

**UNIVERSIDADE DO SAGRADO CORAÇÃO**

**IGOR FASSTRONI CORRÊA**

**DESENVOLVIMENTO DE APLICATIVO MOBILE  
PARA AUXÍLIO NO TRATAMENTO  
FONOAUDIOLÓGICO DE PACIENTES COM  
FISSURAS LABIOPALATINAS**

BAURU  
2016

**IGOR FASSTRONI CORRÉA**

**DESENVOLVIMENTO DE APLICATIVO MOBILE  
PARA AUXÍLIO NO TRATAMENTO  
FONOAUDIOLÓGICO DE PACIENTES COM  
FISSURAS LABIOPALATINAS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

BAURU  
2016

Corrêa, Igor Fastroni

C8241d

Desenvolvimento de aplicativo mobile para auxílio no tratamento fonoaudiológico de pacientes com fissuras labiopalatinas / Igor Fastroni Corrêa. -- 2016.

96f. : il.

Orientador: Prof. Dr. Elvio Gilberto da Silva.

Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Aplicativo mobile. 2. Físsura Labiopalatina. 3. Fonoaudiologia. 4. Android. I. Silva, Elvio Gilberto da. II. Título.

**IGOR FASSTRONI CORRÊA**

**DESENVOLVIMENTO DE APLICATIVO MOBILE PARA AUXÍLIO  
NO TRATAMENTO FONOAUDIOLÓGICO DE PACIENTES COM  
FISSURAS LABIOPALATINAS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

Banca examinadora:

---

Prof. Dr. Elvio Gilberto da Silva  
Universidade do Sagrado Coração

---

Prof. Me. Patrick Pedreira Silva  
Universidade do Sagrado Coração

---

Prof.<sup>a</sup> Esp. Maria Cecília Gomes  
Universidade do Sagrado Coração

Bauru, 2 de dezembro de 2016.

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus por ter me abençoado todo este período, para a elaboração deste trabalho.

Ao meu pai Rinaldo Aparecido Corrêa e minha mãe Maria Aparecida Fastroni Corrêa que sempre me incentivaram em toda minha vida acadêmica, a minha irmã Helen Fastroni Corrêa, que sirva de exemplo para seu futuro acadêmico, ao meu orientador Prof. Dr. Elvio Gilberto da Silva que tanto me ajudou e incentivou neste trabalho, ao Prof. Me. Patrick Pedreira que me ajudou muito durante todo o curso e também durante o desenvolvimento desse trabalho.

Dedico em especial este trabalho à minha avó Candida Lopes Fastroni que infelizmente não está mais entre nós, mas se faz necessário o agradecimento pela contribuição na minha formação pessoal e profissional, minha fonte de inspiração de luta, bondade, integridade, o amor dedicado que me faz fortalecido até hoje.

É a vocês que dedico essa vitória.

“Ainda que eu ande pelo vale da sombra da morte, não temerei mal algum, pois tu estás comigo; a tua vara e o teu cajado me protegem.” (SALMOS, 23:4).

## RESUMO

Este projeto consistiu na criação de um aplicativo mobile Android em Java para auxiliar no processo fonoterapêutico, possibilitando o profissional de fonoaudiologia inserir atividades no qual seus pacientes devem realizar durante o processo fonoterapêutico. O aplicativo permite que o fonoaudiólogo consiga visualizar a evolução de seu paciente, além de acompanhar os exercícios que foram e os que não foram realizados. Este aplicativo utiliza o banco de dados SQLite, e apresenta uma interface simples e que possibilita aos usuários que alcancem seu objetivo de forma simples. Do lado do paciente, mesmo que distante fisicamente do fonoaudiólogo é possível enviar sua evolução via e-mail, possibilitando uma conexão maior entre paciente e fonoaudiólogo. O aplicativo foi desenvolvido em três etapas, onde a primeira delas foi abstrair o problema que os profissionais de fonoaudiologia obtinha no tratamento a longa distância, a segunda etapa foi o desenvolvimento do app utilizando Java SDK nativo do Android. A plataforma Android foi escolhida para o desenvolvimento desse aplicativo por possuir um número maior de usuários ativos. A terceira etapa consistiu no aperfeiçoamento da interface, obtidos através de feedbacks de usuários finais. Atualmente o aplicativo está disponibilizado de forma gratuita no Google Play e se encontra em fase de testes com pacientes de uma instituição de ensino superior.

**Palavras-chave:** Aplicativo mobile. Físsura Labiopalatina. Fonoaudiologia. Android.

## **ABSTRACT**

This project consisted in the creation of an Android mobile application in Java to assist in the phonotherapeutic process, enabling the speech and language pathologist to insert activities in which their patients should perform during the speech therapy process. The application allows the speech therapist to be able to visualize the evolution of his patient, besides accompanying the exercises that were and those that were not performed. This application uses the SQLite database, and has a simple interface that allows users to reach their goal in a simple way. On the patient's side, even if physically distant from the speech-language pathologist, it is possible to send their evolution via e-mail, allowing a greater connection between patient and speech therapist. The application was developed in three stages, where the first one was to abstract the problem that speech-language professionals got in long-distance treatment, the second step was the development of the app using Android Java SDK native. The Android platform was chosen for the development of this application by having a greater number of active users. The third step consisted in improving the interface, obtained through feedback from end users. The app is currently available for free on Google Play and is currently being tested with patients from a higher education institution.

**Keywords:** Mobile application. Labiopalatine Fissure. Speech therapy. Android.



## LISTA DE FIGURAS

Figura 1 – Dispositivos móveis.....	17
Figura 2 - Usuários ativos de dispositivos mobile por plataforma.....	21
Figura 3 - Arquitetura da plataforma Android. ....	22
Figura 4- Arquitetura da plataforma Android. ....	25
Figura 5 - Framework de aplicações. ....	26
Figura 6- Sistema Gerenciador de Banco de Dados. ....	31
Figura 7- Ciclo de aprimoramento de um software partindo de uma boa IHC..	34
Figura 8- Camadas da Engenharia de Software. ....	39
Figura 9- Engenharia de Software na Ciência da Computação.....	40
Figura 10 – Diagramas da UML. ....	42
Figura 11– Exemplo de Diagrama de Casos de Uso.....	44
Figura 12 – Exemplo de Diagrama de Classes. ....	45
Figura 13 – Exemplo de Diagrama de Atividades. ....	46
Figura 14– Defeito x erro x falha. ....	48
Figura 15 - Modelo V – Atividades de desenvolvimento e teste de software....	52
Figura 16 – Diagrama de Classes do aplicativo. ....	60
Figura 17 – Diagrama de Atividades. ....	61
Figura 18 – Diagrama de Casos de Uso. ....	62
Figura 19 – Tela principal do aplicativo. ....	64
Figura 20 – Cadastrar Senha - Área da Fono. ....	65
Figura 21 – Acessar Área da Fono.....	66
Figura 22 – Área restrita da Fonoaudióloga. ....	67
Figura 23 – Cadastrar Exercício.....	68
Figura 24 – Cadastrar Fala Nova. ....	69
Figura 25 – Evolução do tratamento. ....	70
Figura 26 – Enviando e-mail.....	71
Figura 27 – Mostrar todos os exercícios propostos. ....	72
Figura 28 – Mostrar todos os exercícios feitos. ....	72
Figura 29 – Push Notification do Exercício.....	73
Figura 30 – Detalhando o Exercício. ....	74
Figura 31 – Detalhando o Exercício. ....	75
Figura 32 – Instalações por versão do Android. ....	76
Figura 33 – Instalações atuais por dispositivo x modelo. ....	77

## LISTA DE ABREVIATURAS E SIGLAS

<b>API</b>	Application Programming Interface
<b>DVF</b>	Disfunção Velofaríngea
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IHC</b>	Interface Homem/Computador
<b>OHA</b>	Open Handset Alliance.
<b>SQL</b>	Structured Query Language
<b>UML</b>	Unified Modeling Language

## SUMÁRIO

1 INTRODUÇÃO .....	8
2 OBJETIVOS .....	10
2.1 OBJETIVO GERAL .....	10
2.2 OBJETIVOS ESPECÍFICOS .....	10
3 REVISÃO DA LITERATURA .....	11
3.1 FALA .....	11
3.2 FISSURA LABIOPALATINA .....	11
3.3 DISFUNÇÃO VELOFARINGEA .....	12
3.4 TERAPIA EM FISSURA LABIOPALATINA .....	12
3.5 A INFORMÁTICA NO CONTEXTO DA CLÍNICA FONOAUDIOLÓGICA .....	14
<b>3.5.1 A informática na terapia de voz e da fala</b> .....	16
3.6 DISPOSITIVOS MÓVEIS .....	16
<b>3.6.1 Vantagens dos dispositivos móveis</b> .....	18
3.7 ARQUITETURA DE SOFTWARE PARA DISPOSITIVOS MÓVEIS....	18
<b>3.7.1 Mobilidade</b> .....	19
<b>3.7.2 Desenvolvimento de aplicações móveis</b> .....	19
3.8 PLATAFORMA ANDROID .....	20
<b>3.8.1 Histórico</b> .....	20
<b>3.8.2 Definição</b> .....	21
<b>3.8.3 Arquitetura Android</b> .....	22
3.8.3.1 <i>Kernel</i> .....	22
3.8.3.2 <i>Bibliotecas</i> .....	23
3.8.3.3 <i>Runtime</i> .....	24
3.8.3.4 <i>Framework de aplicação</i> .....	25
3.9 JAVA .....	26
3.10 BANCO DE DADOS .....	27
<b>3.10.1 Por que usar um banco de dados?</b> .....	28
<b>3.10.2 Objetivos de um banco de dados</b> .....	29
3.10.2.1 <i>Acesso rápido aos dados</i> .....	29
3.10.2.2 <i>Redução de redundância e inconsistência de dados</i> .....	29

