareça mais na lista. Ao clicar no usuário, não serão mostradas conversas antigas, somente mensagens não lidas. Na Figura 3.2.3 pode-se visualizar o funcionamento do painel de mensagens.



Figura 3.2.3: Representação do painel mensagens.

Ao realizar uma busca a tela de mapa será exibida e mostrará todos os usuários dentro daquele raio que foi definido na tela principal. Caso possua um ou mais usuários próximos e que tenha utilizado o aplicativo no mesmo dia, serão exibidos marcadores possibilitando selecioná-los. Ao clicar no marcador algumas informações de nome e distancia irão aparecer; Para visualizar mais informações do usuário deve clicar no balão que será exibido. O dispositivo envia sua posição atual para o servidor e recebe todas as informações dos usuários cadastrados no banco de dados, como mostra a Figura 3.2.4.



Figura 3.2.4: Representação do mapa.

Na tela de informações do usuário são exibidas algumas informações referentes ao usuário selecionado e é possível efetuar uma conversa através do botão *chat*. Na tela de *chat* é possível conversar com o usuário correspondente.

Na Figura 3.2.5 pode-se visualizar como é feita a interação do usuário com o servidor. O dispositivo B envia uma mensagem para o dispositivo A. A mensagem fica guardada no servidor, o dispositivo A verifica no servidor se existe uma mensagem. Se existir, o servidor envia a mensagem para o dispositivo A.



Figura 3.2.5: Representação do chat.

3.3 Classes da implementação do aplicativo

O software se conectará com o servidor por meio da internet. A implementação foi realizada sob a plataforma Windows e utilizando linguagem Java e servidor em PHP.

Foi necessário criar a tabela “dados” com os seguintes itens *id*, *nome*, *idioma*, *latitude*, *longitude*, *email*, *cidade*, *status* e *data*, a tabela “conversas” com *id_meu*, *id_usuario* e *codigo_mensagem* e a tabela “chat” com *id*, *id_send*, *id_recv*, *status_server*, *mensagem*, *nome*, *código_mensagem*, *nf* e *exclui*. Na tabela *dados* foram inseridos 5 usuários fictícios. Na Figura 3.3.1 pode-se visualizar a tabela “dados” com as informações do banco de dados.

	id	nome	idioma	latitude	longitude	email	cidade	status	data
Delete	368	Monica Mendes	ingles	-22.228974	-54.824823	MonicaMendes@uol.com	Dourados	0	2014-09-09 19:33:23
Delete	369	Lorena Oliveira	Portuguese, English Language	-22.223974	-54.824123	LorenaOliveira@uol.com	Campo Grande	0	2014-09-15 16:47:52
Delete	370	Zenaide Santos	Portugues	-22.226974	-54.825323	Zenaide@ig.com	Maracaju	0	2014-09-15 16:47:52
Delete	371	Pablo Henrique	Portugues, Espanhol	-22.2234737	-54.8245216	Pablo@yahoo.com.br	Cuiaba	0	2014-09-15 16:47:52
Delete	372	Sabrina Mendes	English Language	-22.2234237	-54.824323	Sabrina@uol.com	Dourados	0	2014-09-15 16:47:52
Delete	374	Maria Lucia Silva Oliveira	[Portuguese]	-22.2239585	-54.825323	marialuciajuara@hotmail.com	Juara,	0	2014-09-15 16:05:02
Delete	375	Luciano S. Oliveira	[Portuguese, English Language]	-22.2239804	-54.8243338	luciano_so@hotmail.com	Dourados	0	2014-09-15 18:02:05

Figura 3.3.1: Visualização dos dados pelo phpMyAdmin.

3.3.1 Modulo Aplicação

As principais classes que compõem a aplicação podem ser visualizadas na Tabela 3.3.1.

Classes	Objetivo
MainActivity	Classe de tela inicial, o usuário deve fazer o login
Painel_Principal	Classe principal onde o usuário poderá visualizar informações
IntegracaoFacebook	Classe de Integração efetua o login com o Facebook
Tela_com_Mapa	Classe responsável por exibir o mapa
Painel_Mensagens	Classe com todas as mensagens não visualizadas
Tela_de_chat_do_usuario	Classe responsável por efetuar conversa entre dois usuários
Tela_sobre_Painel_inicial	Classe com informações sobre o aplicativo
Tela_sobre_Painel_Mensagens	Classe com informações sobre painel de mensagens
Informacoes_Usuario_Selecionado	Classe responsável para exibir informações de usuário selecionado no mapa
Envia_Meus_dados_banco_dados	Classe responsável por enviar meus dados para o servidor online
Enviar_Mensagem_Botao_Chat	Classe que efetuar o envio das mensagens digitadas pelo usuário
Dados_Notificacao_Painel_Principal	Classe para efetuar notificação de novas mensagens
AnalisaID_Painel_Mensagem	Classe auxiliar para encontrar um usuário na classe painel mensagem
CustomListViewAdapter	Classe de customização da lista do Painel Mensagens
GlobalClass	Classe auxiliar para notificação
RowItem	Classe de estilo de lista
Tela_Sem_Internet	Classe para informar sobre falta de conexão
Usuario	Classe responsável por armazenar informações do usuário recebidas do banco de dados
UsuarioChat	Classe responsável por auxiliar na Painel Mensagens

Tabela 3.3.1: Classes do modulo da aplicação.

As classes *MainActivity*, *Painel_Principal*, *Tela_de_chat_do_usuario*, *Informacoes_Usuario_Selecionado*, *Tela_Sobre_Painel_Mensagens*, *Tela_sobre_Painel_inicial*, *Tela_Sem_Internet* e *Painel_Mensagens* estendem a classe *Activity*, permite que o usuário interaja com o aplicativo. A classe *Activity* é uma classe

gerenciadora de interface com o usuário, e todos os aplicativos começam com uma *Activity*. Quando o dispositivo não está conectado à internet é feita uma verificação de conectividade e uma mensagem é informada. O usuário receberá uma mensagem caso o dispositivo não esteja com o GPS habilitado

As classes *Envia_Meus_dados_banco_dados*, *Enviar_mensagem_Botao_Chat*, *Painel_Principal*, *Painel_Mensagens*, *Tela_com_Mapas* e *MainActivity* realizam o procedimento de enviar informações para o servidor online.

