

**UNIVERSIDADE ESTADUAL PAULISTA  
FACULDADE DE CIÊNCIAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**KINDI ZHUO MING WEI**

**LIBRAS TRANSLATOR: COMUNICADOR EM LIBRAS PARA DEFICIENTES  
AUDITIVOS**

**BAURU - SP**

**2016**

KINDI ZHUO MING WEI

**LIBRAS TRANSLATOR: COMUNICADOR EM LIBRAS PARA DEFICIENTES  
AUDITIVOS**

Monografia para Trabalho de Conclusão de  
Curso do Curso de Bacharelado em Ciência da  
Computação da Universidade Estadual Paulista  
“Júlio de Mesquita Filho”, Faculdade de  
Ciências, campus Bauru.

Orientador: Dr. João Paulo Papa

BAURU - SP

2016

Wei, Kindi Zhuo Ming.  
Comunicador em libras para deficientes  
auditivos  
42 f. : il.

Orientador: João Paulo Papa

Monografia (Graduação) - Universidade  
Estadual Paulista. Faculdade de  
Ciências, Bauru, 2016

Aplicação Mobile 2. Tradutor 3.  
Libras. I. Universidade Estadual  
Paulista. Faculdade de Ciências. II.  
Libras Translator: Comunicador em  
libras para deficientes auditivos

KINDI ZHUO MING WEI

**LIBRAS TRANSLATOR: COMUNICADOR EM LIBRAS PARA DEFICIENTES  
AUDITIVOS**

Monografia para Trabalho de Conclusão de  
Curso do Curso de Bacharelado em Ciência da  
Computação da Universidade Estadual Paulista  
“Júlio de Mesquita Filho”, Faculdade de  
Ciências, campus Bauru.

**BANCA EXAMINADORA**

Prof. Dr. João Paulo Papa  
DCo – FC – UNESP Bauru  
Orientador

Prof.<sup>a</sup> Dr.<sup>a</sup> Simone das Graças Domingues Prado  
DCo – FC – UNESP Bauru

Prof. Dr. Kelton Augusto Pontara da Costa  
DCo – FC – UNESP Bauru

Bauru, 7 de fevereiro de 2017

BAURU - SP

2016

## **AGRADECIMENTOS**

Primeiramente agradeço meus pais por todo apoio e suporte à toda minha jornada na faculdade e por toda minha vida independentemente dos problemas pessoais da surdez que possuem, nunca deixaram de me apoiar.

Agradeço à minha namorada que sempre me apoiou em todas as minhas decisões em minha vida acadêmica e profissional.

Agradeço também ao Prof. Dr. João Paulo Papa por orientar e alinhar ideias que foram cruciais no desenvolvimento deste projeto.

Agradeço imensamente a Deus que é o que nos move e nos dá forças. Nos deu família e amores para nos dar suporte sempre.

## RESUMO

Deficientes Auditivos têm o sério problema de comunicação com qualquer interlocutor. Porém ainda há surdos, como gostam de ser chamados, que não tiveram algum tipo de educação privilegiada por não terem alguma ajuda educacional.

Com isso, nesta monografia apresento um aplicativo que possa ajudar na comunicação destes deficientes. Este trabalho expõe como foram implementações, testes e conclusões do aplicativo criado para tentar auxiliar este nicho de deficientes específico.

**Palavras-chave:** LIBRAS, Língua Brasileira de Sinais, Tradutor, Deficiente Auditivo, Surdos.

## **ABSTRACT**

Hearing impaired have the serious problem about communication with any interlocutor. However still have some deaf, as they like be to called, who did not have some kind of education because they do not have any educational help.

Therefore, this monography presentes an application that can help the communication of the deficientes. We show the tests and conclusion of the application created to try help that specific niche of hearing impaired.

**Keywords:** LIBRAS, Brazilian Language of Signals, Translator, Hearing Impaired, Deaf.

## LISTA DE FIGURAS

FIGURA 1 – MOTOROLA DYNATAC 8000X .....	18
FIGURA 2 – MOTOROLA PT-550 .....	18
FIGURA 3 – ARQUITETURA ANDROID .....	20
FIGURA 4 – ARQUITETURA IOS .....	21
FIGURA 5 – ARQUITETURA DO WINDOWS PHONE .....	22
FIGURA 6 – ALFABETO EM LIBRAS .....	25
FIGURA 7 – FUNCIONAMENTO DO APLICATIVO .....	27
FIGURA 8 – PROTÓTIPO DO APLICATIVO .....	28
FIGURA 9 – IDE ANDROID STUDIO. AMBIENTE DE DESENVOLVIMENTO .....	29
FIGURA 10 – LIBRAS TRANSLATOR, TELAS DE BOAS VINDAS .....	32
FIGURA 11 – TELA DE ESCOLHA DE CONTEXTO.....	33
FIGURA 12 – TELA DO CONTEXTO ASSUNTO .....	34
FIGURA 13 – TELA DE FRASES .....	34
FIGURA 14 – TELA DE PALAVRAS PARA MONTAGEM DE FRASES .....	14

## **LISTA DE QUADROS**

QUADRO 1 – QUADRO COMPARATIVO ANDROID X IOS .....	24
---	----

## **LISTA DE CÓDIGOS E ALGORÍTMOS**

CÓDIGO 1 – GIFVIEW.JAVA. IMPLEMENTAÇÃO DE CLASSE PARA RODAR GIFS .....	18
CÓDIGO 2 – MAINACTIVITY.JAVA. ARQUIVO PRINCIPAL DE ATIVIDADES DO ANDROID. ....	19
CÓDIGO 3 – ACTIVITY_MAIN.XML. ARQUIVO DE LAYOUT DO ANDROID. ....	20

## **LISTA DE ABREVIACOES E SIGLAS**

<b>3G</b>	Terceira Gerao
<b>4G</b>	Quarta Gerao
<b>GIF</b>	Graphics Interchange Format
<b>IDE</b>	Integrated Development Environment
<b>iOS</b>	iPhone Operational System
<b>LIBRAS</b>	Lngua Brasileira de Sinais
<b>MAC</b>	Macintosh
<b>OS</b>	Operational System
<b>PDA</b>	Personal Digital Assistants
<b>PC</b>	Personal Computer
<b>SMS</b>	Short Message Service
<b>SO</b>	Sistema Operacional
<b>SDK</b>	Software Development Kit
<b>TV</b>	Televiso
<b>XML</b>	Extensible Markup Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>13</b>
1.2	PROBLEMA .....	14
1.3	JUSTIFICATIVA .....	15
<b>2.1</b>	<b>OBJETIVO GERAL .....</b>	<b>15</b>
<b>2.2</b>	<b>OBJETIVOS ESPECÍFICOS.....</b>	<b>15</b>
<b>3</b>	<b>ESTRUTURA DO PROJETO.....</b>	<b>15</b>
<b>4</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>17</b>
4.1	SISTEMAS OPERACIONAIS.....	17
4.2	SMARTPHONES E SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS 17	
4.2.1	<i>HISTÓRICO</i> .....	17
4.2.2	<i>ANDROID</i> .....	19
4.2.3	<i>iOS</i> .....	20
4.2.4	<i>WINDOWS PHONE</i> .....	21
4.3	DESENVOLVIMENTO <i>MOBILE</i> .....	23
<b>5</b>	<b>LIBRAS .....</b>	<b>24</b>
<b>6</b>	<b>MATERIAIS E MÉTODOS.....</b>	<b>27</b>
<b>7</b>	<b>PROBLEMÁTICAS E DIFICULDADES.....</b>	<b>30</b>
<b>8</b>	<b>VIABILIDADE.....</b>	<b>31</b>
<b>9</b>	<b>APLICAÇÃO.....</b>	<b>32</b>
9.1	CONTEXTO: ASSUNTO .....	33
9.2	CONTEXTO: PALAVRAS.....	34
<b>9</b>	<b>TESTES.....</b>	<b>14</b>
<b>10</b>	<b>CONCLUSÕES.....</b>	<b>15</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>16</b>
	<b>ANEXO A – IMPLEMENTAÇÃO DE GIF EM ANDROID.....</b>	<b>18</b>

# 1 INTRODUÇÃO

A deficiência auditiva, ou surdez comumente falado, é um fato que por muitas vezes é passado despercebido por todos. Historicamente, as escolas e internatos da Europa foram os primórdios da comunicação em sinais, e quando alguém era pego fazendo-os, eram punidos severamente. A primeira educação para surdos ocorreu com Pedro Ponce De Leon, também na Europa, mas somente para os filhos da nobreza. Pedro Ponce era monge e iniciou, na Espanha, a primeira escola para surdos em um monastério. Ensinava latim, grego, italiano, física e astronomia. Antes deste primeiro ensino aos surdos, esses eram considerados incapazes de se educar, ficando à margem da sociedade. Após um século (em 1756), L'Epeé, na França, criou a primeira escola para surdos, onde defendia a Língua de Sinais como linguagem natural dos surdos e que fosse considerada como uma forma de comunicação como qualquer outra língua.