3.10.2.3	Compartilhamento de dados .....	30
3.10.2.4	Aplicação de restrições de segurança.....	30
3.10.2.5	Sistema Gerenciador de Banco de Dados .....	30
<b>3.10.3</b>	<b>SQLite .....</b>	<b>32</b>
3.11	INTERFACE HOMEM-COMPUTADOR .....	32
3.12	SOFTWARE .....	35
<b>3.12.1</b>	<b>Características de um software .....</b>	<b>36</b>
<b>3.12.2</b>	<b>Tipos de software .....</b>	<b>37</b>
3.13	ENGENHARIA DE SOFTWARE .....	39
3.14	A LINGUAGEM DE MODELAGEM UNIFICADA (UML) .....	40
<b>3.14.1</b>	<b>Os diagramas de modelagem UML .....</b>	<b>41</b>
3.14.1.1	O diagrama de casos de uso .....	43
3.14.1.2	O diagrama de classes.....	44
3.14.1.3	O diagrama de atividades .....	45
3.15	ANÁLISE E LEVANTAMENTO DE REQUISITOS.....	46
3.16	TESTE DE SOFTWARE .....	47
3.17.1.1	Níveis de teste de software.....	51
3.17.1.2	Técnicas de teste de software.....	53
3.17.1.2.1	Teste de Carga.....	53
3.17.1.2.2	Teste de Funcional.....	53
4	TRABALHOS CORRELATOS .....	54
5	METODOLOGIA.....	55
5.1	FERRAMENTAS UTILIZADAS.....	56
<b>5.1.1</b>	<b>Hardware .....</b>	<b>56</b>
<b>5.1.2</b>	<b>Softwares.....</b>	<b>56</b>
5.2	LINGUAGENS DE PROGRAMAÇÃO E MODELAGEM.....	58
5.3	MODELAGEM DO APLICATIVO.....	58
<b>5.3.1</b>	<b>Diagrama de classes .....</b>	<b>59</b>
<b>5.3.2</b>	<b>Diagrama de atividades.....</b>	<b>60</b>
<b>5.3.3</b>	<b>Diagrama de casos de uso.....</b>	<b>61</b>
5.4	DESENVOLVIMENTO DO APLICATIVO .....	62
<b>5.4.1</b>	<b>Escolha da plataforma.....</b>	<b>62</b>
6	RESULTADOS E DISCUSSÕES .....	63

6.1 FUNCIONAMENTO.....	63
6.2 BUSCA E DOWNLOAD DO APLICATIVO .....	76
7 CONSIDERAÇÕES FINAIS .....	78
8 TRABALHOS FUTUROS .....	80
9 SUBMISSÕES E PUBLICAÇÕES.....	81
10 REFERÊNCIAS.....	82
APÊNDICE A.....	88

## 1 INTRODUÇÃO

A fala do paciente com fissura labiopalatina pode apresentar alterações, mesmo após as correções primárias do lábio e do palato. As alterações decorrentes da falha do funcionamento velofaríngeo, mais especificamente envolvem hipernasalidade, emissão de ar nasal, fraca pressão intraoral e uso de ponto articulatório atípico como as articulações compensatórias (AC) (PEGORARO-KROOK et al., 2010; BAYLIS; MUNSON; MOLLER, 2011) caracterizando a sintomatologia associada à disfunção velofaríngea (DVF).

A presença da DVF é diagnosticada através da avaliação fonoaudiológica perceptivo-auditiva da fala e pode ser complementada com avaliação instrumental (nasofaringoscopia e videofluoroscopia) (SWEENEY; SELL, 2008; KUMMER, 2014), possibilitando a definição da melhor conduta de tratamento desta condição. As abordagens de gerenciamento da DVF incluem procedimentos físicos (como repalatoplastias e faringoplastias ou a prótese de palato com bulbo faríngeo), e comportamentais (como fonoterapia para correção das ACs ou para correção dos erros de funcionamento velofaríngeo). A fonoterapia, por sua vez, pode ser associada ao uso da prótese de palato (PEGORARO-KROOK; AFERRI; UEMEOKA, 2009; PEGORARO-KROOK et al., 2010; BISPO et al., 2011).

A correção das articulações compensatórias e dos erros do funcionamento velofaríngeo (como a velofarínge hipodinâmica, por exemplo) envolve o uso de diversas estratégias fonoterapêuticas, tendo como base a sistematização do fechamento velofaríngeo durante sons orais (SCHNEIDER; SHPRINTZEN, 1980; KUEHN; MOLLER, 2000).

No Hospital de Reabilitação de Anomalias Craniofaciais da Universidade de São Paulo (HRAC/USP) é oferecido um Programa de Fonoterapia Intensiva (PFI) no Serviço de Prótese de Palato, em que os pacientes recebem entre três e quatro atendimentos diários em módulos de três à quatro semanas, os quais podem ser repetidos até obter-se a correção das alterações de fala. Assim como proposto por Schneider(1980) e colaboradores, o PFI do Serviço de Prótese de Palato do HRAC (PFI-SPP-HRAC) também requer um envolvimento intenso do paciente e/ou seus

familiares de forma a substituir o padrão velho inadequado da fala (com presença de AC e velofaringe hipodinâmica), e pelo padrão novo o qual requer treino diário e consistente.

Durante os atendimentos sob supervisão dos terapeutas o paciente pratica a fala nova (sem os erros articulatórios e com fechamento adequado da velofaringe) monitorando os resultados e seguindo uma proposta de exercícios hierarquizada de acordo com a complexidade fonético-fonológica desenvolvida na terapia. Nos intervalos entre os atendimentos diários, e também nos intervalos entre os módulos de terapia recomenda-se a prática consistente dos exercícios, o que, no entanto, é realizado sem o acompanhamento do terapeuta.

O desenvolvimento da computação móvel tem proporcionado o surgimento de novas tecnologias e aplicações. Em especial, uma das aplicações possíveis é o seu uso nas áreas da saúde e em ambientes hospitalares.

O uso de um aplicativo móvel pode favorecer a prática dos treinos de fala feitos nos intervalos entre os atendimentos diários e entre os módulos.

Um aplicativo móvel, ou “app” como é denominado entre os usuários, é um *software* desenvolvido para ser instalado em um dispositivo eletrônico móvel, como um telefone celular, por exemplo (BASTOS, 2014). Este app pode ser obtido pelo usuário (de forma gratuita ou comprado em lojas *online*).

Considerando que o HRAC da USP proporciona programas de fonoterapia intensiva, e ao refletirmos que a Tecnologia da Informação (TI) com todos os seus recursos está cada vez mais contribuindo para as áreas da saúde, conhecer quais contribuições um aplicativo móvel pode trazer para o tratamento terapêutico, é de grande interesse para a Fonoaudiologia.

Esta pesquisa teve como proposta desenvolver um aplicativo para auxiliar no processo fonoterapêutico favorecendo a execução dos treinos, e o uso do novo padrão de fala durante o tratamento de fonoterapia intensiva, dando ao paciente e ao profissional da saúde maior flexibilidade no tratamento.

## 2 OBJETIVOS

A seguir será apresentado o objetivo geral e os objetivos específicos que norteia este trabalho.

### 2.1 OBJETIVO GERAL

Desenvolver um aplicativo que auxilie no processo fonoterapêutico favorecendo a execução dos treinos e o uso do novo padrão de fala durante tratamento de fonoterapia intensiva.

### 2.2 OBJETIVOS ESPECÍFICOS

- a. Pesquisar sobre fissura labiopalatina, alterações decorrentes da falha do funcionamento velofaríngeo, hipernasalidade, disfunção velofaríngea, descrever a evolução da informática na área da saúde entre outros termos que norteiam o desenvolvimento desta pesquisa;
- b. investigar a estrutura de sistemas computacionais para dispositivos móveis;
- c. estudar a linguagem de programa Java Android que será utilizada para o desenvolvimento da proposta;
- d. modelar a proposta a ser desenvolvida utilizando a linguagem de modelagem unificada (UML);
- e. projetar o banco de dados Sqlite, que servirá para armazenar os alarmes, datas, horário, se o exercício foi realizado ou não;
- f. utilizar estratégias de IHC (Interface Homem/Computador), usabilidade, recursos multimídia que garantam a qualidade na construção do aplicativo;
- g. desenvolver o aplicativo proposto;
- h. avaliar o sistema desenvolvido junto aos Centrinho da USP e propor melhorias;



### 3 REVISÃO DA LITERATURA

A seguir encontra-se o referencial teórico utilizado para o desenvolvimento desta pesquisa.

#### 3.1 FALA

Uma das funções do complexo processo da comunicação é a fala, sendo uma das formas de manifestação da linguagem. A fala proporciona se expressar, conversar, expressar seus pensamentos com um interlocutor. Para que a fala seja concretizada, aspectos como organização e planejamento do ato motor são fundamentais, ligados à integridade auditiva e neuromuscular, além da normalidade anatômica dos órgãos responsáveis pela sua produção (GENARO; FUKUSHIRO; SUGUIMOTO, 2007).