As classes *CustomListViewAdapter*, *RowItem* e *UsuarioChat* são utilizadas apenas para organização da *ListView*.

A classe *GlobalClass* é utilizada para auxiliar nas notificações.

3.3.2 Diagrama de caso de uso

A Figura 3.3.2 mostra o caso de uso do aplicativo. As interações do usuário dependem do mesmo efetuar login. Isto é feito na tela de login do aplicativo, a qual é mostrada assim que o aplicativo é iniciado. Somente pessoas que utilizam o Facebook poderão utilizar o sistema, já que o mesmo utiliza informações da conta do usuário no Facebook.

Após efetuar o login, o usuário pode efetuar uma busca de pessoas. Esta busca procura por pessoas que se cadastraram no aplicativo através do Facebook e que estão em um determinado raio de distância do usuário do aplicativo. Este raio é definido pelo próprio usuário.

Ao realizar uma busca, os usuários próximos serão mostrados em um mapa. A partir deste mapa é possível obter informações dos usuários próximos. No mapa pode-se escolher um usuário para estar visualizando mais informações e é possível iniciar uma conversa com este usuário.

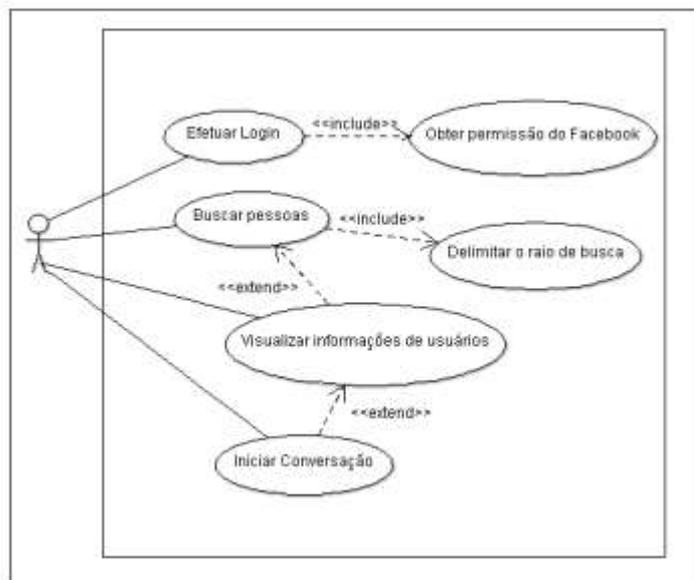


Figura 3.3.2: Diagrama de caso de uso do aplicativo.

3.3.3 Diagrama de Atividade

A Figura 3.3.3 mostra o diagrama de atividades do aplicativo. Nesta pode-se visualizar as interações que o usuário pode realizar.