O Brasil deu início a oficialização da linguagem de sinais com Eduard Huet, professor surdo, mestre em Paris, e que veio aos cuidados de Dom Pedro II e que, antes dele, os surdos eram sempre esquecidos em relação à educação. Assim sendo, o referido professor teve a intenção de criar uma escola aos moldes de uma escola da Europa de educação aos surdos.

Os primeiros passos da LIBRAS (Língua Brasileira de Sinais) no Brasil deu-se com o alfabeto manual de origem Francesa, e assim difundido pelos próprios surdos advindos de outros lugares do país. Assim, em 26 de setembro de 1857, fundou-se no Rio de Janeiro a primeira escola para surdos do Brasil, sendo que hoje esta data é comemorada o dia nacional dos Surdos no Brasil (Círculo Fluminense de Estudos Filológicos e Linguísticos, 2012).

Com todos os conhecimentos levados por anos e sendo criada uma base sólida da LIBRAS no país, hoje tem-se a Federação Nacional de Educação e Integração dos Surdos (FENEIS), conforme destacado abaixo.

Em 1977, foi fundada a Federação Nacional de Educação e Integração dos Deficientes Auditivos-FENEIDA. A representatividade dos surdos estava comprometida, pois a nova entidade era composta apenas por pessoas ouvintes. Como resposta a essa exclusão, em 1983, a Comunidade Surda criou uma Comissão de Luta pelos Direitos dos Surdos, um grupo não oficializado, mas com um trabalho significativo na busca de participação nas decisões da diretoria da FENEIDA. Até então esse direito lhes era negado por não se acreditar na capacidade de que poderiam coordenar uma entidade. Mas, não demorou muito e devido à grande

credibilidade adquirida, a Comissão conquistou a presidência da FENEIDA. Em 16 de maio de 1987, na Assembleia Geral, a nova diretoria reestruturou o estatuto da instituição, que passou a se chamar Federação Nacional de Educação e Integração dos Surdos-FENEIS, com sedes regionais espalhadas pelo Brasil, e a matriz no Rio de Janeiro. (Federação Nacional de Educação e Integração dos Surdos, 2016)

## 1.2 PROBLEMA

Atualmente os surdos possuem uma grande quantidade de tecnologia (aplicativos, dispositivos) disponível. Entretanto, seus criadores não pensaram na real necessidade e acessibilidade que tais dispositivos deveriam ter. Aos poucos, esse paradigma vem sendo quebrado e os programadores e educadores estão se adaptando à real necessidade das pessoas com deficiência auditiva. Porém, comunicação ainda é um grande problema. A tecnologia existe, mas não é bem utilizada, ou não é compreendida ou não é sabida por eles.

Uma simples consulta médica ainda é um problema para os deficientes auditivos devido à falta de compreensão da LIBRAS. Uma tentativa de resolução de problemas no banco ainda é um problema. Infelizmente, são raras as pessoas que estão capacitadas em atender e compreender os problemas que os surdos precisam resolver, sendo sempre a falta de comunicação como empecilho. Não há pessoas capacitadas para a comunicação e assim entender o problema a ser solucionado.

Existem no mercado facilitadores de comunicação para que surdos possam ter a facilidade de comunicação com a sociedade e que possam ajuda-los, porém ainda não é compreendido e não é transmitido esse conhecimento a eles. Ainda há grande parte da população surda que viveu a margem da sociedade e que não teve a oportunidade de uma boa educação ou até mesmo nenhuma. Estes têm dificuldades de aprendizado dessas aplicações, dispositivos existentes e até mesmo de qualquer tipo de aprendizagem do conhecimento. Desta forma, essas pessoas não compreendem muito bem o português escrito pela sociedade, aumentando ainda mais a barreira social.

Assim sendo, o principal problema a ser atacado por esta proposta é o fato de muitas pessoas com deficiência auditiva não saberem a língua Portuguesa. Assim, a comunidade carece de um **interpretador** que possa traduzir a língua escrita para a linguagem de sinais, e vice-versa.

### **1.3 JUSTIFICATIVA**

Os problemas sempre enfrentados pelos surdos na sociedade e a necessidade de maior independência destes deficientes devem ser levados em consideração. O aproveitamento da tecnologia para resolução de problemas é comum, e agora deve-se aliar a necessidade de resolução deste problema de comunicação com a grande oferta de tecnologia que há atualmente. Produtos já existentes traduzem o português para a linguagem de sinais, porém nenhum traduz dos sinais para o português. Assim, nenhum atende todo o público alvo.

### **2.1 OBJETIVO GERAL**

Construir um sistema que interprete o que o surdo, locutor de uma conversa, deseja através da escolha de imagens no aplicativo, monte uma frase ou texto e o transmita para quem esteja tentando se comunicar, interlocutor. Assim sendo, a presente proposta de trabalho de conclusão de curso visa o estudo e implementação de um sistema que possa traduzir frases criadas em LIBRAS para a linguagem Portuguesa. Conforme dito anteriormente, existem muitos deficientes auditivos que não conhecem o português, o que dificulta ainda mais a sua comunicação com as pessoas, pois os deficientes não conseguem, pelo menos, escrever a sua frase para poder se comunicar com alguém que não tenha conhecimento da linguagem LIBRAS.

### **2.2 OBJETIVOS ESPECÍFICOS**

Desenvolver um aplicativo em Android utilizando os estudos sobre o armazenamento de arquivos em Android, tipos de Banco de Dados e bibliotecas, fazendo a comunicação do Banco com a aplicação.

## **3 ESTRUTURA DO PROJETO**

A aplicação para *smartphones* funciona como um tradutor em “tempo real” do que o deficiente auditivo passará ao aplicativo, ou seja, palavras, letras ou números (imagens, GIFs) em modelo LIBRAS. Como resultado final, o aplicativo retorna a frase em português. Tal

tarefa envolve o armazenamento de imagens, GIFs, Banco de Dados em dispositivos móveis (Android) e bibliotecas.

Primeiramente foi feito o levantamento de dados com deficientes auditivos para entender as reais necessidades e dificuldades perante a sociedade.

Em seguida compreendeu-se o que podia ser feito para ser elaborado e implementado na aplicação para que fosse ser útil para o deficiente auditivo.

Posteriormente, estudou-se as tecnologias existentes de Banco de Dados para dispositivos móveis e também armazenamento de arquivos em na plataforma Android. Compreender a capacidade do dispositivo móvel, analisar as suas limitações e pequenos testes de aplicação também foram realizados nessa etapa.

Após estes passos inicializou-se a programação do aplicativo com o conjunto de todos os dados recolhidos e estudados.

Em sequência foram feitas a programação e implementação dos últimos detalhes e testes do aplicativo com um fluente. Por fim foram realizados últimos testes com fluentes e deficientes auditivos.

## **4 FUNDAMENTAÇÃO TEÓRICA**

### **4.1 SISTEMAS OPERACIONAIS**

Sistema operacional (S.O.) tem o trabalho de fornecer aos programas do usuário um modelo melhor, mais simples e mais limpo e lidar com o gerenciamento de todos os recursos de hardware (Tanenbaum, 2008).

Para Cardoso, Magalhães e Faina (2002, p. 2) é difícil, para a maioria dos usuários, definir o que é um sistema operacional e que parte do problema decorre do fato do sistema operacional realizar duas funções básicas sendo que uma se destaca sobre a outra: Sistema operacional como uma máquina virtual onde o S.O. trabalha com abstrações de mais alto nível onde detalhes de implementação destas abstrações não são visíveis, como acesso as operações de entrada e saída. Já a outra função, Sistema operacional como um Gerenciador de Recursos, o S.O. tem a função de gerenciar a forma adequada os recursos de hardware (discos, memórias, processadores, interface de rede etc.).

Historicamente o S.O. passou pela primeira geração (1945-1955) onde trabalhavam com válvulas e plugs. Passaram por transistores e processamento em batch (1955-1965), circuitos integrados e multiprogramação (1965-1980) e o que ainda existem hoje a quarta geração de sistemas operacionais, computadores pessoais e estações de trabalho. Porém, com o surgimento dos smartphones, estes sistemas operacionais podem ser considerados

### **4.2 SMARTPHONES E SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS**

#### **4.2.1 HISTÓRICO**

Os celulares foram imaginados pelo seu inventor, Martin Cooper, considerado o pai do celular, em abril de 1973 e fabricados pela Motorola primeiramente para portabilidade, para poder levar um telefone a qualquer lugar e, claro, para comunicação simples de voz (o primeiro celular fabricado foi o DynaTAC 8000x da Motorola). Eram utilizados única e exclusivamente para voz. A figura 1 o primeiro celular da história.