Ainda segundo os autores citados anteriormente, quando o indivíduo tem distúrbio de fala, geralmente repercute negativamente junto a sociedade, no caso da fissura labiopalatina, justamente a fala é o aspecto mais estigmatizante. As alterações de fala causadas pela fissura labiopalatina são as mais variadas, elas podem ser desde uma leve distorção de algum fonema, causada por deformidade dentofacial, até o desenvolvimento da hipernasalidade, o que tornam a fala ininteligível.

Genaro, Fukushiro e Suguimoto (2007) ainda ressaltam que as alterações de fala causam fatores como difícil adaptação social, inibição e desconforto, isso tudo compromete a autoestima, que por sua vez, é influenciada pelos valores socioculturais do meio em que o indivíduo vive.

#### 3.2 FISSURA LABIOPALATINA

Fissuras labiais e/ou palatinas são malformações congênitas, caracterizadas por uma falta de fusão do lábio superior e/ou palato, má formação mais frequente da região da cabeça e pescoço e uma das anomalias mais frequentes em geral (BAPTISTA, 2007).

Grande parte das alterações da fala presentes na fissura palatina relaciona-se direta ou indiretamente à disfunção no mecanismo velofaríngeo, o

qual pode comprometer o desenvolvimento da fala quanto aos aspectos fonéticos e fonológico (GENARO; FUKUSHIRO; SUGUIMOTO, 2007).

Os autores acima citados complementam que os distúrbios compensatórios decorrentes da fissura labiopalatina começam a se desenvolver na fase de aquisição da linguagem oral, devido a diversos fatores físicos desencadeadores, tais como palato aberto, fistulas ou insuficiência velofaríngea; ou ainda, fatores fisiológicos agravantes ou mantenedores, fatores esses como a disfunção velofaríngea por erro de aprendizado, inabilidades linguísticas, deficiência auditiva dentre outras.

### 3.3 DISFUNÇÃO VELOFARÍNGEA

O termo Disfunção Velofaríngea (DVF) é empregado na literatura para designar, de uma forma genérica, qualquer tipo de função velofaríngea anormal. Porém, também podem ser encontrados na literatura os termos incompetência e insuficiência velofaríngea (GENARO; FUKUSHIRO; SUGUIMOTO, 2007).

A DVF é motivo de grande preocupação dos profissionais de fonoaudiologia que atuam na prevenção, na avaliação, no diagnóstico e na reabilitação das alterações na produção de fala associadas às malformações craniofaciais. Dentre as alterações de fala que podem ser encontradas nesses sujeitos, destacam-se as articulações compensatórias (AC), e que devido às implicações podem prejudicar na vida social, educacional e até mesmo profissional dos sujeitos (MARINO; DUTKA; KROOK; GREGIO, 2011).

Para os autores referidos acima, um mecanismo velofaríngeo competente, decorrente da ação muscular sinérgica das estruturas do véu palatino e da faringe, é uma condição fundamental para que haja uma ressonância oro nasal equilibrada e para a geração de uma pressão aérea intra-oral adequada durante a fala.

### 3.4 TERAPIA EM FISSURA LABIOPALATINA

O tratamento fonoaudiológico de um paciente com fissura labiopalatina deve ser iniciado ainda na maternidade, pois o bebê apresentará alterações

nas funções vitais de deglutição, sucção e respiração. A família fica ansiosa devido à dificuldade de alimentar o bebê e esta tensão dificulta ainda mais o ato. O fonoaudiólogo deve orientar a amamentação natural, porém, em casos onde isso não é possível, deve orientar o bico de mamadeira ortodôntico, por ser mais semelhante ao bico do seio materno. A postura da amamentação deve ser a mais ereta possível, a fim de evitar refluxo nasal, entrada de leite na tuba auditiva e engasgos. A mamada deve ocorrer dos dois lados proporcionando a estimulação e propriocepção também do lado fissurado (SILVA, 1999).

Silva (1999) propõe que o tratamento precoce aborde algumas áreas básicas:

- a) **alimentação**: deve ser usada a técnica que mais se aproxime da alimentação normal;
- b) **hábitos orais**: indica o uso de chupeta ortodôntica, a fim de evitar hábitos como sucção digital e para levar a língua a uma posição mais anterior;
- c) **sensibilidade**: deve ser trabalhada para fornecer estímulos sensoriais na parte anterior da cavidade oral, evitando que os movimentos compensatórios se fixem, influenciando na aquisição da fala;
- d) **linguagem e fala**: devem ser estimuladas ao máximo, pois a criança poderá ter atrasos;
- e) **audição**: a criança fissurada tem propensão a apresentar otites de repetição devido ao mau funcionamento do músculo tensor do véu palatino, podendo ter *déficits* auditivos que interferem no desenvolvimento da fala e linguagem.

A mãe tem que estar bem orientada sobre a amamentação que é de extrema importância para o bebê, e o profissional habilitado para passar as informações necessárias quanto os cuidados da alimentação é o fonoaudiólogo. Geralmente surgem dificuldades quanto ao fato da amamentação, e o aleitamento materno é imprescindível, pois auxilia na prevenção de infecções, combate a anemia e fortalecimento da musculatura da face e boca, além de manter a produção de leite da mãe (GOMES, 2009).

Gomes (2009) ainda complementa que a sucção é o início do tratamento fonoaudiológico.

### 3.5 A INFORMÁTICA NO CONTEXTO DA CLÍNICA FONOAUDIOLÓGICA

A informática tem se tornado cada dia mais presente em nossas vidas, e na realidade política, econômica, social, cultural, educacional, entre outras, e vem provocando mudanças muitas vezes significativas. As crianças estão crescendo e se desenvolvendo em meios informatizados, o computador, *smartphone*, *tablet* fazem parte do seu cotidiano e da sua cultura.

É possível encontrar a informática aplicada a diversas áreas do conhecimento. No campo da fonoaudiologia clínica, a informática também pode ser uma grande aliada no processo terapêutico, sendo incorporada a diferentes procedimentos clínicos fonoaudiológico, como na avaliação e no diagnóstico, no processo terapêutico, no arquivamento de informações, na comunicação entre profissionais, na comunicação entre pacientes, na atualização profissional e na parte administrativa e burocrática. Dantas (2006) destaca que o profissional de fonoaudiologia enquanto terapeuta deve enxergar o computador como um grande aliado, em vez do contrário, de procurar distanciá-lo do paciente. A autora ainda complementa dizendo que o computador quando bem utilizado, se torna um grande fator e pode trazer muitos benefícios ao tratamento. Seguindo essa linha de raciocínio, Forte (1998) acrescenta que o uso da tecnologia no ambiente terapêutico não tira o fonoaudiólogo de cena, pelo contrário, esse pode usufruir dessa tecnologia para conseguir novas perspectivas para desenvolvimento de seu trabalho clínico.

O uso da informática no meio clínico fonoaudiológico é datado a partir dos anos 80, época que o computador começou a ser levado em consideração, principalmente na aplicação no ambiente reeducacional e clínico. A prática foi utilizada em um primeiro momento para incentivar os pacientes com diagnósticos de distúrbios de leitura e escrita, pacientes esses que resistiam repetidamente, ao uso do papel do lápis (FOZ; BURSZTYN, 1998).

As autoras acima citadas complementam que atualmente, o uso do computador já está incorporado no ambiente clínico e, além de possibilitar o trabalho mais fácil com diferentes tipos de mídia, como a escrita, a oral, a

visual e a auditiva, permite também que o conhecimento seja construído pela interação, valorizando o papel do profissional de fonoaudiologia como um mediador do processo de aquisição e desenvolvimento da linguagem.

O trabalho terapêutico apoiado no uso computador, sendo um resultante de um processo dinâmico, estabelece um relacionamento contínuo entre três eixos. O primeiro dos três eixos é representado pelo paciente, que é o construtor do seu conhecimento e o mesmo que deve ser motivado a ampliar suas experiências. Nessa etapa surge a importância de um ambiente terapêutico adequado para o paciente se sentir acolhido, possibilitando o mesmo a expressar mais livremente e sem indução suas vontades e desejos, e também suas dificuldades sem medo de se sentir diferente (FORTE, 1998).

O segundo eixo segundo Forte (1998) tem a terapeuta como a mediadora do processo de aprendizagem do indivíduo. Após o fonoaudiólogo conhecer as dificuldades e facilidades de seu paciente, torna-se responsável pelo estabelecimento das condições favoráveis para a reabilitação de seu paciente, possibilitando que seus pacientes estabeleçam novos significados e ampliem as suas realizações, contribuindo para melhora de sua autoestima e no processo terapêutico.

Para complementar, Forte (1998) destaca que o terceiro e último eixo é representado pelo computador, utilizado como uma ferramenta para explorar a criatividade do paciente, sendo um elemento a mais para ajudar o profissional no tratamento de seu paciente.

Segundo Keske e Bernardi (2005), a informática na clínica fonoaudiológica pode contribuir de maneira positiva tanto na evolução do paciente quanto para a dinâmica terapêutica, uma vez que proporciona melhor desenvolvimento do paciente enriquecendo o processo terapêutico. Com isso os autores concluem que a informática nesse meio só acrescenta resultados positivos e satisfatórios.

Apesar das diversas possibilidades de inserção da Informática no contexto da clínica fonoaudiológica, o que ainda se encontra nas clínicas terapêuticas é a utilização baixa ou desordenada dos recursos disponíveis para favorecer o trabalho do fonoaudiólogo. No entanto, a fonoaudiologia não está totalmente alheia a esta nova era em que a informática domina as mais variadas áreas da vida humana, um exemplo disso é a crescente utilização das

tecnologias da informação no contexto da clínica fonoaudiologia, desde editores de texto até sistemas estatísticos com alto nível de complexidade (SANTOS; TRINDADE; FERNANDES; VIDOR, 2012).