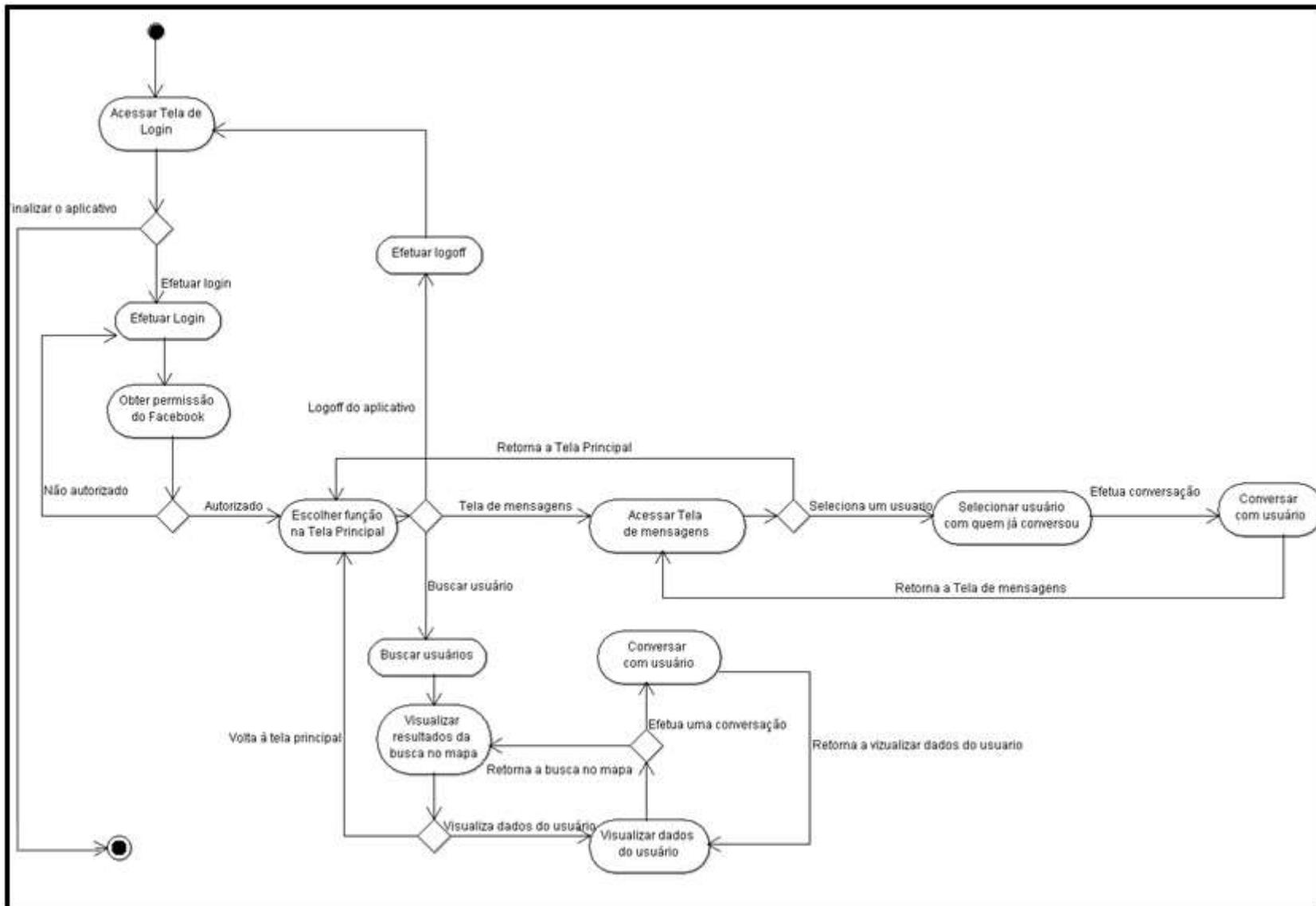


Figura 3.3.3: Diagrama de Atividade.

Capítulo 4

Resultados e Discussões

Os testes do software determinaram se a aplicação atingiu suas especificações e funcionou da maneira correta para o ambiente projetado. No teste foram utilizados dispositivos móveis e também foram feitas entradas manuais de dados diretamente no banco de dados.

4.1 Detalhes dos testes

Para realizar o *login* na aplicação o usuário deve clicar em *Registrar com o Facebook*, como pode ser observado na Figura 4.1.1



Figura 4.1.1: Tela inicial da aplicação.

Em seguida, a aplicação vai solicitar ao usuário para que permita que o Facebook disponibilize algumas informações. Pode-se ver a mensagem solicitando aceitar as permissões na Figura 4.1.2



Figura 4.1.2: Tela solicitando permissão.

Ao permitir a aplicação receber as informações, a tela principal será exibida com as informações de *nome*, *cidade*, *e-mail* e *idioma*, como pode ser visto na Figura 4.1.3



Figura 4.1.3: Tela principal da aplicação.

Para que possa ser feita a busca o usuário deve arrastar o *SeekBar* até a distância desejada entre 1 metro a 1.000 metros. Após escolher a distância basta tocar o botão Localizar. Com isto a tela com o mapa será exibida. Somente serão exibidos os usuários que estiveram *online* no dia atual, assim como mostra a Figura 4.1.4.



Figura 4.1.4: Tela mapa com marcador.

Caso o usuário deseje mudar o tipo do mapa ou atualizar os marcadores um menu oferece essa opção. O aplicativo oferece dois tipos de mapas: *Normal* e *Satellite*. Para mudar basta selecionar um dos dois. A Figura 4.1.5 mostra o menu.

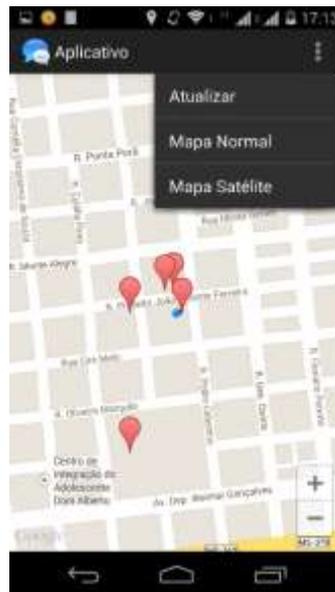


Figura 4.1.5: Tela mapa com menu.

Para visualizar a distância e o nome de um determinado usuário basta clicar em cima do marcador. Uma janela será exibida informando esses dados assim como mostra a Figura 4.1.6.



Figura 4.1.6: Tela mapa com janela no marcador.

Após a janela ser exibida no marcador, basta clicar na janela para que possa ser possível visualizar todas as informações disponíveis assim como mostra a Figura 4.1.7.



Figura 4.1.7: Tela de informações do usuário selecionado.

Para poder conversar com o usuário basta clicar no ícone Chat. A tela de chat será exibida logo em seguida. Basta iniciar a conversa. Este procedimento pode ser visto na Figura 4.1.8.

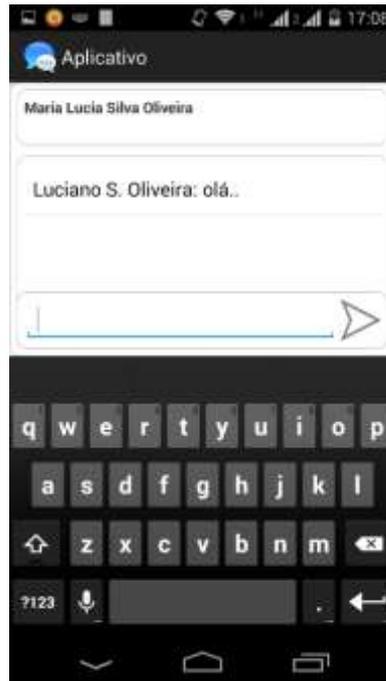


Figura 4.1.8: Tela de Chat.

Ao enviar uma mensagem para um determinado usuário uma notificação será exibida informando que existe uma mensagem, como mostra a Figura 4.1.9.

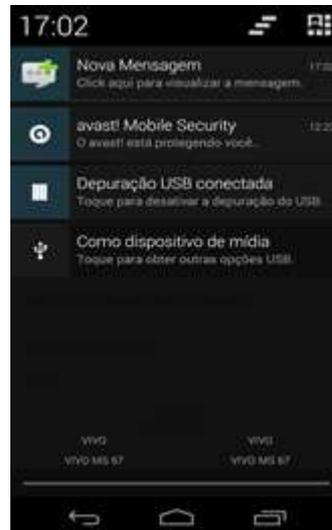


Figura 4.1.9: Tela de notificação informando nova mensagem.