Figura 1 – Motorola DynaTAC 8000x



Fonte: Site Vintage Mobile Phones<sup>1</sup>

No Brasil o primeiro celular lançado no país foi o Motorola PT-550 que foi vendido primeiramente no Rio de Janeiro e depois em São Paulo. O celular já era um pouco mais compacto conforme figura 2.

Figura 2 – Motorola PT-550



Fonte: Techtudo – O primeiro celular da história<sup>2</sup>

<sup>1</sup> Disponível em: <[http://www.vintagemobilephones.com/Motorola\\_DynaTAC\\_Series.html](http://www.vintagemobilephones.com/Motorola_DynaTAC_Series.html)> Acesso em janeiro de 2017.

<sup>2</sup> Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2011/07/o-primeiro-celular-da-historia.html>>. Acesso em janeiro de 2017

Com o passar do tempo estes dispositivos móveis foram ficando mais complexos, assim a comunicação com esses dispositivos, antes simples, foram aperfeiçoando-se, passando por mensagens de texto (SMS) e PDAs e chegando no patamar atual com um dispositivo que mais se assemelha com um computador pessoal do que celular, propriamente dito. Para chegarmos nos sistemas operacionais móveis atuais, o PDA com o Palm OS foi um dos promissores com a sua ideia da multi-tarefa. Outro promissor SO da época foi o Symbian. Segundo Tanenbaum (2008), a Nokia, com o Symbian ofereceu ao usuário um ótimo sistema com gerenciamento de arquivos, multi-tarefa, multi-threading, armazenamento expansível e uma das coisas mais importantes para os programadores, orientação a objetos que dá mais dinamismo para a programação e abstração.

Atualmente existem três sistemas operacionais: Android da Google, iOS da Apple e Windows Phone da Microsoft.

#### 4.2.2 ANDROID

Conforme Beavis (2008) a Android Inc. foi criada em 2005 na Califórnia por Andy Rubin, Rich Miner, Nick Sears e Chris White. Criaram a empresa para embarcar o *software* em câmeras digitais, porém viram o mercado de *mobile* muito fraco com a Nokia (Symbian) Windows Mobile. Assim resolveram encarar as duas empresas e criar um bom S.O. A Google viu a empresa com bons olhos e adquiriram a Android Inc. incorporando-a ao seu grande número de produtos. Futuramente, o S.O. tornou-se o carro chefe da empresa.

Com o crescente avanço da tecnologia hoje tem-se diversas plataformas onde pode-se criar e programar aplicações. O Android é um deles. S.O. que é proprietária da Google e que hoje concorre diretamente por usuários junto à Apple. Atualmente, este S.O. móvel, é o móvel mais utilizado do mundo pela sua alta aceitação do público e ser utilizada por diversas empresas do mercado.

Baseada em UNIX já se expandiu para outras plataformas como carros e TVs.

Para criar aplicações nativas da plataforma da Google, deve-se ter conhecimento em Java e Programação Orientada a Objetos.

Em sua arquitetura, conforme figura 3, o *Kernel* é quem comanda todo o acesso ao *hardware*. Bibliotecas fazem a ponte do Android Runtime (S.O. Android) com o *Kernel* e, assim, tem-se o acesso a todos os recursos de *hardware* do dispositivo.

Figura 3 – Arquitetura Android



Fonte: Página destinada aos desenvolvedores Android<sup>3</sup>

### 4.2.3 iOS

O iOS foi apresentado em uma convenção da Apple pelo nome maior da companhia, Steve Jobs, e inicialmente não tinha criado uma linguagem de programação acessível para os programadores em geral. Porém, após grandes vendas, foi criado um SDK e linguagem de programação proprietária para a criação de aplicações dos desenvolvedores fazendo crescer ainda mais as opções de boas aplicações para usuários fazendo a concorrência com o Android crescer ainda mais.

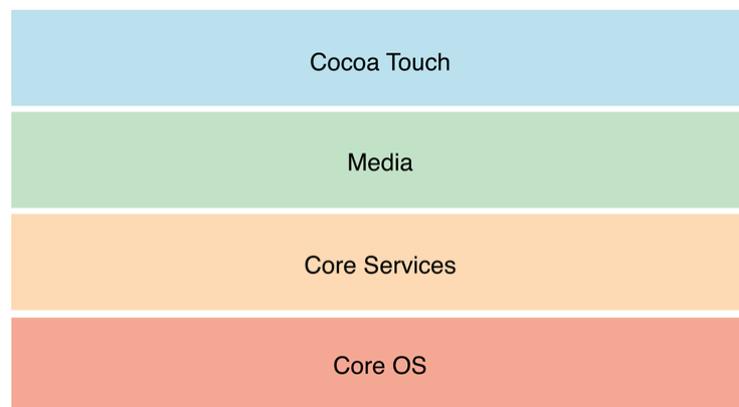
Sistema Operacional proprietário da Apple, tem plataforma totalmente fechada e que somente roda em dispositivos da empresa. Com isso, torna-se difícil o acesso às

<sup>3</sup> Disponível em: <<https://developer.android.com/guide/platform/index.html>> Acesso em janeiro de 2017.

informações mais aprofundadas do S.O., porém, é sim possível o desenvolvimento de aplicações nativas com o SDK que a própria Apple disponibilizou aos desenvolvedores.

A arquitetura da plataforma, conforme figura 4, consiste em quatro camadas: *Core OS*, a *Core Services*, a de *Mídia* e *Cocoa Touch*. Todo o sistema foi baseado no MAC OS X. *Cocoa Touch* oferece ferramentas e infraestrutura para implementar eventos e aplicações à interface do dispositivo. *Media* é responsável por fornecer recurso de áudio e vídeo, possuem biblioteca como a *OpenGL*. *Core Services* fornece serviços fundamentais como banco de dados, GPS, *AddressBook* etc. Ultimamente, *Core OS* é o kernel (núcleo) do sistema, junto a ela há drivers e interfaces básicas do sistema.

Figura 4 – Arquitetura iOS



Fonte: Página destinada aos desenvolvedores iOS<sup>4</sup>

#### 4.2.4 WINDOWS PHONE

Visando entrar e competir no mercado de *mobile* fortemente a Microsoft lançou a plataforma de S.O. proprietária. Anteriormente já havia alguma atenção com o desenvolvimento e criação de S.O. para dispositivos móveis com o Windows CE, Pocket PCs, Windows Mobile 6 etc.

Com o sucesso exponencial de Android e iOS, o Windows Phone 7 foi criado e lançado em 2010 com o intuito de entrar no mercado *mobile*, crescer e roubar boa parte da fatia do mercado da Google e Apple. A Microsoft tentou e ainda tenta, com todos os esforços este feito do mercado de *smartphones* dos outros gigantes. Com o sucesso que o S.O. de computadores pessoais que detém, imaginava que pudesse ter o mesmo caminho nos

<sup>4</sup> Disponível em:

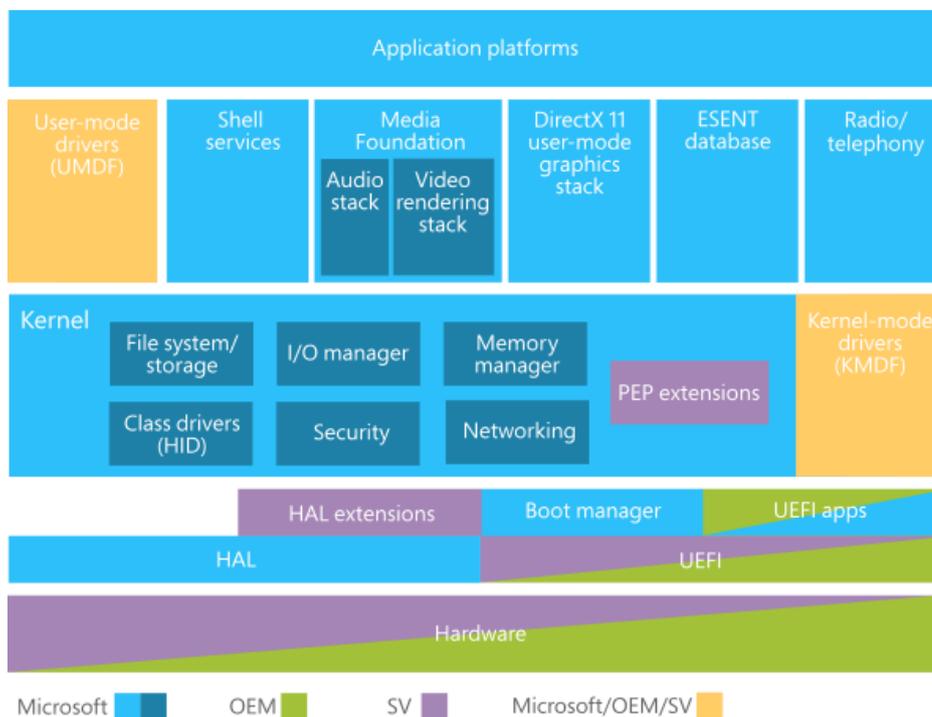
<<https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>> Acesso em janeiro de 2017.