### 3.5.1 A informática na terapia de voz e da fala

Os laboratórios de voz e fala, como são chamados os *softwares* desenvolvidos para o trabalho na área de voz e articulação, quando permitem a análise desses dois padrões, trazem muitos pontos positivos tanto para a avaliação quanto para terapia (CAPOVILLA, 1997).

Os avanços que têm sido alcançados são notoriamente visíveis quando são utilizados conhecimentos da informática para a área de avaliação da voz e da fala (CAMARGO; PINHO, 1998). Os autores reforçam que com o passar dos anos, os padrões de normalidade e os característicos dos quadros patológicos de voz e fala estão sendo estudados graças a tecnologia computacional.

Os estudos estão contribuindo principalmente para a área da análise acústica, melhorando a capacidade de comparar dados entre várias patologias.

## 3.6 DISPOSITIVOS MÓVEIS

Muito mais do que assistentes pessoais ou agendas eletrônicas, os dispositivos móveis passaram a ser computadores que podem ser facilmente levados a qualquer lugar, criados para atender profissionais e pessoas em movimento que necessitam de rapidez, facilidade e segurança no acesso a informações corporativas e pessoais. Além disso, as grandes inovações trazidas pela tecnologia *wireless* fizeram com que a indústria deste setor tenha crescido absurdamente nos últimos anos, tornando-se uma das mais eficientes e rápidas áreas tecnológicas do mundo, permitindo que as pessoas comuniquem-se de forma barata e fácil sem ficarem presas aos seus telefones ou computadores de mesa (LAUDON; LAUDON, 1999).

Quando são mencionados dispositivos móveis, é feita uma referência a aparelhos (Figura 1) que são encontrados facilmente no cotidiano das pessoas e que estão se tornando cada vez mais eficazes quando se fala de comunicação e, de preferência, *on-line*. Esses equipamentos permitem que os

usuários se desloquem junto com seu ambiente computacional e tenham um acesso constante às fontes de informações.

Figura 1 – Dispositivos móveis.



Fonte: Romeiro (2005).

Em se tratando especificamente do aparelho celular, há aproximadamente uma década foi um artigo de luxo, hoje em dia tornou-se um bem de consumo bastante acessível. À medida que a demanda por funcionalidades aumenta, as empresas de telefonia celular vêm acrescentando novas tecnologias a tais aparelhos, variando seus tamanhos, fazendo modificações desde *hardware* até *software*, estimulando nos consumidores o desejo de possuir o mais recente modelo de um determinado celular.

Os celulares ainda possuem como principal função o serviço de voz, porém a transmissão de dados vem tornando cada vez mais frequente. Como exemplo pode-se citar a possibilidade de ter acesso a informação e serviços a qualquer hora acessando a Internet. Pode-se dizer que os celulares estão tornando-se pequenos computadores, possuindo a vantagem do tamanho reduzido (ROMEIRO, 2005).

### 3.6.1 Vantagens dos dispositivos móveis

Do ponto de vista empresarial, os dispositivos móveis são ótimos geradores de informação, podendo ser utilizado na automatização do processo, até nas coletas de informações estratégicas, pois com suas reduzidas dimensões podem ser transportados e estar presentes em todas as situações em que um profissional pode atuar (SCHAEFER, 2004).

Romeiro (2005) destaca algumas vantagens dos dispositivos móveis em relação aos microcomputadores:

- a) **Tamanho:** bastante reduzidos e muito mais leves do que os PCs, podendo ser transportados de forma muito mais prática;
- b) **Fácil manuseio:** os dispositivos móveis possuem uma *interface* gráfica simples de manusear se comparado aos computadores;
- c) **Consumo de energia:** por serem menores e mais econômicos gastam menos energia que os computadores visto que o tempo de recarga é menor;
- d) **Custos operacionais:** como os dispositivos móveis são mais compactos e possuem atividades específicas, estes aparelhos não possuem alguns periféricos internos, como discos rígidos e discos flexíveis, diminuindo consideravelmente os custos com a manutenção.

Segundo o autor supracitado, outra característica que ajuda no desenvolvimento da comunicação sem fio é o fato de que as pessoas estão cada vez mais dependentes das informações disponibilizadas na Internet, o que antes poderia ser feito apenas via terminal remoto, agora pode ser acessado via dispositivo móvel. A tecnologia sem fio disponibiliza ao usuário a possibilidade de obter informações que lhes sejam úteis, a qualquer momento ou qualquer lugar.

### 3.7 ARQUITETURA DE SOFTWARE PARA DISPOSITIVOS MÓVEIS

A arquitetura de *software* para dispositivos móveis consiste em aplicações elaboradas para serem executadas no ambiente do dispositivo



móvel nativo. Essas arquiteturas devem respeitar as regras de cada ambiente se considerando os diversos dispositivos existentes no mercado atualmente, como Android, IOS, Windows Phone, entre outros. Executando nativamente, a aplicação consegue-se acessar todos os recursos que a estrutura concede, como por exemplo: GPS, câmera, acelerômetro, lista de contatos, entre outros. Consequentemente seguindo essa estrutura nativa, abre-se um leque muito grande e vasto de possibilidades para aplicações (BUDEL; MOLOSSI, 2011).

### **3.7.1 Mobilidade**

É a capacidade de poder se deslocar ou ser deslocado facilmente, no contexto da computação móvel, são dispositivos funcionalmente poderosos, oferecendo a capacidade de realizar facilmente um conjunto de funções, aplicações, sendo também capazes de conectar-se, obtendo e fornecendo dados segundo Lee, Schneider e Schell (2005).

Romeiro (2005) enfatiza que a capacidade de poder continuar uma comunicação e manter o envio de dados constante mesmo quando em movimento pode ser considerada uma das melhores vantagens de um dispositivo móvel.

### **3.7.2 Desenvolvimento de aplicações móveis**

As aplicações móveis são projetadas, desenvolvidas e implantadas dentro de um contexto por razões de negócio, melhorando a produtividade, o aumento de precisão e outras métricas, precisando também ser integradas a aplicações existentes. Nesse sentido podemos questionar: O Desenvolvimento de Aplicações Móveis é Fácil? Provavelmente é mais difícil, havendo muitas dificuldades que precisam ser vencidas, incluindo ergonomia, conectividade e considerações sobre telas de tamanho reduzido (FERNANDES; SILVA; ARAÚJO, 2010).

Ainda, em conformidade com os autores anteriormente citados, parte do custo do desenvolvimento de aplicações e dispositivos corresponde simplesmente às despesas intrínsecas de produzir dispositivos

tecnologicamente avançados. Além disso, é preciso desenvolver e testar uma nova aplicação móvel em diversos dispositivos, o que aumentará o custo total.

Em outras palavras a mobilidade significa que as pessoas estão usando dispositivos portáteis e funcionalmente poderosos. Esses dispositivos permitem que os usuários realizem um conjunto de funções de aplicação quando desconectados e que também sejam capazes de se conectar para recuperar dados e fornecê-los a outros usuários, aplicações e sistemas (FERNANDES; FERNANDES; SILVA; ARAÚJO, 2010).

### 3.8 PLATAFORMA ANDROID

A seguir será apresentado um pouco do histórico da plataforma Android.

#### 3.8.1 Histórico

Em 5 de novembro de 2007, o OHA (Open Handset Alliance) anunciou a plataforma Android. O Open Handset Alliance é um grupo de mais de 30 empresas que desenvolvem essa plataforma. As empresas dessa aliança estão trabalhando juntas para oferecer uma plataforma de desenvolvimento que permita aos desenvolvedores implementarem e estenderem as aplicações dos seus dispositivos móveis (AQUINO, 2007).

Aquino (2007) destaca que a aliança OHA é composta por um grupo bastante heterogêneo de empresas, que compreende desde operadoras de celular a fabricantes de *handsets*. Dessa aliança, fazem parte: operadoras de celular, companhias de semicondutores, fabricantes de *handset*, companhias de *software*.

De acordo com Rabello (2000?), o OHA é um grupo de empresas desenvolvedoras de tecnologia, que se uniram para acelerar a inovação e oferecer aos consumidores um celular mais barato e com maior usabilidade e mobilidade.

### 3.8.2 Definição

O Android é um sistema operacional para dispositivos móveis, baseado em uma plataforma de código aberta sob a licença apache, permitindo que os fabricantes possam modificar seu código fonte, essa liberdade permite adicionar novos recursos e incorporá-los ao sistema, desta forma o software evolui constantemente. O sistema possui cerca de 12 milhões de linhas de código, sendo: 3 milhões em XML; 2.8 milhões em C; 2.1 milhões em Java; e 1.75 milhões em C++. (LECHETA, 2010).

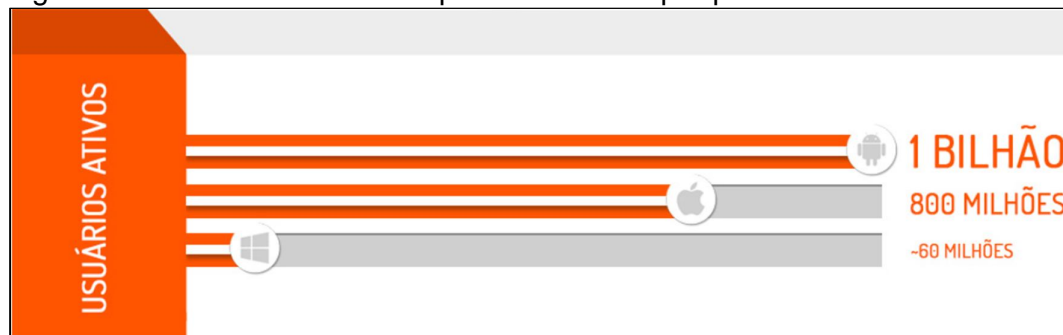
Aquino (2007) define o Android como uma pilha de *softwares* para dispositivos móveis que inclui um sistema operacional, um *middleware* e um conjunto de aplicações chaves.