Para visualizar a mensagem basta clicar na notificação e um painel será aberto basta selecionar o usuário que estiver com “new” para visualizar a mensagem, como demonstra a Figura 4.1.10.

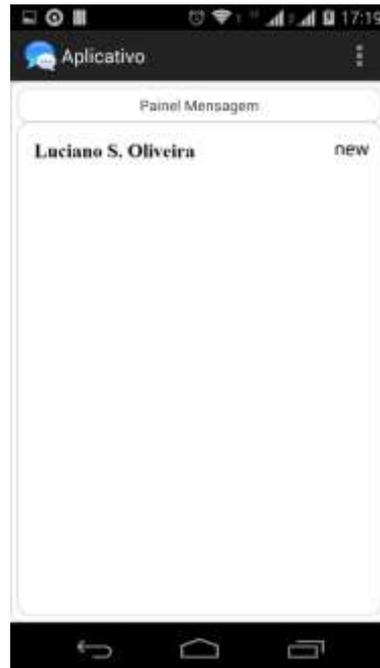


Figura 4.1.10: Tela painel de mensagens.

Ao clicar no usuário, uma tela de chat será exibida e assim é possível visualizar a mensagem recebida. As mensagens não são armazenadas, assim, o usuário somente terá uma única oportunidade de visualizar mas será possível conversar com o usuário outras vezes. Na Figura 4.1.11 é mostrada uma mensagem recebida e uma mensagem sendo enviada.



Figura 4.1.11: Tela de chat recebendo e enviando mensagens.

Na Figura 4.1.12 mostra o usuário recebendo a mensagem que foi enviada.



Figura 4.1.12: Tela usuário recebendo mensagem.

Caso o usuário já tenha enviado ou recebido alguma mensagem, um painel ficará disponível para mostrar todos os usuários com quem foram trocadas mensagens, assim como mostra na Figura 4.1.13



Figura 4.1.13: Menu com tela de mensagens.

Para acessar ao painel de mensagens clique em mensagens e uma nova tela será exibida, como mostra a Figura 4.1.14



Figura 4.1.14: Tela de Painel de Mensagens.

Caso o usuário queira excluir quem enviou uma mensagem do painel de mensagens, basta dar um longo clique ou em cima do usuário e clicar em deletar. Feito isto o usuário não será exibido mais no painel de mensagem, somente será exibido se receber uma nova mensagem dele ou enviar uma mensagem para ele. Isto pode ser visualizado na Figura 4.1.15.



Figura 4.1.15: Painel de mensagem, deletando usuário.

Capítulo 5

Conclusão

Hoje, se comunicar em inglês é indispensável para qualquer profissão. Uma grande empresa no momento da contratação dará preferência para um funcionário que entenda e fale o idioma.

Vivemos em uma era globalizada onde um trabalhador, independente da nacionalidade, pode atuar no exterior ou em contato direto com estrangeiros constantemente. Além disso, o idioma está presente no nosso cotidiano, seja nas músicas, filmes, TV a cabo, publicidade e principalmente na internet, que pode conectar pessoas espalhadas em diversos lugares do planeta para uma conversa ou reunião.

Para aprender e ganhar fluência em qualquer idioma o estudante deve sempre estar praticando, seja conversando, escrevendo ou lendo. É importante iniciar com um curso básico para aprender as regras verbais e a gramática e posteriormente pode: se inscrever em um curso de conversação para melhorar a fluência ou utilizar o aplicativo apresentado neste trabalho para localizar pessoas nas proximidades que também queiram praticar o inglês e melhorar a fluência.

O aplicativo apresentado neste trabalho foi testado e com ele foi possível localizar pessoas cadastradas no banco de dados da aplicação que falam outro idioma, através de informações obtidas pelo Facebook. Para encontrar as pessoas foi utilizado o GPS do dispositivo móvel. Diante dessas informações concluímos que este aplicativo é mais uma ferramenta que vem para auxiliar a interação entre as pessoas e a sociedade em geral.

5.1 Trabalhos futuros

Nesta seção serão apresentadas algumas possíveis melhorias que podem ser agregadas e estudadas mais adiante.

- Sistema de banco de dados SQLite

Implementar o sistema com banco de dados SQLite para armazenar as mensagens trocadas pelos usuários.

- Correção de possíveis erros.

Correção de possíveis erros do aplicativo.

- Disponibilizar foto

Adicionar função para mostrar a foto de usuários selecionados.

- Criptografia das mensagens trocadas

Criptografar todas as mensagens e informações do usuário.