*smartphones*. Porém, mesmo com um sistema bom, poucos desenvolvedores aderiram a plataforma deixando-a isolada em relação aos demais S.O. Existem muitas aplicações que não são compatíveis com este sistema.

Hoje, com o lançamento do Windows Mobile 10, a Microsoft tenta vender o sistema como um S.O. que execute em qualquer aplicativo em qualquer dispositivo, sendo móvel ou computador pessoal e, assim, alavancar as vendas e se permanecer vivo no segmento *mobile*.

O funcionamento deste SO baseia-se basicamente em camadas de *hardware*, *kernel* (núcleo), aplicação e interface (inserida na camada de *drivers*). Os fabricantes são responsáveis pelo fornecimento dos *drivers* e bibliotecas para interação do *kernel* (hoje, com o Windows Phone 10 é utilizado o mesmo *kernel* do Windows NT, mesmo *kernel* do Windows de PCs. Antes do Windows 7, o *kernel* utilizado era do Windows CE) com o *hardware* e da própria aplicação com o *hardware* (com certa abstração para facilitação dos programadores) conforme figura 5.

Figura 5 – Arquitetura do Windows Phone



Fonte: Windows Phone Hardware Development<sup>5</sup>

<sup>5</sup> Disponível em: <[http://sysdev.microsoft.com/en-us/Hardware/oem/docs/Getting\\_Started/Windows\\_Phone\\_architecture\\_overview](http://sysdev.microsoft.com/en-us/Hardware/oem/docs/Getting_Started/Windows_Phone_architecture_overview)> Acesso em janeiro de 2017.

### 4.3 DESENVOLVIMENTO *MOBILE*

O desenvolvimento *mobile* está a todo vapor no que se diz em mercado de trabalho e oportunidades. Atualmente, segundo Exame (2016), 57% da população brasileira possui *smartphones* e o gasto médio com telefonia por voz cai e o 3G está em 48,7% dos que possuem *smartphones*, já o 4G está em 13,5%, é visível o tamanho da importância do desenvolvimento *mobile* em todo o mundo.

Os aplicativos móveis têm a vantagem de não necessitarem de uma grande equipe de programação para o desenvolvimento. Há ainda a facilidade de comunicação e interação dos aplicativos internos do celular.

As aplicações do iOS são desenvolvidas na linguagem Objective C, com grandes semelhanças ao tão conhecido C, têm a restrição da necessidade de programar somente em ambiente Mac. Somente inserido ao ambiente proprietário da Apple é possível realizar testes de aplicativos que necessitem rodar em dispositivos iPhone. Sua IDE é a XCode, ambiente padrão da Apple para produção de aplicações que rodam em Mac OS X. As aplicações iOS levam a grande vantagem de não terem problemas com integração e *layout*, visto que todos os dispositivos são produzidos somente pela própria fabricante do sistema.

O Android dá a liberdade de implementar e codificar aplicações em qualquer sistema operacional, Mac, Windows e Linux. Assim, o nicho de programadores é muito maior desta plataforma, visto que existe maior flexibilidade do S.O. Outra vantagem do Android ao iOS é que o *kernel* de qualquer Android se baseia em Linux, possibilitando que muitas aplicações possam rodar em *background*. Com todas as vantagens, Android possui uma vasta quantidade de aplicações disponíveis. Pode-se visualizar o comparativo de S.O. no quadro 1.

A linguagem de programação é o Java conjuntamente ao XML (Extensible Markup Language). O XML é utilizado para marcar e projetar as interfaces ao usuário. Seu IDE é o Eclipse e o Android Studio, lançado em maio de 2013 facilitando a instalação e acesso a todos os recursos necessários ao programador Android. Com o Eclipse torna-se necessário a instalação do próprio IDE e posteriormente *plug-ins* para programar e emular o Android.

Quadro 1 – Quadro comparativo Android x iOS

<b>Característica</b>	<b>Android</b>	<b>iOS</b>
Liberdade de Desenvolvimento	X	
Emulador e Debugador no SDK	X	X
Facilidade de Portabilidade		X
Código Aberto	X	
Construtor de Interface	X	X
IDE	Eclipse, Netbeans etc.	XCode
Linguagem de Programação	Java com XML	Objective-C
Nível de Fragmentação	Alto	Baixo
Nível de Segurança	Baixo	Alto

Fonte: ENACOMP 2011<sup>6</sup>

## 5 LIBRAS

LIBRAS é uma sigla que representa Língua Brasileira de Sinais e que infelizmente poucos têm conhecimento. Esta língua foi criada para a comunicação entre eles e para tentarem se comunicar com qualquer outro ouvinte, como chamam pessoas não surdas. Segundo Silva, Junior, Lima (2008) “a necessidade de comunicação obriga a quebra da barreira comunicacional. Sem acesso à língua oral, os surdos desenvolveram línguas de sinais, permitindo a comunicação por um meio gesto-visual”.

A LIBRAS foi regulamentada e oficializada pela Lei 10.436 em 2002 pelo decreto 5.626 em 2006. Conforme Silva, Junior, Lima (2008), esta conquista foi possível graças à luta de pessoas surdas, pais, intérpretes, professores, professores da área, e em especial, comunidades e associações de surdos. Vale lembrar que até 2002, LIBRAS era considerada uma linguagem e não língua. Costa (2008, apud Silva, Junior, Lima 2008, p. 4) descreve a diferença entre língua e linguagem:

O termo linguagem deve ser entendido como a faculdade mental que distingue os humanos de outras espécies animais e possibilita nossos modos específicos de pensamento, conhecimento e interação com os semelhantes. É a

<sup>6</sup> Disponível em: <[http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011\\_submission\\_54.pdf](http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011_submission_54.pdf)>. Acesso em janeiro de 2017.

capacidade específica à espécie humana de se comunicar por meio de um sistema de signos (ou língua). A língua é, então, entendida como forma de realização da linguagem; como sistema linguístico necessário ao exercício da linguagem na interlocução ou como instrumento do qual a linguagem se utiliza na comunicação.

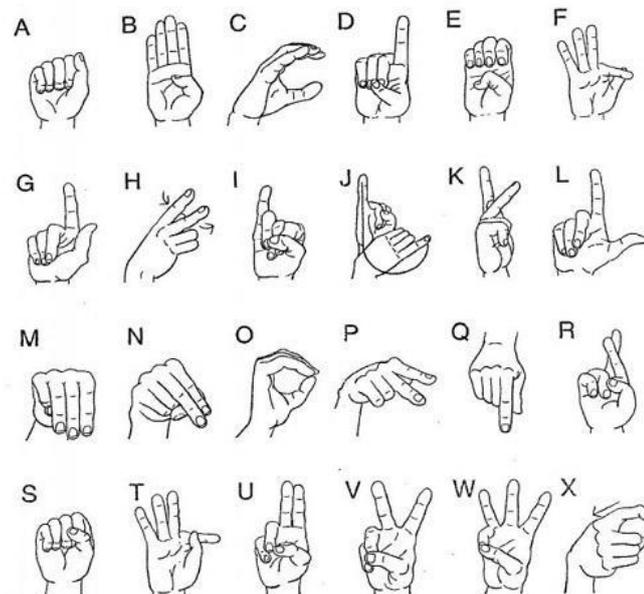
Levando em consideração o que Costa (2008) diz e com o reconhecimento da LIBRAS como uma língua pelo governo brasileiro, a LIBRAS torna-se o instrumento de comunicação legal e oficial das pessoas surdas.

Segundo último Censo Demográfico do IBGE (2010), existem 45,6 milhões de pessoas com algum grau de deficiência e que representam 5,8% da população e destas, 9,7 milhões de pessoas têm algum grau de surdez e 347.481 não ouvem de modo algum.

A quebra de paradigmas ainda é constante no mundo dos deficientes e em especial, neste trabalho, dos surdos. Está claro que estes deficientes a cada dia precisam romper barreiras. Os surdos sofreram e ainda sofrem preconceitos e banalização de sua língua.