Ainda segundo Aquino (2007), as aplicações para essa plataforma são escritas usando a linguagem de programação Java e executam sobre o Dalvik, uma máquina virtual customizada para dispositivos com restrições de recursos, como pouca capacidade computacional, baixa capacidade de armazenamento e baterias com baixo nível de energia.

O Google disponibiliza um *site* onde é possível encontrar tutoriais, arquivos de ajuda e programas necessários para desenvolver uma aplicação para Android (LAET, 2010).

A plataforma com o maior número de usuários ativos é o Android, seguida do iOS e Windows Phone, como pode ser observado na pesquisa realizada em 2014 e retratada na Figura 2.

Figura 2 - Usuários ativos de dispositivos mobile por plataforma.



Fonte: Hamann (2014).

### 3.8.3 Arquitetura Android

Segundo Laet (2010), a arquitetura do Android é dividida em cinco camadas: Kernel, Bibliotecas, Runtime, Framework de Aplicação e Aplicativos, conforme ilustra a Figura 3.

Figura 3 - Arquitetura da plataforma Android.



Fonte: Android a nova plataforma móvel – Parte II. ([2000?]).

#### 3.8.3.1 Kernel

A tradução da palavra Kernel em português significa núcleo. Na informática o kernel é a parte principal do sistema operacional. Sua função é fazer a comunicação entre o *software* e o *hardware*, fazendo com que ambos tenham uma excelente comunicação. Os dispositivos móveis funcionam corretamente graças ao sistema operacional que gerencia todos os componentes e fornece aos programas do usuário uma *interface* com o *hardware* mais simples (TANENBAUM, 2003).

Aquino (2007) ressalta que a arquitetura do Android é baseada no kernel do GNU/Linux, versão 2.6. O kernel do sistema funciona como uma camada de abstração entre o *hardware* e o restante da pilha de *softwares* da plataforma. O

kernel GNU/Linux possui vários recursos necessários para a execução de aplicações, como gerenciamento de memória, gerenciamento de processos, pilha de protocolos de rede, módulo de segurança e vários módulos do núcleo de infraestrutura. Como o sistema operacional é conhecido, também facilita o surgimento de melhorias aos *drivers* já existentes.

### 3.8.3.2 Bibliotecas

A plataforma Android conta com várias bibliotecas utilizadas por vários componentes do sistema. As bibliotecas são disponibilizadas de forma gratuita para que os desenvolvedores possam realizar facilmente algumas rotinas básicas, tais como: mostrar mensagens, realizar cálculos aritméticos entre outros. Elas também são as responsáveis pelas funções que o Android obtém, como manipulação de imagens, sons, vídeos, banco de dados etc. (LAET, 2010).

Aquino (2007) cita que o Android inclui um conjunto de bibliotecas C/C++ usados pelos vários componentes do sistema. As funcionalidades são expostas através do *framework* do Android.

A autora supracitada ainda destaca algumas das bibliotecas do núcleo:

- a) **Surface Manager**: Controla e gerencia o acesso ao subsistema de display. Compõe transparentemente camadas gráficas 2D e 3D de múltiplas aplicações.
- b) **3D libraries**: Uma implementação baseada na especificação do OpenGL 1.0.
- c) **SGL**: Biblioteca usada para compor gráficos 2D.
- d) **Media libraries**: Essas bibliotecas suportam playback e gravação de muitos formatos de áudio e de vídeo, bem como imagens estáticas, incluindo MPEG4, H.264, MP3, AAC, AMR, JPG e PNG.
- e) **FreeType**: É uma biblioteca usada para renderizar fontes.
- f) **SSL**: Fornece encriptação de dados enviados pela Internet.
- g) **SQLite**: É uma biblioteca C que implementa um banco de dados SQL embutido. Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo RDBMS em separado. SQLite não é uma biblioteca de cliente usada para

conectar com um servidor de banco de dados, mas sim o servidor. A biblioteca SQLite lê e escreve diretamente para e do arquivo do banco de dados no disco.

- h) **LibWebCore**: Biblioteca base para o navegador do Android e visões da web.
- i) **System C Library**: Uma implementação do libc derivada do BSD.

### 3.8.3.3 Runtime

Os aplicativos Android possuem a classe *Runtime* que permite a outros aplicativos do próprio sistema interagirem em tempo de execução com o seu aplicativo. A plataforma é composta pela máquina virtual Dalvik. Qualquer aplicação em Android roda dentro de seu próprio processo. Esta máquina virtual foi desenvolvida para que os dispositivos possam suportar múltiplas máquinas virtuais eficientemente. Estas máquinas virtuais executam arquivos com extensão “.dex”. Um arquivo com extensão “.dex” nada mais é do que uma espécie de *bytecodes* de Java otimizados para o Android (RABELLO, [2000?]).

De acordo com Aquino (2007), toda aplicação Android executa seu próprio processo, com sua própria instância da máquina virtual Dalvik.

Ainda, em conformidade com o autor citado anteriormente, a Dalvik executa classes compiladas por um compilador da linguagem Java. Os arquivos “.class” gerados são transformados no formato “.dex” pela ferramenta dx, incluída no SDK (*Software Development Kit*) do Android. Esses arquivos “.dex” são executados pelo Dalvik. A máquina virtual Dalvik também usa o kernel do GNU/Linux para prover a funcionalidade de múltiplas *threads* e gerenciamento de memória de baixo nível. A Figura 4 ilustra o Android *Runtime*.

Figura 4- Arquitetura da plataforma Android.



Fonte: Aquino (2007).

#### 3.8.3.4 Framework de aplicação

O *Framework* é um conjunto de módulos integrados que visam o reaproveitamento de código, ajudando no aumento de produtividade na hora do desenvolvimento pelo desenvolvedor (PASSOS, 2009).

Passos (2009) relata que na plataforma Android esses *frameworks* são como uma caixa de ferramentas (chamados de toolkit), onde todas as aplicações fazem uso. Todas as aplicações em Android usam o mesmo *framework* e as mesmas APIs<sup>1</sup>. Alguns desses *frameworks* trabalham escondidos do programador, rodando em *background*.

Segundo Aquino (2007), os desenvolvedores têm acesso completo à mesma API que é usada pelas aplicações core da plataforma. A arquitetura da aplicação foi projetada para simplificar o reuso dos componentes. Qualquer componente pode publicar suas capacidades e quaisquer outros componentes podem então fazer uso dessas capacidades, sujeito às restrições de segurança reforçadas pelo *framework*. Esse mesmo mecanismo permite que os componentes sejam substituídos por outros em tempo de desenvolvimento. O fundamento de todas as aplicações do *framework* é um conjunto de serviços e sistemas, conforme pode ser observado na Figura 5.

---

<sup>1</sup> Conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla **API** refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos".

Figura 5 - Framework de aplicações.



Fonte: Aquino (2007).

### 3.9 JAVA

Java é uma linguagem de programação de alto nível, que segue o paradigma de programação orientada a objeto. Devido sua portabilidade, Java pode ser executada em diferentes ambientes operacionais. A linguagem também possui a presença de mecanismos de tratamento de exceções que tornam as aplicações mais robustas, não permitindo que elas falhem mesmo quando estão rodando sob condições anormais. As bibliotecas que fazem parte de Java definem interfaces portáveis (ROMEIRO, 2005).

A Linguagem Java foi concebida para utilização em pequenos dispositivos eletrônicos inteligentes, entretanto, com as dificuldades de financiamento desse setor na época e, principalmente, devido ao surgimento da Internet a partir de 1993, novas oportunidades apareceram e a Sun dedicou, então, maiores investimentos nessa área (FURGERI, 2002 citado por SILVA, 2006). A partir de 1995, a Sun anunciou o Java, não apenas como mais uma linguagem de programação, mas como uma nova plataforma de desenvolvimento.

Dessa forma, a linguagem de programação Java começou a ser utilizada na elaboração de páginas para a World Wide Web, proporcionando a produção de páginas com conteúdo interativo e dinâmico, inicialmente com o uso de Applets (aplicativos para smartphones) com imagens em movimento (FURGERI, 2002 citado por SILVA, 2006). Desde o desenvolvimento do Java, diversas versões da ferramenta surgiram. Atualmente, o principal kit de desenvolvimento em Java está na Versão 1.3, obviamente, de domínio da Sun. O Java pode atuar em conjunto com outras linguagens, como é o caso de HTML, em que as aplicações podem ser embutidas em documentos HTML, podendo ser transmitidas e utilizadas na Internet. Os programas escritos em



Java funcionam como um acessório que é colocado no computador do usuário no momento que ele acessa um site qualquer, isto é, o computador do usuário passa a executar um programa armazenado no servidor Web, que é transferido para sua máquina no momento do acesso.

Segundo Furgeri (2002 citado por SILVA, 2006), a linguagem Java também tem sido usada para a criação de processos automáticos na *Web*. O grande diferencial do Java em relação às outras linguagens de programação, refere-se ao fato de que ela foi concebida originalmente para ser usada no ambiente da *World Wide Web*.