Referências Bibliográficas

- [1] ANDROID-DEVELOPERS. Android, The world's most popular mobile platform. Disponível em <<http://developer.android.com/about/index.html>> Acessado em 25 jul. 2014.
- [2] ANDROID-UOL. Sistema operacional do Google conquista espaço no mercado de smartphones. Disponível em <<http://android.uol.com.br/o-que-e-google-android.jhtm>> Acessado em 07 maio 2014.
- [3] AND-DEV-VIS. Android Developer Visibility for Your Apps, 2013. Disponível em <<http://developer.android.com/distribute/googleplay/about/visibility.html>> Acessado em 25 jul. 2014.
- [4] BORDIN, M. V. Introdução a Arquitetura Android, Faculdade de Sistema de Informação, Sociedade Educacional Três de Maio, 2012. Disponível em <<http://sites.setrem.com.br/stin/2012/anais/Maycon.pdf>> Acessado em 09 maio 2014.
- [5] CORREIA, A. G. S. Aplicações e Serviços Baseados em Localização, Pontifícia Universidade Católica do Rio de Janeiro, 2004. Disponível em <<http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/04/AdolfoCorreia-Mono.pdf>> Acessado em 14 maio 2014.
- [6] DEVELOPERWORKS. Explorando Funcionalidades das APIs do Facebook, 2011. Disponível em <<http://imasters.com.br/artigo/20274/apis/explorando-funcionalidades-das-apis-do-facebook/>> Acessado em 23 de set. 2014.
- [7] FIGUEIREDO, M. S.; NAKAMURA, E. Computação Móvel: Novas Oportunidades e Novos Desafios. T&C Amazônia, Junho de 2003.
- [8] FUTURO, M. A era da Computação Móvel, 2014. Disponível em <<http://marketingfuturo.com/a-era-da-computacao-movel/>> Acessado em 08 maio 2014.
- [9] GARGENTA, M. Learning Android. O'Reilly Media, 2011.
- [10] GOOGLE-DEVELOPERS. Google Maps Android API V2. Disponível em <<https://developers.google.com/maps/documentation/android/map?hl=pt-br>> Acessado em 23 set. 2014
- [11] GUEDES, E. M. P. Estudo de Técnicas Híbrida de Localização de Estações Móveis Baseada em TDoA e AoA, Instituto Militar de Engenharia, 2003. Disponível em <http://www.pgee.ime.eb.br/pdf/edna_guedes.pdf> Acessado 30 jul.2014
- [12] HAIGES, S. The location API, 2003. Disponível em <<http://java.sys-con.com/node/37747>> Acessado 14 maio 2014.
- [13] HARDT, D. The OAuth 2.0 Authorization Framework draft-ietf-oauth-v2-31, 2012. Disponível em <<https://tools.ietf.org/html/draft-ietf-oauth-v2-31>> Acessado 23 set. 2014.

- [14] JOHNSON, T. M. Métodos de Localização Baseados no Cell-ID. Disponível em <http://www.wirelessbrasil.org/wirelessbr/colaboradores/thienne_johnson/cell_id/cell_id.html> Acessado em 27 jul. 2014.
- [15] JSON. Introdução ao JSON. Disponível em <<http://json.org/json-pt.html>> Acessado em 13 outubro 2014.
- [16] LI, M. Facebook SDK for Android. Disponível em <<https://github.com/facebook/facebook-android-sdk>> Acessado em 23 set. 2014.
- [17] MORENO, A. C. Brasil perde oportunidades por falta de domínio do inglês, diz especialista, 2012. Disponível em <<http://g1.globo.com/educacao/noticia/2012/07/brasil-perde-oportunidades-por-falta-de-dominio-do-ingles-diz-especialista.html>> Acessado em 07 maio 2014;
- [18] MORIMOTO, C. E. Entendo o A-GPS, 2009. Disponível em <<http://www.hardware.com.br/livros/smartphones/entendendo-gps.html>> Acessado em 25 jul. 2014.
- [19] OHA-AND-REV. Android Review, 2007. Disponível em <http://www.openhandsetalliance.com/android_overview.html> Acessado em 08 maio 2014.
- [20] RABELLO, R. R. Android: um novo paradigma de desenvolvimento. Disponível em <http://www.cesar.org.br/site/files/file/WM18_Android.pdf> Acessado em 08 maio 2014.
- [21] ROCHA, M. N. Computação Móvel: Serviços Baseados em Localização, Universidade Federal de Viçosa, 2002. Disponível em <<http://www.ufv.br/nugeo/ufvgeo2002/resumos/mnacif.pdf>> Acessado em 09 maio 2014.
- [22] SANTANA, R. C. Computação móvel, histórico da evolução, Universidade de São Paulo 2008. Disponível em <<http://grenoble.ime.usp.br/~gold/cursos/2008/movel/mono/HistoricoComputacaoMovel.pdf>> Acessado em 08 maio 2014.
- [23] SILVA, L. A. D. Apostila de Android, 4ª ed, 2012. Disponível em <http://www.agenciadream.com/uploads/download/download_bf2f5d7803d78d0de48d133a0aa450dd.pdf> Acessado 25 jul. 2014.
- [24] SHEJWAL, M. Android Architecture. Disponível em <http://www.tutorialspoint.com/android/android_architecture.htm> Acessado 25 jul. 2014.
- [25] TOSIN, C.E.G. A Plataforma Android. Disponível em <http://www.softblue.com.br/blog/home/postid/11/CONHECENDO_O_ANDROID> Acessado em 09 maio 2014.
- [26] VELOSO, T. Google Android: Conheça todos os nomes, sabores e versões da plataforma móvel, 2010. Disponível em <<https://tecnoblog.net/56850/google-android-versoes-nomes-sabores/>> Acessado em 27 jul. 2014.

Apêndice A

Instalação e configuração do Eclipse

1. Entre na página oficial de *download* do Eclipse, escolha um dos pacotes Eclipse Standard 4.4 ou Eclipse IDE for *Java Developers*, em seguida escolha a versão do seu sistema operacional (32 ou 64 bits).