Saber LIBRAS não basta saber o alfabeto. O alfabeto em LIBRAS, também conhecido como alfabeto datilológico é mais comumente utilizado para designar nomes próprios ou escritas de palavras, mas não para comunicação. Aprender o alfabeto é somente o início do aprendizado em Língua de Sinais. É necessário após o aprendizado do alfabeto, vide figura 6, aprender os sinais de palavras, cidades etc., além de estruturas gramaticais (AcessaSP, 2015).

Figura 6 – Alfabeto em LIBRAS



Fonte: ACESSA SP. Governo de São Paulo<sup>7</sup>

---

<sup>7</sup> Disponível em: <<http://rede.acessasp.sp.gov.br/blogada/basta-saber-o-alfabeto>>. Acesso em janeiro de 2017.

## 6 MATERIAIS E MÉTODOS

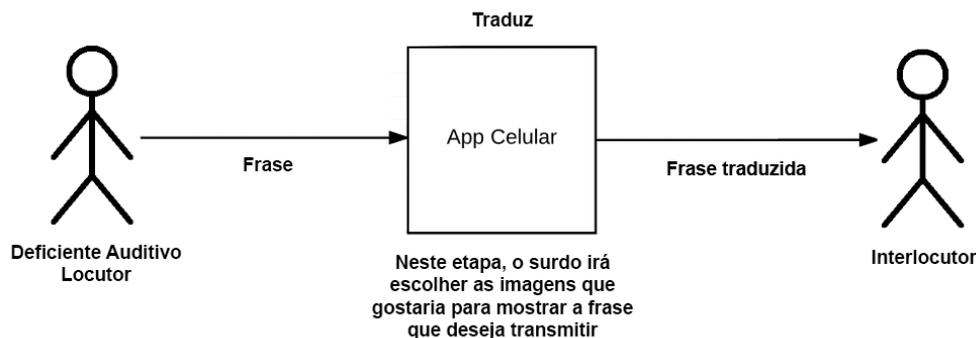
Para todo o projeto de pesquisa utilizou-se um notebook com processador Intel Core i7-3537U CPU @ 2.00GHz 2.5GHz, 8 GB de RAM com placa de vídeo Intel Graphics integrada ao processador e Windows 10 64 bits. Além do notebook, foram utilizados dois celulares: Moto X com Android Lollipop 5.1 e Moto G com Android Marshmallow 6.0. Para programação e testes da aplicação utilizou-se IDE Android Studio. Em pesquisas utilizou-se navegador Mozilla Firefox.

Para o desenvolvimento do trabalho foram necessários estudos sobre a LIBRAS no Brasil, as dificuldades encontradas, conversas com surdos e conversas com familiares surdos.

Posteriormente, efetuou-se o estudo sobre desenvolvimento *mobile*. Dentro deste segmento, estudou-se desenvolvimento híbrido e desenvolvimento dedicado em cada plataforma.

Após a etapa de estudos, estudou-se o propósito da aplicação e desenhou-se um esboço de qual seria a funcionalidade da aplicação conforme figura 7.

Figura 7 – Funcionamento do aplicativo



Fonte: Elaborada pelo autor

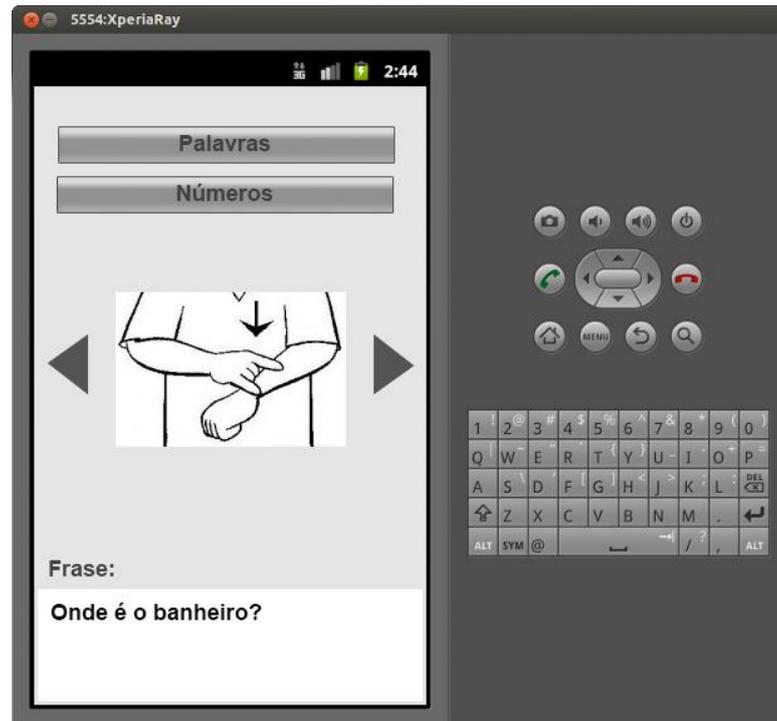
Efetuoou-se testes com aplicação híbrida para verificar viabilidade.

Sem a viabilidade no desenvolvimento de aplicação híbrida, iniciou-se o estudo em desenvolvimento Android para aplicação nativa no S.O.

Verificou-se viabilidade neste desenvolvimento de aplicação nativa e iniciou-se desenvolvimento de aplicação com Android Studio.

Criou-se um esboço de qual seria a interface da aplicação para o usuário conforme figura 8.

Figura 8 – Protótipo do aplicativo



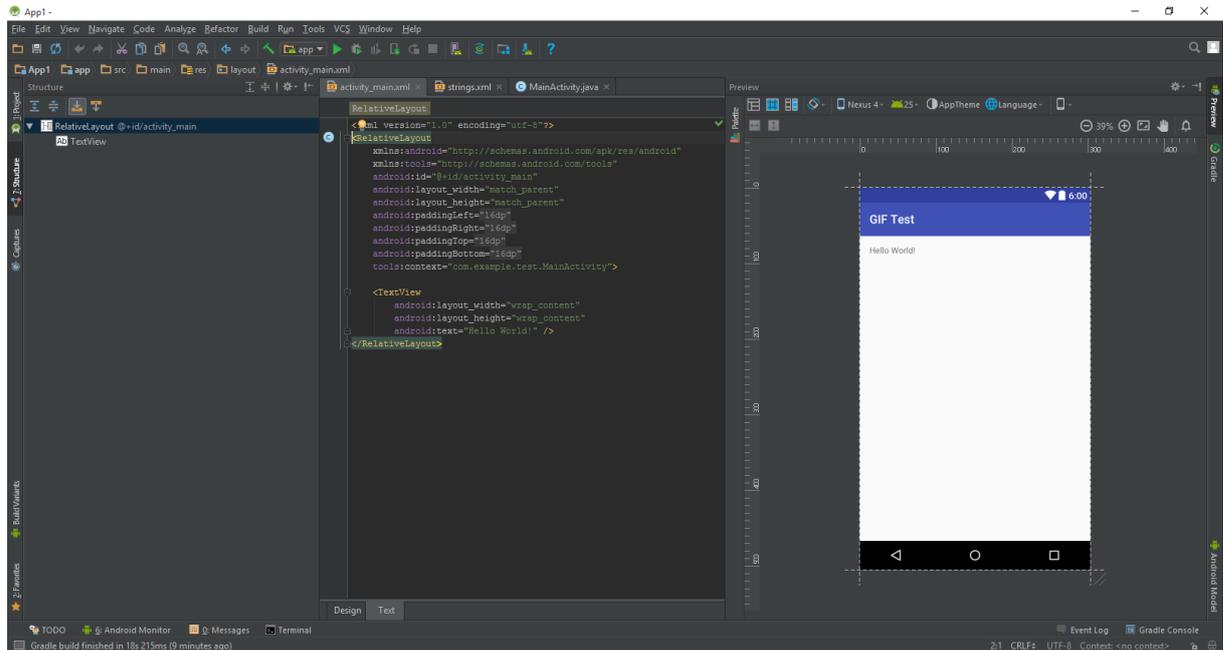
Fonte: Elaborada pelo autor

A ferramenta de interface utilizada, Android Studio, vide figura 9, é o um Ambiente de Desenvolvimento Integrado (IDE) desenvolvida pela Google voltada para desenvolvimento, *debug* e testes Android. É baseada no eclipse com diversas modificações para facilitar o trabalho do desenvolvedor. A versão atual é a 2.2.3. Para desenvolver para Android deve-se utilizar Java e XML, porém, basta um bom conhecimento em Orientação à Objetos.

Possui a ferramenta de *AutoComplete* que facilita para as implementações mais ágeis pelos programadores. A função desta ferramenta é de completar automaticamente determinada chamada de funções, métodos, classes etc.

Utilizou-se este Ambiente de desenvolvimento por ser o mais indicado hoje para desenvolvimento de aplicações Android e prático *debug* e testes em emuladores que podem ser instalados com a IDE.