Romeiro (2005) ressalta que a linguagem Java tem a intenção de ser usada em ambientes de rede/distribuídos. Nessa direção, grande ênfase tem sido colocada na segurança. O Java permite a construção de sistemas livres de vírus e adulterações. Com essas qualidades, Java pode ser utilizada para a criação de vários tipos de aplicativos, desde aplicações *standalone* (local) até aplicações designadas para serem controladas pelo *software* que as executa, tais como *applets*, *servlets* ou *midlets*.

### 3.10 BANCO DE DADOS

Antes de entrarmos na definição de banco de dados, é preciso esclarecer a diferença entre Informação e dados. Informação é qualquer fato ou conhecimento do mundo real e que pode ou não ser registrada/armazenada. Dado é a representação da informação, que pode estar registrado em papel, num quadro de aviso ou no disco rígido do computador. (FERRARI, 2010).

Bancos de dados são dados armazenados em documentos ou *softwares*. Antes da evolução da informática, os armazenamentos de dados eram feitos em papéis, pastas e etc.

Os bancos de dados (do inglês *Data Base*) são lugares eletrônicos que armazenam dados. (COSTA, 2011). Os bancos de dados podem ser comparados com arquivos físicos, que servem para guardar documentos, portanto, funcionam com a mesma ideia, porém, no meio virtual.

Para Geremia (2010), um banco de dados pode ser definido como uma coleção de dados ou registros relacionados. Esses registros ou dados são fatos que podem ser gravados e que possuem um significado implícito

representando aspectos do mundo real. A autora citada ainda complementa que um banco de dados é projetado, construído e povoado por dados atendendo a uma proposta específica, ou seja, segundo a necessidade do usuário ou de um grupo de usuários.

Os bancos de dados organizam as informações em tabelas. A estrutura de um banco de dados é feita a partir de tabelas que possuem pelo menos uma coluna, mas geralmente possuem bem mais. Por fim, pode-se definir que um banco de dados é uma coleção de uma ou mais tabelas que contem informações relacionadas, por isso o nome de banco de dados relacional, no qual as tabelas relacionam informações umas com as outras (COSTA, 2011).

Geremia (2010) esclarece que os bancos de dados podem ser de qualquer tamanho e de complexidade variável. Um exemplo de um banco de dados simples e menos complexo seria uma agenda telefônica, na qual provavelmente os registros armazenados nesse banco de dados seriam o nome, o endereço, e claro, o telefone de um grupo de pessoas. Mas podemos ter bancos de dados maiores e mais complexos, como por exemplo, os registros de uma biblioteca que tenha, por exemplo, um milhão de exemplares de livros, onde esses livros poderiam ter diferentes dados registrados como o título, o autor, o assunto, a edição, o ano de impressão, o número de páginas e outros dados que a biblioteca achar necessário.

### **3.10.1 Por que usar um banco de dados?**

A utilização de um banco de dados pode facilitar e muito a vida das pessoas interessadas nos dados registrados, além de proporcionar agilidade e disponibilidade de acesso as informações corretas e atualizadas a qualquer momento, mas o fato mais interessante do armazenamento digital é como podemos tratar esses dados: a) primeiramente, podemos armazená-los diretamente em um banco de dados; b) estando os dados guardados em um banco de dados, posteriormente, quando for necessário fazer uma consulta rápida (GEREMIA, 2010).

Geremia (2010) cita como exemplo, uma busca por um conjunto de dados, busca essa que pode ser feita em segundos, ou minutos, em um banco de dados digitalizado, dependendo do desempenho da máquina utilizada,

enquanto que, se a mesma busca fosse feita manualmente, poderia levar meses até todos os dados de interesse serem computados, o que exigiria um trabalho braçal muito grande e com certa probabilidade de haver erros.

### **3.10.2 Objetivos de um banco de dados**

A seguir são elencados os principais objetivos de um banco de dados.

#### *3.10.2.1 Acesso rápido aos dados*

Um sistema convencional de arquivos não atende de modo eficiente a recuperação de dados, porém, num sistema de banco de dados, basta escrever o programa necessário para realizar a consulta e obtém-se o resultado desejado em um tempo muito pequeno. Por exemplo, uma empresa deseja saber em qual região a densidade de seus clientes é maior. Se for feita uma busca por CEP nos arquivos da empresa, essa busca terá de ser feita manualmente, ou seja, o funcionário encarregado dessa pesquisa terá de separar manualmente cada cliente por CEP. Isso pode levar um tempo muito grande. Porém, se a busca for feita através de um banco de dados, esse tempo é reduzido consideravelmente, pois podemos ter acesso a todos os clientes de um mesmo CEP de uma única vez. Além disso, a probabilidade de erro diminui consideravelmente (GEREMIA, 2010).

#### *3.10.2.2 Redução de redundância e inconsistência de dados*

Num sistema que não possua banco de dados, cada aplicação possui seus próprios arquivos, o que leva a uma duplicidade considerável nos dados armazenados, seja em papel, ou em arquivos digitais que não tenham o formato e a organização de um banco de dados, o que acaba gerando um desperdício de espaço de armazenamento, pois uma mesma informação poderá ser gravada diversas vezes e em diferentes lugares, entretanto, em um banco de dados, basta gravar a informação uma única vez. O uso de banco de dados tende a eliminar também a inconsistência de dados, pois de acordo com

Geremia (2010), a gravação/atualização seria feita uma única vez e estaria disponível para todos os usuários.

#### 3.10.2.3 *Compartilhamento de dados*

De acordo com o autor supracitado, a partir de um banco de dados disponível em uma máquina, cotas isoladas de dados podem ser acessadas por vários usuários simultaneamente, para posteriormente serem utilizadas para diferentes fins.

#### 3.10.2.4 *Aplicação de restrições de segurança*

O autor anteriormente citado ainda destaca que um sistema de banco de dados pode restringir o acesso aos dados de acordo com a permissão de acesso de cada usuário. Por exemplo, um funcionário que é encarregado de fazer o controle do estoque de uma fábrica, teoricamente, não precisa ter acesso ao salário de cada funcionário dessa fábrica, já que salário é considerado uma informação com certo nível de confidencialidade. Portanto, o banco de dados pode restringir os dados que cada usuário vai acessar, tornando mais segura a manipulação dos dados.

#### 3.10.2.5 *Sistema Gerenciador de Banco de Dados*

O SGBD é um *software* responsável pelo gerenciamento, manutenção e compartilhamento de grandes repositórios de dados, ou seja, ele é usado para manter o banco de dados (HEUSER, 2009).

Para Date (2003):

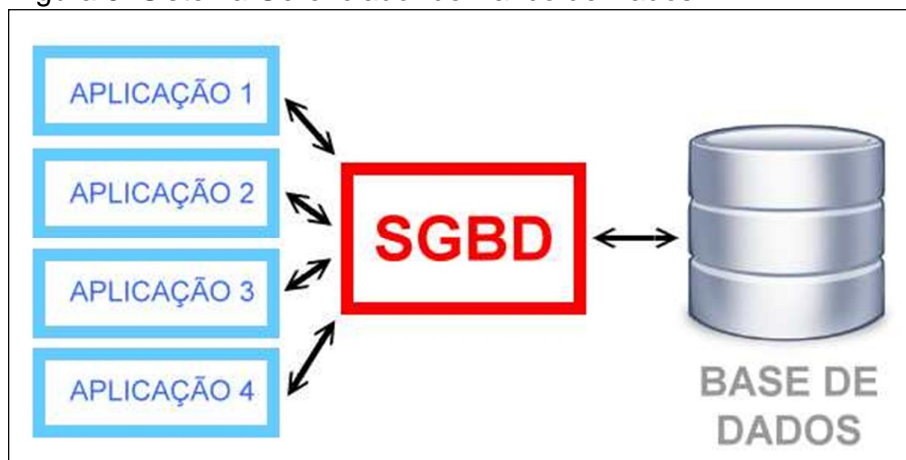
Um sistema de banco de dados é basicamente apenas um sistema computadorizado de manutenção de registros. O banco de dados, por si só, como o equivalente eletrônico de um armário de arquivamento, ou seja, ele é um repositório ou recipiente para uma coleção de arquivo de dados computadorizados (DATE, 2003, p. 3).

O SGBD mantém uma estrutura paralela aos objetos criados, que guardam todas as informações estruturais deste objeto, e estas informações estruturais são chamadas de metadados. Estes são úteis no processo de

migração ou atualização de um modelo de dados por conterem os dados sobre a precisão de um campo, o tipo de dado, se ele pode ser nulo ou não, além de muitas outras (DATE, 2003).

A Figura 6 ilustra um exemplo do funcionamento de um SGBD.

Figura 6- Sistema Gerenciador de Banco de Dados.



Fonte: Silva (2016).

Em 1974 a IBM foi responsável pelo desenvolvimento de uma linguagem de consulta estruturada chamada SQL (*Structured Query Language*), que passou a ser utilizada pela maioria dos SGBDs nas tarefas de manipulação de dados e regras de acesso ao banco. A partir de 1986 a SQL passou a ser padronizada pela American National Standards Institute (ANSI) e pela International Organization for Standardization (ISO).

Uma grande vantagem da linguagem SQL é que ela simplifica o acesso aos dados, pois toda a informação é passada aos SGBDs de forma textual. Essa característica, atrelada ao fato de o gerenciamento residir no próprio SGBD, facilita a adequação do modelo para diferentes plataformas e bancos de dados existentes (HUTH, 2002). Atualmente os Sistemas de Gerenciamento de Banco de Dados mais populares são os softwares proprietários Microsoft SQL-Server, IBM DB2 e Oracle, e os livres são o PostgreSQL, MySQL e o SQL-Lite (adequado a dispositivos móveis).