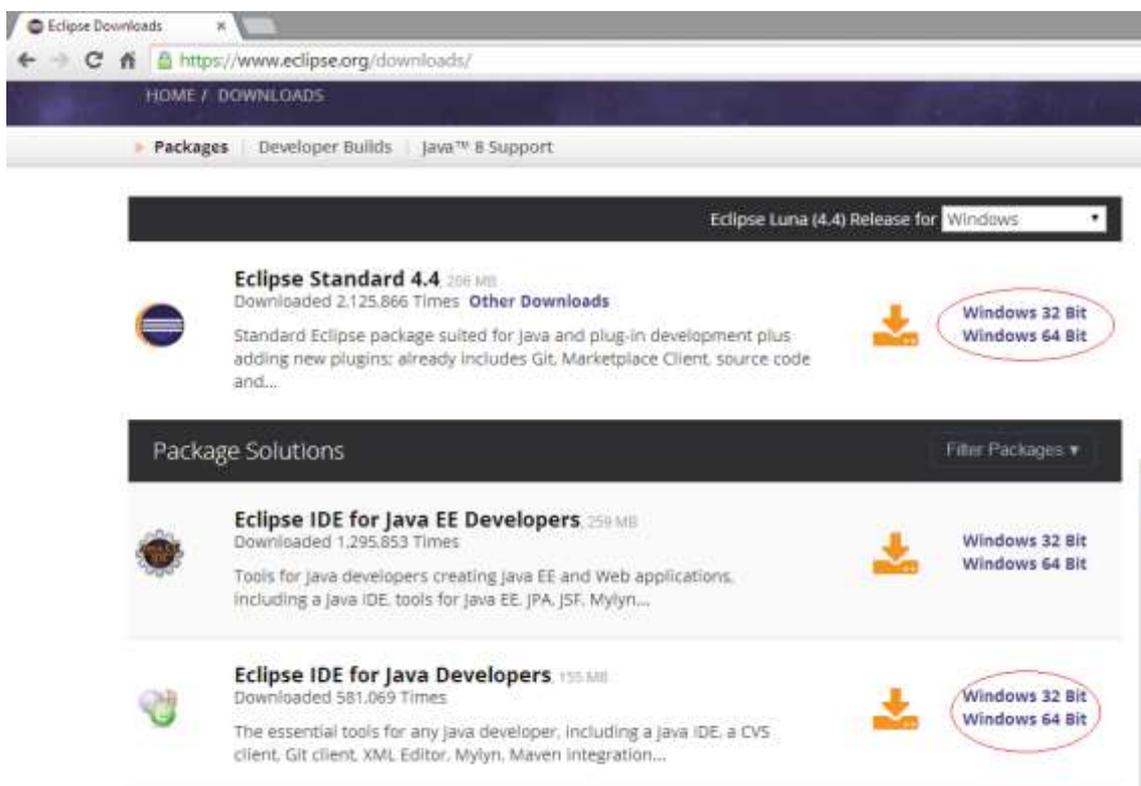


Figura A. 1: Página oficial do Eclipse.

2. Após realizar o *download*, descompacte a pasta em um lugar a sua escolha, o Eclipse não precisa ser instalado, então abre a pasta onde foi descompactado e execute o eclipse.exe.

Apêndice B

Instalação das APIs

B.1 Facebook API

1. Entre na página Oficial de download do Facebook Developers e faça o download do SDK Como podemos visualizar na Figura B.1.1.

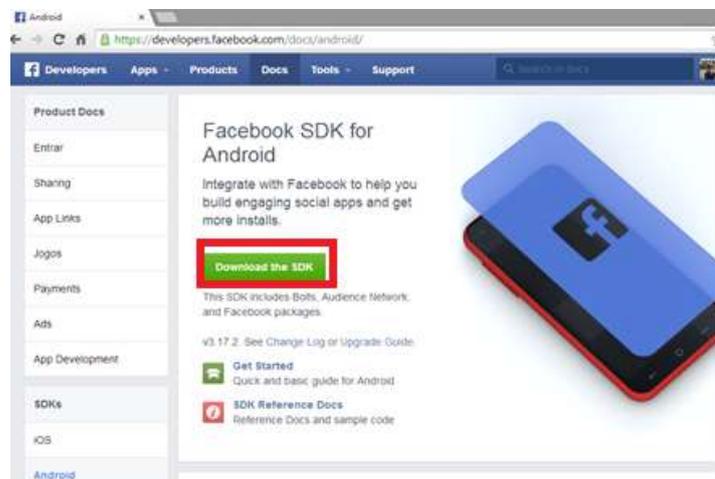


Figura B.1.1: Pagina oficial Facebook Developers.

3. Após o *download*, descompacte o arquivo em diretório a sua escolha.
4. Abre o Eclipse IDE.
5. Click em *File*.
6. Click em *Import*
7. Selecione "*Existing Android Code into Workspace*" e click em "*Next*", podemos visualizar na Figura B.1.2

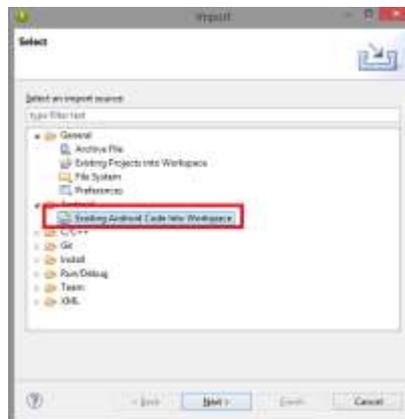


Figura B.1.2: Tela de Import do Eclipse.

7. Clique em “Browse” e localize a pasta onde descompactou o arquivo, por fim clique em “Finish”. Podemos ver na Figura B.1.3.

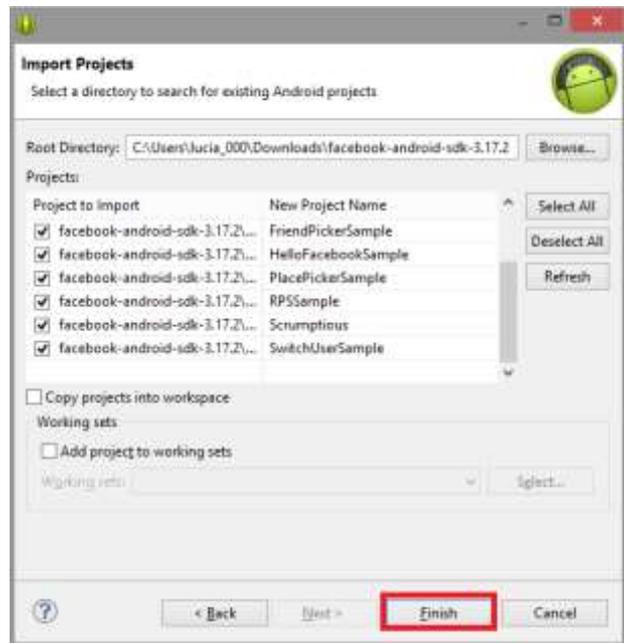


Figura B.1.3: Import API.