Figura 9 – IDE Android Studio. Ambiente de desenvolvimento



Fonte: Elaborada pelo autor

Estudou-se sobre diversos tipos de armazenamento de GIFs para a aplicações no Android a fim de evitar problemas.

Pesquisas e estudos sobre implementações GIF em aplicações Android efetuadas a fim de entender e implementar o algoritmo mais profissional.

Após etapas de implementações, conforme códigos no anexo A, testes e estudos, efetuou-se testes com intérpretes para ter-se algum *feedback* e trabalhar em bugs e sugestões.

Com isto, efetuou-se implementação de ideias compartilhadas com intérpretes.

Em busca de melhoria contínua, efetuou-se testes com o público alvo, surdos, e assim obter o real *feedback* da aplicação.

Com *feedbacks* colhidos, implementou-se os apontamentos e ideias compartilhadas com surdos.

Finalizou-se com últimos testes com surdos com última versão do aplicativo do projeto.

## 7 PROBLEMÁTICAS E DIFICULDADES

Muitos deficientes auditivos não tem a oportunidade de estudos como qualquer pessoa teria. A inclusão social destas pessoas, hoje, já está num estágio bastante avançado, porém ainda não o suficiente para que elas tenham a devida inserção.

O surdo precisa, sempre, para uma simples consulta médica levar intérprete. O intérprete é, segundo Silva, Junior, Lima (2008), responsável pela comunicação de falantes de duas línguas distintas. Os surdos precisam, não somente para a finalidade citado anteriormente, mas também para, por exemplo, diversas outras necessidades como uma conversa com um gerente para tirar dúvidas de sua conta ou para um simples questionamento na rua para saber onde podem encontrar o banco mais próximo e, até mesmo, informações do posto de gasolina mais próximo é necessário um intérprete.

Desde criança o surdo encontra dificuldades para aprendizado e comunicação visto que a língua que está aprendendo não é a que ela entende. A falta de compreensão da língua falada a impede de desenvolver uma linguagem.

Um dilema ainda frequente no sistema educacional brasileiro decidir se deve ou não inserir o surdo dentro da escola regular, desde a educação infantil até o ensino médio (Pozzer, 2015). Não existe a devida atenção nos cursos de pedagogia para surdos, uma matéria completa de LIBRAS para possibilitar inclusão destes alunos. Isso leva à questionamentos frequentes: Existe preparo de professores para tal? Existe algum apoio ou material pedagógico para os professores ou eles teriam que procurar por conta própria? Com todos estes dilemas, quem sempre sai perdendo é o surdo e sua família.

Existem professores que buscam a auto formação inclusiva e se especializam em LIBRAS e braile, porém soma-se ao fato da inclusão social que o pedagogo almeja com o dever deste educador de sempre buscar a harmonização do ensino inclusivo com o restante dos alunos da classe. Em conjunto desta problemática surge outra: Os cursos de pedagogia estão requerendo e provocando esta sensação de mudança em parte da educação pedagógica das escolas brasileiras?

## 8 VIABILIDADE

Com a enorme difusão dos *smartphones* e tendo em vista que grande parte dos deficientes auditivos, ou surdos, como eles gostam de ser chamados, possuem em mãos, este projeto torna-se viável. A internet está com tecnologia o suficiente para dar suporte a qualquer tipo de surdo, caso necessário.

Outra viabilidade é a da aplicação ser projetada para ser totalmente gratuita e que qualquer biblioteca utilizada é gratuita.

Todos os testes efetuados ratificam a viabilidade do projeto já que todos os surdos que utilizaram o aplicativo retornaram um *feedback* positivo e viram utilidade da aplicação para os fins desejados.

Foram feitos, também, testes para geração de aplicações híbridas, com o Xamarin<sup>8</sup> e com JQuery Mobile<sup>9</sup> e PhoneGap<sup>10</sup>. Todos estes testes foram malsucedidos, já que com qualquer plataforma existiam empecilhos que faziam.

---

<sup>8</sup> Disponível em: <<https://www.xamarin.com/platform>>. Acesso em janeiro de 2017.

<sup>9</sup> Disponível em: <<http://api.jquerymobile.com/>>. Acesso em dezembro de 2016.

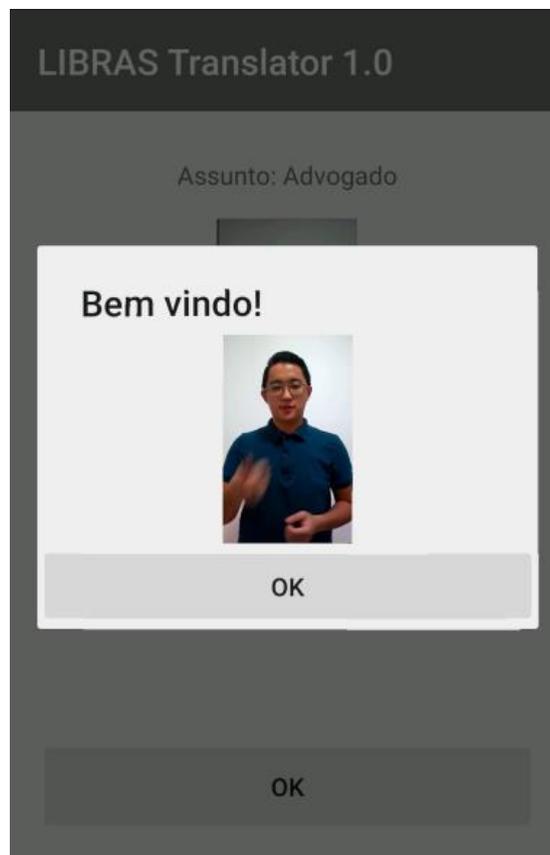
<sup>10</sup> Disponível em: <<http://docs.phonegap.com/>>. Acesso em dezembro de 2016.

## 9 APLICAÇÃO

A aplicação é destinada às pessoas com surdez que não tiveram a devida educação para compreender o português corretamente e tem como sua língua primária a língua de sinais.

Primeiramente o usuário, ao abrir a aplicação, visualizará um pop-up desejando boas vindas e explanando o que deve ser feito, conforme figura 10, poderá prosseguir na aplicação buscando palavras ou buscando assuntos.

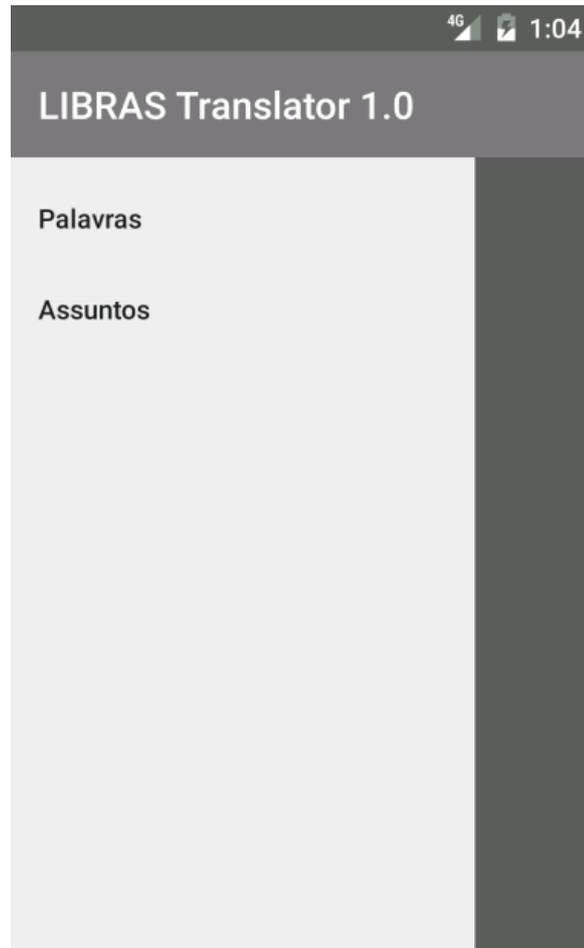
Figura 10 – LIBRAS Translator, telas de boas vindas



Fonte: Elaborado pelo autor

Após a tela de boas vindas, o usuário poderá prosseguir escolhendo se deseja utilizar o aplicativo por palavras ou por assunto.