Estas técnicas, aplicadas ao contexto do desenvolvimento de um aplicativo mobile, podem por sua vez, contribuir significativamente nos estudos clínicos ligados às fissuras labiopalatinas.

### 3.10.3 SQLite

O SQLite é um banco de dados *Open Source*, utilizado nativamente no Android. O SQLite é um banco de dados que suporta padrão relacional como a sintaxe SQL. A grande vantagem do SQLite é seu pouco requerimento de memória em tempo de execução (aproximadamente 250 KBytes). Para utilizar o SQLite no Android não deve ser feita nenhuma configuração especial, apenas criar classes com instruções em linguagem SQL, as quais na verdade são criadas automaticamente pelo banco. Como qualquer outro banco de dados o SQLite suporta os principais tipos de dados: TEXT (formato texto, similar a *String* em Java), INTEGER (similar a *Long* em Java), REAL (similar a *Double* em Java) (VOGEL, 2011).

O SQLite é capaz de criar o arquivo em disco para ler e escrever. Esse arquivo criado possui a extensão “.db”. Por padrão, as bases de dados do SQLite são criadas e salvas no diretório “DATA/data/APP\_NAME/databasees/FILENAME.db, onde “APP\_NAME” é o nome da aplicação e “FILENAME” é o nome da base (GONÇALVES, 2011).

### 3.11 INTERFACE HOMEM-COMPUTADOR

Interação Homem-Computador (IHC), é a área da computação responsável por investigar o design, avaliar e implementar *interfaces*<sup>2</sup> para seres humanos visando que os mesmos tenham uma melhor usabilidade da maneira mais intuitiva e eficiente (SANTOS; TEIXEIRA, 2010).

Foi em meados dos anos 80 que o termo Interação Humano-Computador (IHC) foi adotado. O termo surgiu para descrever uma nova área de pesquisa. O foco de interesse do IHC é muito mais amplo que somente o *design de interfaces*, e sim abrange todos os aspectos relacionados com a interação entre usuários e computadores (ROCHA; BARANAUSKAS, 2003).

Pode-se destacar que um dos principais desafios que a IHC encara é o fato de acompanhar a evolução tecnológica. Devido ao rápido desenvolvimento

---

<sup>2</sup> Elemento que proporciona uma ligação física ou lógica entre dois sistemas ou partes de um sistema que não poderiam ser conectados diretamente.

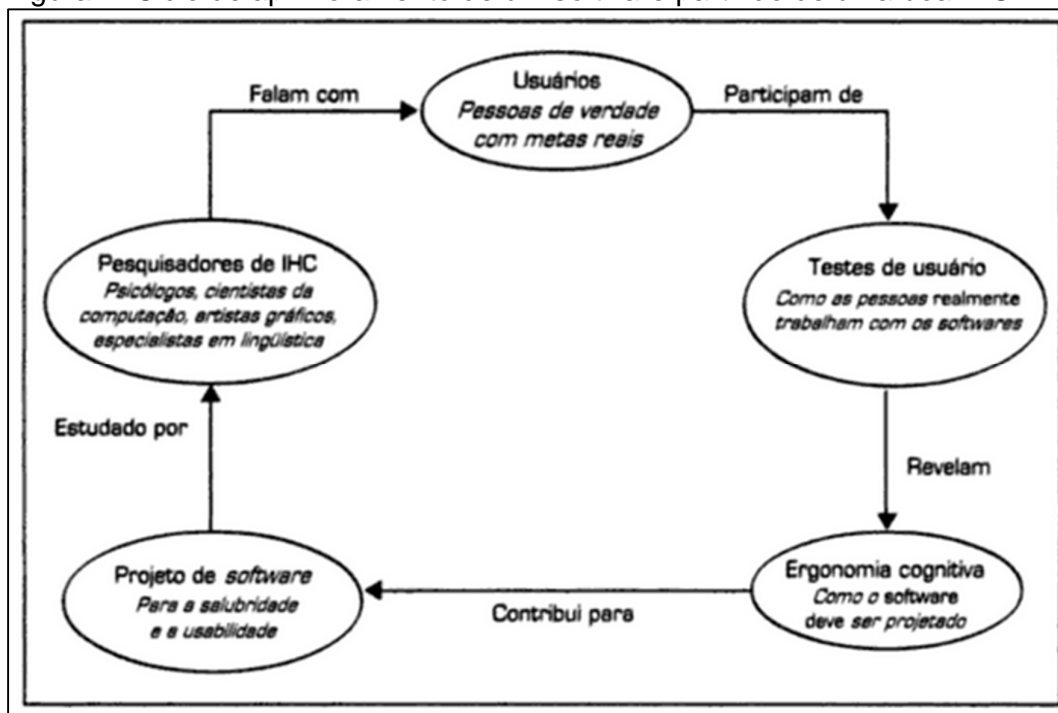
da tecnologia, sem dúvida alguma ela é uma área com muitos riscos e vários desafios (ROCHA; BARANAUSKAS, 2003).

A grande maioria dos usuários não consegue dominar por completo uma aplicação sem antes já aparecer muitas novas funcionalidades adicionais, que são constantemente adicionadas pela equipe responsável pela aplicação. Pensando nisso uma das grandes perguntas que a IHC faz é: Como adicionar novas funcionalidades sem excluir ou tornar difícil a vida dos seus usuários antigos? (OLIVEIRA, 2010).

Não podemos se prender com a associação errada de que IHC é apenas para computadores pessoais como o próprio nome sugere, uma vez que toda interação entre objetos distintos é feita através de uma *interface* (OLIVEIRA, 2010).

Também não se pode aceitar que a Interface Homem-Computador seja uma área exclusivamente da computação. Nessa área os horizontes se expandem além da computação, são produtos de trabalhos interdisciplinares, que integram diversos profissionais de campos distintos dentre os quais podem ser citados: projetistas gráficos, escritores, engenheiros ergonômicos, psicólogos, antropólogos, sociólogos e analistas de sistema, todos esses profissionais trabalham em busca de um objetivo de aperfeiçoar a aprendizagem da funcionalidade dos sistemas, gerando uma maior facilidade de aumento de produtividade para todos os usuários de um sistema. (FERNANDES, 2009). Assim, o desenvolvimento de uma boa *interface* não é resultado de um bom desenvolvedor de sistemas, mas sim do esforço coletivo de diversos indivíduos que podem contribuir com o produto final. Este ciclo pode ser observado na Figura 7 que demonstra a forma como uma *interface* pode ser refinada.

Figura 7- Ciclo de aprimoramento de um software partindo de uma boa IHC.



Fonte: BAWA (1997, p. 43.).

Para que os aplicativos se tornem amplamente aceitos e efetivamente usados, eles necessitam ser muito bem projetados. Isso não quer dizer que o *design* deve ser adequado a todas as pessoas, mas sim, que os aplicativos devem focar em um grupo alvo para solucionar suas necessidades, focando sempre nas capacitações e limitações desse grupo. Certamente, os usuários não devem ser obrigados a entender como funciona a programação do aplicativo, assim como o motorista não tem a obrigação de entender como funciona a mecânica de um carro, porém as posições dos pedais, direção e câmbio tem um impacto importantíssimo sobre o motorista, da mesma forma que o *design* de um aplicativo tem efeito sobre seus usuários (ROCHA; BARANAUSKAS, 2003).

As empresas produtoras de *software* tem despertado a algum tempo a ideia de que ao melhorar os aspectos físicos da *interface* do usuário proporciona maiores chances de sucesso no mercado. Com isso ao longo de tempo surgiu um novo termo que vem sendo muito utilizado – *interface* amigável ou sistema amigável (*user-friendly*). O significado do amigável nesse termo está associado ao fato dos elementos das telas serem esteticamente mais agradáveis ou bonitos (ROCHA; BARANAUSKAS, 2003).



A área da Ciência da Computação contribui promovendo conhecimento sobre o que existe de tecnologia possível, e oferece ideias de como usufruir da melhor maneira possível dessas tecnologias. (ROCHA; BARANAUSKAS, 2003).

### 3.12 SOFTWARE

De acordo com Pressman (2011), *software* pode ser definido como sendo; a) instruções que quando executadas, produzem a função e o desempenho desejados; b) estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; c) documentos que descrevem a operação e uso dos programas.

Sommerville (2012) define *softwares* como sendo programas de computador e documentação associada, sejam esses programas criados para um indivíduo ou para o mercado de *software* de forma geral. Também é importante ressaltar que um *software* considerado bom deve providenciar ao usuário funcionalidade e desempenho, além de ser confiável e fácil de manter e usar.

O *software* se difere do *hardware* através de uma simples palavra: físico. Enquanto o *hardware* possui uma parte totalmente física, que se consegue tocar no que está sendo construído, o *software* vai à contra partida, necessitando de uma abstração maior, já que o mesmo é impalpável.

Para Pressman (2011, p. 33), diferentemente do *hardware*, o “*software* não de desgasta”, justamente por não se tratar de uma peça palpável, não existe peças de reposição para um *software*, porém ao mesmo tempo em que o *software* não se desgasta ele sofre por outro lado - a depreciação, onde com o passar do tempo ele não atende mais as necessidades, tendo que ser remodelado ou até mesmo refeito.