8. Feito todos procedimentos o Eclipse está pronto para utilizar a API do Facebook.

9. Agora deve criar um aplicativo no site oficial do Facebook para que a aplicação possa estar se conectando. Abra novamente o site oficial do Facebook developers e clique em “Apps” e depois em “Create a New App”. Podemos visualizar na Figura B.1.4.



Figura B.1.4: Criando um aplicativo no site do Facebook.

10. Após, preencha todas as informações que foi solicitado. Figura B.1.5

Create a New App
Get started integrating Facebook into your app or website.

Display Name
[The name of your app or website]

Namespace
A unique identifier for your app (optional)

Is this a test version of another app? Saiba mais.

Categoria
Escolha uma categoria ▾

By proceeding, you agree to the Facebook Platform Policies

Figura B.1.5: Dados para criar um aplicativo.

11. Em seguida você vai precisar associar sua chave *Android Hash* com o aplicativo. O *Hash* é uma chave utilizada pelo Facebook como uma verificação de segurança para autenticar. Vá para Settings no painel à esquerda, click no botão “+ *Add Platform*” e selecione *Android*.

12. A forma mais fácil de obter a chave *Hash* é utilizando o código na Figura abaixo B.1.6

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Add code to print out the key hash
    try {
        PackageInfo info = getPackageManager().getPackageInfo(
            "com.facebook.samples.hellofacebook",
            PackageManager.GET_SIGNATURES);
        for (Signature signature : info.signatures) {
            MessageDigest md = MessageDigest.getInstance("SHA");
            md.update(signature.toByteArray());
            Log.d("KeyHash:", Base64.encodeToString(md.digest(), Base64.DEFAULT));
        }
    } catch (NameNotFoundException e) {

    } catch (NoSuchAlgorithmException e) {

    }

    ...
}
```

Figura B.1.6: Código para adquirir chave Hash. Fonte: Facebook developers.

13. Crie uma aplicação e coloque o código citado na Figura B.16, execute a aplicação e vai estar disponibilizando a chave *Hash*, após vá ao site e coloque o a chave *Hash* e click em salvar. Podemos visualizar na Figura B.1.7. Após isto você já está apto a utilizar a API do Facebook.

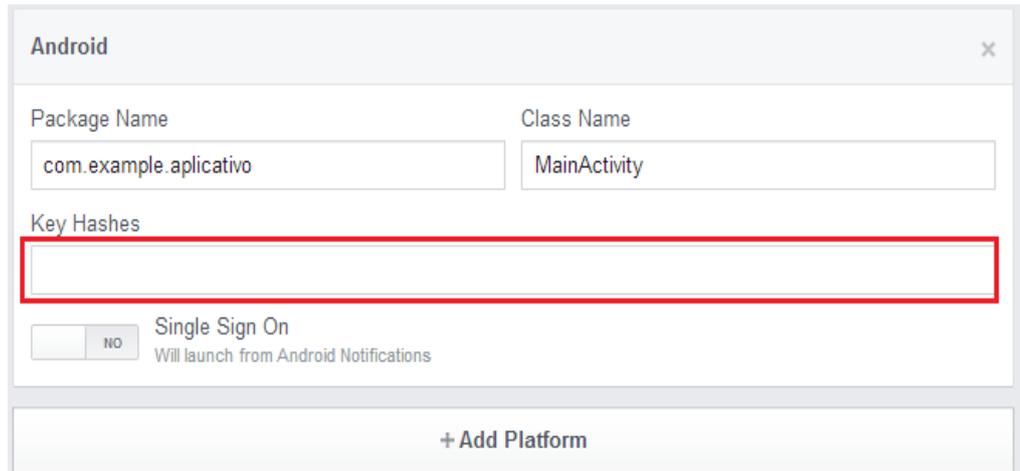


Figura B.1.7: Inserindo chave Hash.

B.2 Google Maps API

1. Para utilizar o Google Maps API Android, deve-se selecionar no menu do Eclipse Windows > Android SDK Manager > Extras > Google Play Services, assim como mostra a Figura B.2.1

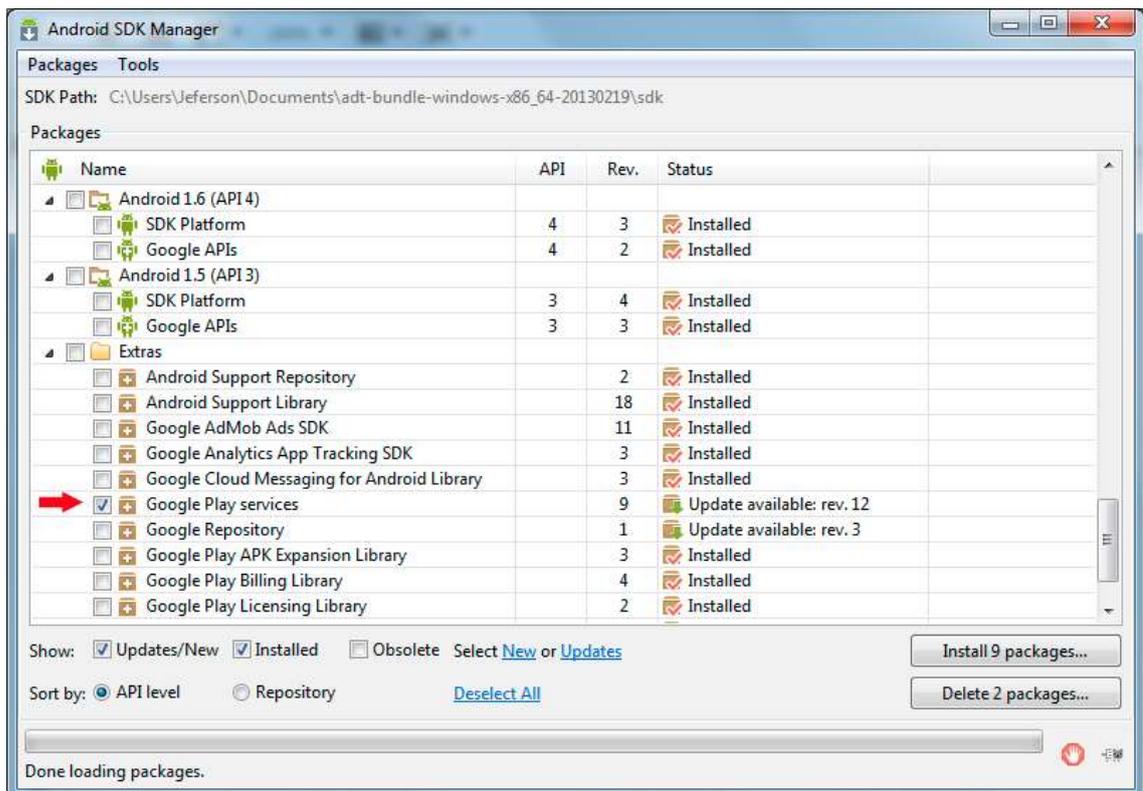


Figura B.2.1: Baixando Google Play Services.

2. Para podermos utilizar o Google Maps, precisa-se de uma chave. Por padrão, ela é armazenada no mesmo diretório do seu dispositivo AVD:

3. Abra um Terminal do Windows (cmd) e navegue até a pasta “bin” do jdk (C:\Program Files\Java\jre7\bin), e execute a seguinte linha de comando