Figura 11 – Tela de escolha de contexto

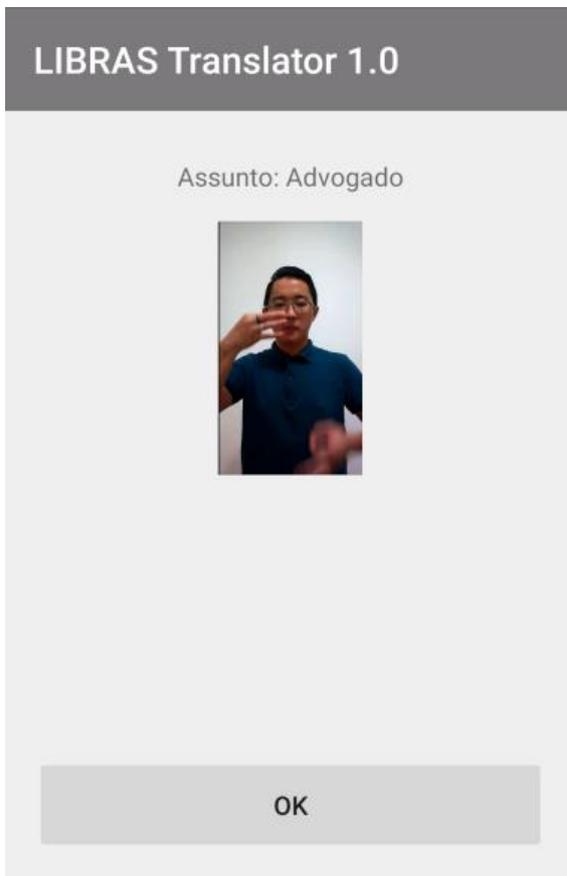


Fonte: Elaborado pelo autor

## 9.1 CONTEXTO: ASSUNTO

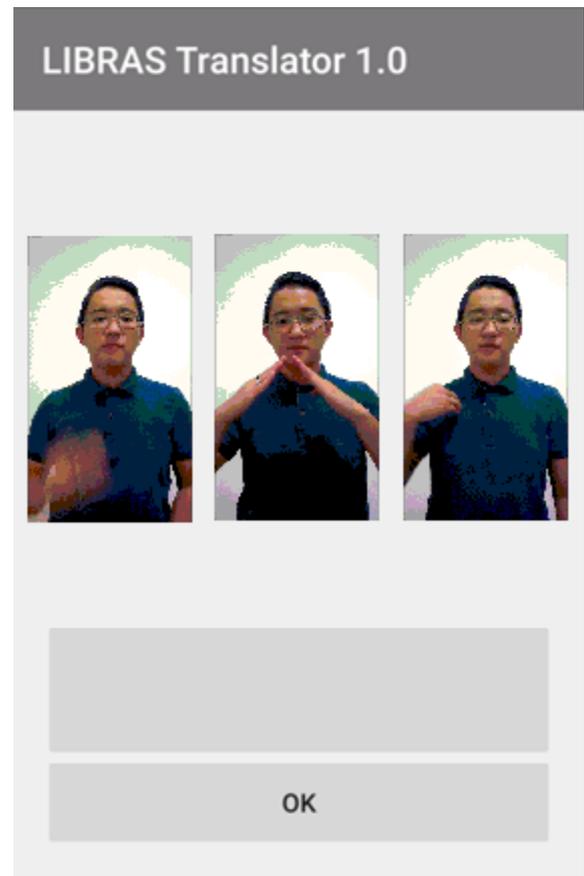
O usuário fazendo a escolha do contexto assunto, conforme figura 12, precisará escolher o assunto que deseja e, assim, terá frases com uma combinação de GIFs, conforme figura 13, que foram elaboradas juntamente ao *feedback* dos intérpretes e surdos.

Figura 12 – Tela do contexto assunto



Fonte: Elaborado pelo autor

Figura 13 – Tela de frases

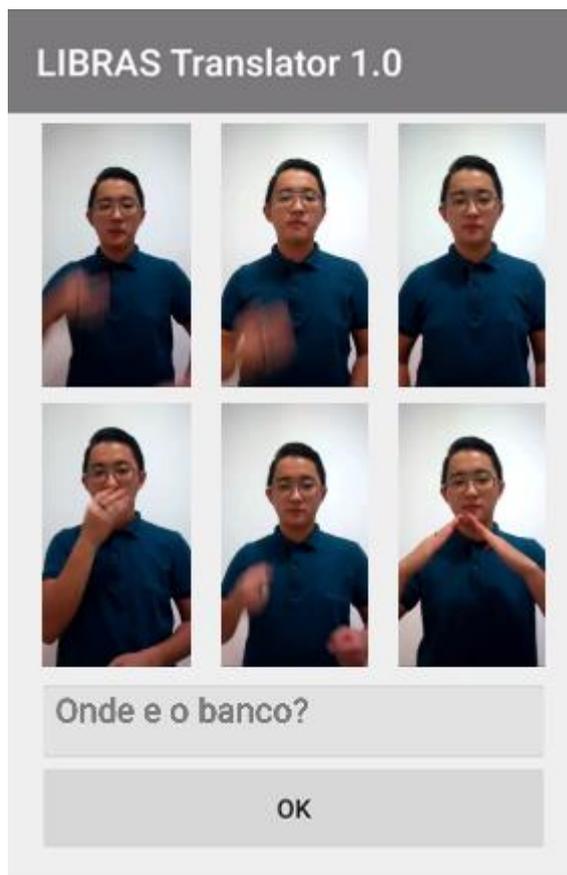


Fonte: Elaborado pelo autor

## 9.2 CONTEXTO: PALAVRAS

O usuário fazendo a escolha do menu de contexto de palavras o usuário encontrará todas as palavras através de GIFs que foram gravados individualmente para ser adicionado na aplicação. Com isso, o usuário selecionará todas as palavras desejadas para completar uma frase e a aplicação se encarregará de montar a frase.

Figura 14 – Tela de palavras para montagem de frases



Fonte: Elaborada pelo autor

## 9 TESTES

O aplicativo foi projetado para que o surdo conseguisse ter a frase necessitada o mais rápido possível para que não perdesse o ouvinte que estivesse ajudando-o foi implementado diretamente em Android. A escolha do Sistema Operacional foi com o embasamento de que é o SO mais difundido de todos com 96% dos *smartphones* vendidos sendo que os restantes se dividem em Windows Phone e iOS (Tecnoblog 2016).

O projeto precisaria de múltiplos testes para que o funcionamento agradasse todos, portanto foram feitos com intérpretes e surdos.

Um dos testes feitos foi a tentativa de implementação de GIF em aplicação híbrida, porém sem sucesso. Foi estudado implementação de GIF em Xamarin (Utilizado para desenvolvimento em linguagem C#, desenvolvido pela Microsoft) e também em JQuery Mobile (linguagem JQuery adaptada para aplicações *mobile* adequando, por exemplo, com responsividade todos os aplicativos para *smartphones*) em conjunto com PhoneGap (Plataforma produzida pela Adobe para fazer o *build* do projeto em programação *web* e gerar o arquivo para Android ou iOS). Em ambos os casos, os testes não foram efetivos. Na primeira implementação, haviam mais bibliotecas para serem estudadas para utilização que tornaram inviáveis. Já no segundo caso, GIFs não executam em aplicações geradas pelo PhoneGap.

## 10 CONCLUSÕES

Durante o desenvolvimento da aplicação foram estudados alguns aspectos no que se diz de melhoria de tecnologia de desempenho e funcionalidade que, assim que visto alguma queda de performance ou bugs, foram implementados como as diversas formas de implementação de GIFs nas aplicações Android, bibliotecas disponíveis. Também houve o cuidado de como seria feito o armazenamento estes arquivos GIFs.

As pesquisas feitas com intérpretes e surdos foram de extrema importância ao projeto possibilitando futuramente melhorar a aplicação. Diversas outras implementações podem ser feitas pela vasta experiência que o pesquisador possa ter ou fazendo outros trabalhos de campo entrevistando surdos e efetuando testes.

Assim, com a implementação do trabalho junto aos deficientes e diversos testes, foi percebido facilidade de uso dos surdos e intérpretes e elogios por parte dos deficientes que fizemos os testes. A aplicação gerou boas facilidades aos surdos para que, assim, continuem rompendo parte da barreira da comunicação.

Como trabalho futuro outros trabalhos e pesquisas podem ser desenvolvidos visto que a quantidade de palavras é enorme e poderá sempre ser incrementado à aplicação.

Outros incrementos e pesquisas podem ser feitos em conjunto com intérpretes e surdos com mais questionários a eles para melhoria. Uma boa implementação seria de mais frases pré-configuradas conforme necessidade de surdos.

Estudar outras possíveis implementações de armazenamento e GIFs, efeitos, nunca tirando de vista o propósito, ajudar sempre o deficiente auditivo.

## REFERÊNCIAS BIBLIOGRÁFICAS

Acessasp Governo do Estado de São Paulo. **Basta saber o alfabeto?**. 2015. Disponível em: <<http://rede.acessasp.sp.gov.br/blogada/basta-saber-o-alfabeto>>. Acesso em 10 de janeiro de 2017.

Almeida, M. P.; Almeida, M. E. **Círculo Fluminense de Estudos Filológicos e Linguísticos**. Revista Philologus, Ano 18, nº 54. Portal Filologia. Disponível em: <<http://www.filologia.org.br/revista/54supl/031.pdf>>. Acesso em 25 de março de 2016.

Almeida, André Gustavo Duarte de. Instituto Federal de Educação, Ciência e Tecnologia, Rio Grande do Norte. **Introdução ao Android**. Disponível em: <<https://docente.ifrn.edu.br/andrealmeida/disciplinas/2012.2/minicurso-introducao-ao-android/parte-01-conhecendo-o-sistema-e-primeiro-programa>>. Acesso em 11 de julho de 2016.

Apple Inc. **Swift**. Disponível em: <<http://www.apple.com/br/swift>>. Acesso em 20 de agosto de 2016.

Avancini, V.; Mognon, P. H.; Kazumi, D.; Stefani, G. **Windows Phone**. Departamento de Informática e Estatística UFSC, Trindade, 2012.

Beavis, G. **A complete history of Android**. 2008. Disponível em <<http://www.techradar.com/news/phone-and-communications/mobile-phones/a-complete-history-of-android-470327>>. Acesso em 03 de janeiro de 2017.

Cardozo, E.; Magalhães, M. F.; Faina, L. F. **Introdução aos Sistemas Operacionais**. Faculdade de Engenharia Elétrica e de Computação Unicamp, Campinas, 2002, p. 2.

Exame. **57% da população brasileira usa smartphone, diz estudo**. 2016. Disponível em <<http://exame.abril.com.br/tecnologia/57-da-populacao-brasileira-usa-smartphone-diz-estudo>>. Acesso em 02 de janeiro de 2017.

Google Inc. **Android**. Disponível em: <[https://www.android.com/intl/pt-BR\\_br](https://www.android.com/intl/pt-BR_br)>. Acesso em 03 de agosto de 2016.

Google Inc. **Arquitetura da plataforma**. Disponível em: <<https://developer.android.com/guide/platform/index.html>>. Acesso em 10 de janeiro de 2017.

Instituto Nacional de Educação de Surdos INES. **Dicionário da Língua Brasileira de Sinais**. 2010. Disponível em: <[http://www.ines.gov.br/dicionario-de-libras/main\\_site/libras.htm](http://www.ines.gov.br/dicionario-de-libras/main_site/libras.htm)>. Acesso em 3 de outubro de 2016.

IBGE. Censo Demográfico 2010. **Características Gerais da População, Religião e Pessoas com Deficiência**. Disponível em: <[http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd\\_2010\\_religio\\_deficiencia.pdf](http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religio_deficiencia.pdf)>. Acesso em 20 de janeiro de 2017.

Inbraud Instituto Brasileiro de Reabilitação Auditiva. **O Perfil da deficiência auditiva no Brasil**. Disponível em: <<http://www.inbraud.org.br/o-perfil-da-deficiencia-auditiva-no-brasil/>>. Acesso em 20 de janeiro de 2010.

Microsoft. **Windows Phone architecture overview**. 2015. Disponível em: <[http://sysdev.microsoft.com/en-us/Hardware/oem/docs/Getting\\_Started/Windows\\_Phone\\_architecture\\_overview](http://sysdev.microsoft.com/en-us/Hardware/oem/docs/Getting_Started/Windows_Phone_architecture_overview)>. Acesso em 10 de janeiro de 2017.

Mendonça, V. R. L.; Bittar, T. J.; Dias, M. S. **Um estudo dos Sistemas Operacionais Android e iOS para o desenvolvimento de aplicativos**. 2011. Disponível em: <[http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011\\_submission\\_54.pdf](http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011_submission_54.pdf)>. Acesso em 12 de dezembro de 2017.

Portal Educação. **Dificuldades de Aprendizagem em Surdos**. 2013. Disponível em: <<https://www.portaleducacao.com.br/pedagogia/artigos/26871/dificuldades-de-aprendizagem-em-surdos>>. Acesso em 10 de janeiro de 2017.

Pozzer, A. **A inclusão de alunos surdos em escola regular e os desafios para a formação de professores**. Departamento de Ciências Humanas URI, Frederico Westphalen, 2015.

Portal FENEIS, **Federação Nacional de Educação e Integração dos Surdos de São Paulo**. Disponível em: <<http://www.feneissp.org.br/index.php/sobrenos>>. Acesso em 28 de março de 2016.

Silva, A. T. C.; Júnior, M. R. M.; Lima, F. J. **O Intérprete de Língua Brasileira de Sinais no ensino fundamental e seu papel na escola comum**. Centro de Educação DAEPE UFPE, Recife, 2008.

Silveira, C. H. Universidade Federal de Santa Maria - Centro de Educação. **Curso a Distância de Especialização em Educação Especial. Disciplina LIBRAS I**. Disponível em: <<http://coral.ufsm.br/edu.especial.pos/images/libras.pdf>>. Acesso em 01 de abril de 2016.

Tanenbaum, A. S. **Sistemas Operacionais Modernos**. Pearson, 2008, p. 1.

Techtudo. **O primeiro celular da história**. 2011. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2011/07/o-primeiro-celular-da-historia.html>>. Acesso em 20 de dezembro de 2016.

Tecnoblog. **95,5% dos smartphones vendidos no Brasil são Androids**. Disponível em: <<https://tecnoblog.net/203749/android-ios-market-share-brasil-3t-2016>>. Acesso em 10 janeiro 2017.

Vintage Mobile Phones. **Motorola DynaTAC Series**. 2008. Disponível em: <[http://www.vintagemobilephones.com/Motorola\\_DynaTAC\\_Series.html](http://www.vintagemobilephones.com/Motorola_DynaTAC_Series.html)>. Acesso em 15 de janeiro de 2017.

## ANEXO A – IMPLEMENTAÇÃO DE GIF EM ANDROID

Há várias formas para implementação de GIF no Android, uma delas, e a utilizada, é demonstrada pelo código 1. O arquivo MainActivity.java e o activity\_main.xml também são demonstrados pelos códigos 2 e 3 respectivamente.

Código 1 – GifView.java. Implementação de classe para rodar GIFs

```
/**
 * Created by kindi.wei on 09/11/2016.
 */

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Movie;
import android.os.SystemClock;
import android.util.AttributeSet;
import android.view.View;

import java.io.InputStream;

public class GifView extends View{

    private InputStream gitInputStream;
    private Movie gifMovie;
    private int movieWidth, movieHeigth;
    private long movieDuration;
    private long movieStart;

    public GifView(Context context){
        super(context);
        init(context);
    }

    public GifView(Context context, AttributeSet attrs){
        super(context, attrs);
        init(context);
    }

    public GifView(Context context, AttributeSet attrs, int defStyleAttr){
        super(context, attrs, defStyleAttr);
        init(context);
    }

    private void init(Context context){
        setFocusable(true);
        gitInputStream =
context.getResources().openRawResource(R.drawable.banco);

        gifMovie = Movie.decodeStream(gitInputStream);
        movieWidth = gifMovie.width();
        movieHeigth = gifMovie.height();
        movieDuration = gifMovie.duration();
    }

    @Override
```

```

protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    setMeasuredDimension(movieWidth, movieHeight);
}

public int getMovieWidth(){
    return movieWidth;
}

public int getMovieHeight(){
    return movieHeight;
}

public long getMovieDuration(){
    return movieDuration;
}

protected void onDraw(Canvas canvas){

    long now = SystemClock.uptimeMillis();

    if(movieStart == 0){
        movieStart = now;
    }

    if(gifMovie != null){
        int dur = gifMovie.duration();
        if(dur == 0){
            dur = 1000;
        }

        int relTime = (int)((now - movieStart) % dur);

        gifMovie.setTime(relTime);

        gifMovie.draw(canvas, 0, 0);

        invalidate();
    }
}
}

```

Fonte: Elaborado pelo Autor

Código 2 – MainActivity.java. Arquivo principal de Atividades do Android.

```

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    TextView textViewInfo;
    GifView gifView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        gifView = (GifView) findViewById(R.id.gif_view);
    }
}

```

```

        textViewInfo = (TextView) findViewById(R.id.textinfo);

        String stringInfo = "";
        stringInfo += "Duration" + gifView.getMovieDuration() + "\n";
        textViewInfo.setText(stringInfo);
    }
}

```

Fonte: Elaborado pelo Autor

Código 3 – activity\_main.xml. Arquivo de layout do Android.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <com.example.test.GifView
        android:id="@+id/gif_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/textinfo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="info..." />
</LinearLayout>

```

Fonte: Elaborado pelo Autor