```
keytool -list -v -keystore "C:\Users\”USER”\android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

Agora copie a SHA-1, podemos visualizar na Figura B.2.2

```
C:\Program Files\Java\jre8\bin>keytool -list -v -keystore "C:\Users\lucia_000\android\debug.keystore" -alias androiddebugkey -storepass android -keypass android:
Nome do alias: androiddebugkey
Data de criação: 06/08/2014
Tipo de entrada: PrivateKeyEntry
Comprimento da cadeia de certificados: 1
Certificado[1]:
Proprietário: CN=Android Debug, O=Android, C=US
Emissor: CN=Android Debug, O=Android, C=US
Número de série: 1f2caaea
Válido de: Wed Aug 06 15:48:33 BRT 2014 a: Fri Jul 29 15:48:33 BRT 2044
Fingerprints do certificado:
MD5: 1F:AE:BC:BF:A8:D9:29:41:01:64:21:41:10:70:07:13
SHA1: 59:0E:64:33:AC:7A:BF:B0:4A:D5:ED:10:50:11:ED:1B:01:EE:50:01
SHA256: 6A:83:EC:1C:53:A1:5E:80:CB:56:F4:47:17:20:21:0D:71:00:ED:70:07:
95:7E:05:F0:AA:39:74:D4:4C:59:62
Nome do algoritmo de assinatura: SHA256withRSA
Versão: 3

Extensões:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B4 3E A9 ED 53 28 EA A6 F0 CC BC 07 9C 47 41 34 .>..S<.....GA4
0010: 92 30 2B 00 .0+.
]
]
]
```

Figura B.2.2: Obtendo chave SHA-1.

4. Assim que obter a chave, devemos registrar no site da API do Google, acessando o seguinte link: <https://code.google.com/apis/console/>, será necessário estar logado com uma conta de Gmail para utilizar, no lado esquerdo click em APIs e localize Google Maps Android API V2, e mude para ON, podemos visualizar na Figura B.2.3.

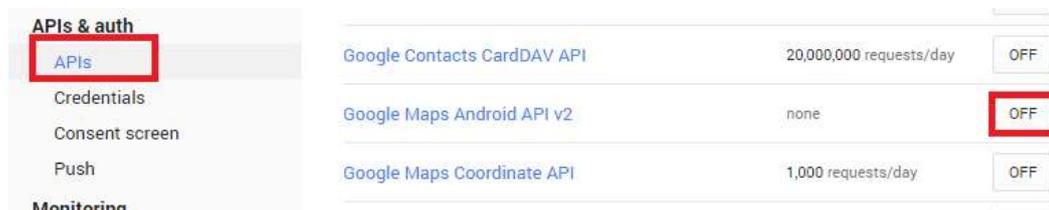


Figura B.2.3 Habilitando Google Maps Android API V2.

5. A seguir click em “Credentials” e depois em “Create new key”, um pop-up será exibido, click em “Android key”, depois insira a chave SHA-1 de seu certificado, Figura B.2.4

ACCEPT REQUESTS FROM AN ANDROID APPLICATION WITH ONE OF THE CERTIFICATE FINGERPRINTS AND PACKAGE NAMES LISTED BELOW

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

Create

Cancel

Figura B.2.4: Inserir chave SHA-1.

6. Próximo passo é copiar a API Key disponibilizada pelo Google play, para ser utilizada na aplicação, assim como mostra a Figura B.2.5.

Key for Android applications

API KEY	AlzaSyDBLRJ1quRKuWKN_6s_vmT00vWmca_uq
ANDROID APPLICATIONS	59:0E:64:33:AC:7A:BF:B0:4A:D5:EB:18:3C:11:EB:4B:01:EE:58:84;com.example.aplicativo
ACTIVATION DATE	Mar 26, 2014 12:47 AM
ACTIVATED BY	carolinejarama@gmail.com (you)

Edit allowed Android applications Regenerate key Delete

Figura B.2.5: Obtendo API Key.

7. Após seguir todos os procedimentos, já pode utilizar a API do Google Maps.