O autor ainda enfatiza que no início da era tecnológica, a importância do *software*, conhecido como a parte lógica da computação, era pequena, mas com o passar do tempo a sua importância foi se tornando cada vez maior.

Embora no processo de desenvolvimento do *software* existam semelhanças ao de fabricação do *hardware*, são processos distintos, uma vez que o grande custo no desenvolvimento do *software* se concentra na parte de

engenharia, com isso pode-se afirmar que os projetos de *softwares* diferem, e não podem ser geridos como se fossem projetos de fabricação. (PRESSMAN, 2011).

Pressman (2011) segue destacando que o *software* tem o papel de distribuir o produto mais importante na nossa era, que é a informação, tem o poder de transformar dados pessoais, como por exemplo uma transação financeira, de uma forma que possam ser mais úteis em um determinado contexto.

Ainda segundo Pressman (2011), nos dias atuais, indústrias de *softwares* se tornaram grandes dominantes na economia mundial, e são gerenciadas por grandes equipes especializadas em softwares, mudando a visão do antigo programador isolado e solitário de antigamente.

### 3.12.1 Características de um software

De acordo com Sommerville (2012) existem quatro características essenciais para que um *software* seja considerado bom. São essas: manutenibilidade, confiança e proteção, eficiência e aceitabilidade.

- a) **Manutenibilidade:** Trata-se do fator de manutenção de um *software*. De acordo com o autor, todo programa deve ter potencial de evoluir de acordo com a necessidade do usuário através de manutenção em sua estrutura. Assim, fica clara a importância da capacidade de crescimento para que um sistema resista ao tempo;
- b) **Confiança e proteção:** Estes são os aspectos que definem a segurança de um bom *software*. De acordo com o autor, um *software* confiável é aquele que não causa nenhum tipo de prejuízo em caso de falhas ou tentativas de intrusão.
- c) **Eficiência:** Esta característica é definida pelo bom uso dos recursos solicitados para o funcionamento do *software*, sem que haja desperdício de memória e tempo de processamento;
- d) **Aceitabilidade:** O *software*, ao ser desenvolvido, deve levar em consideração os outros recursos e componentes que são utilizados no cotidiano do usuário para o qual ele é designado. Assim, um bom *software* tem que ser compatível e compreensível para tal usuário.

Essas características podem ser observadas em todos os sistemas de *software* que são utilizados no dia-a-dia e que são conhecidos pela população em geral. Sendo esses aspectos intrínsecos dos programas consolidados na cultura de informação atual, pode-se afirmar que é primordial para qualquer *software* bem sucedido que cubra essas quatro características.

### 3.12.2 Tipos de software

Pressman (2000) define que um *software* pode ser aplicado em qualquer contexto em que uma determinada quantidade rotinas possa ser estabelecida, o que gera a necessidade de especificação ao desenvolver um novo programa, dada a abrangência em que pode ser contextualizado. O autor apresenta os seguintes tipos de *software* como exemplo de aplicações em potencial:

- a) **Software de sistema:** são coleções de *software* que servem como intermediário entre outros programas e a própria máquina. É característica desse tipo de *software* a interação com os componentes físicos de um computador através do gerenciamento dos recursos computacionais;
- b) **Software de tempo real:** Trata-se dos programas que tem a capacidade de monitoramento, análise e controle de eventos no mundo. O autor define como componentes desse tipo de *software* um sistema de coleta de dados e formatação desses dados para a linguagem compreensível do programa, um sistema de análise dos dados coletados, um componente de controle que permite que o programa responda ao mundo real e um componente de monitoramento que gerencia todos os outros;
- c) **Software de negócio:** O autor afirma que sistemas de processamento de informações empresariais compõe a maior de todas as áreas de aplicação de *software*. Esse tipo de *software* pode desempenhar incontáveis funções dentro de uma empresa, incluindo reestruturação de informação para facilitar a compreensão dos usuários, gerenciamento de recursos, conversão de dados e agentes de tomada de decisão;

- d) **Software científico e de engenharia:** De acordo com o autor, esse modelo de *software* é caracterizado pelos algoritmos de processamento numérico. Esse tipo de programa está presente em todo tipo de contexto acadêmico moderno, por exemplo, a astronomia e biologia molecular;
- e) **Sistemas embarcados:** Esse tipo de *software* está presente no cotidiano mais que qualquer outro, pois se trata dos sistemas de controle de aparelhos e automóveis que possuem algum tipo de interação digital com o usuário, como computadores de bordo e eletrodomésticos;
- f) **Software de uso pessoal:** O autor aponta para esse tipo de *software* como aquele que tem surgido desde a década de 90 como sendo o mais abrangente e variado entre todos os outros tipos, pois se trata de *softwares* que abordam todo tipo de necessidade pessoal, incluindo entretenimento, computação gráfica, multimídia e processadores de texto;
- g) **Software baseado em web:** Esse tipo de *software* possui como principal característica uma quantidade ilimitada de conteúdo, pois sua funcionalidade é de capturar dados através da rede para que estes sejam mostrados ao usuário;
- h) **Software de inteligência artificial:** Segundo o autor, esse tipo de *software* tem como funcionalidade a resolução de problemas complexos que não poderiam ser resolvidos através do sistema computacional convencional. Sistemas especialistas, redes neurais e sistemas de reconhecimento de padrões são exemplos da aplicação de *softwares* de inteligência artificial.

Sommerville (2012) ainda se refere a outros tipos de aplicação, como sistemas para modelagem e simulação, sistemas de coleta de dados, sistemas baseados em transações e sistemas de sistemas.

### 3.13 ENGENHARIA DE SOFTWARE

A engenharia de *software* é uma disciplina essencial para o estabelecimento de um conjunto de boas práticas que viabilizam a distribuição e comercialização de *softwares* empresariais e de uso pessoal, pois antes de ser explorada a fundo, não existiam garantias e segurança de que o acordo entre usuário e desenvolvedor seria mantido com sucesso.

Sommerville (2007) diz que a engenharia de *software* é uma disciplina que se ocupa de todos os aspectos da produção de um *software*, partindo dos estágios iniciais como especificação do sistema, e até estágios mais avançados que seria de manutenção desse sistema.

No entanto, Pressman (1995) classifica a engenharia de *software* como uma “tecnologia em camadas”. Todas as etapas da engenharia de *software* devem ser apoiadas pelo compromisso com a qualidade. Trata-se de uma pirâmide, onde a base seria a camada da qualidade, na próxima camada encontram-se os processos, acima dos processos - os métodos, e acima desses, as ferramentas, como podem ser observados na Figura 8.

Figura 8- Camadas da Engenharia de Software.



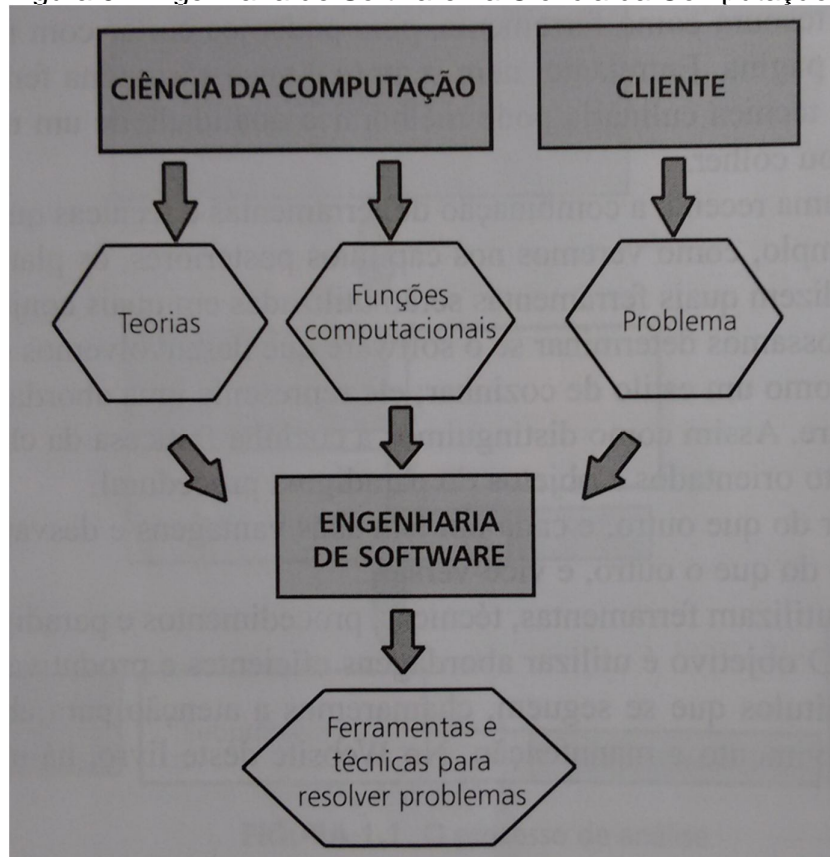
Fonte: Pressman (2011, p. 39).

Ainda segundo o autor, com o passar dos anos, a engenharia de *software* evoluiu e foram criados *softwares* para apoiar esse desenvolvimento. Esses *softwares* são bem vindos, porém, não excluem a necessidade de intervenção humana. Não existe um modelo ideal na engenharia de *software*, devido as suas divergências.

Sommerville (2003) complementa que todos os modelos compartilham de atividades fundamentais como a especificação, projeto e implementação, validação e evolução.

A engenharia de *software* utiliza de teorias e linguagens de programação como ferramentas para solucionar problemas. Ao invés de provar teorias ou pesquisar profundamente como funcionam os algoritmos, ela trata o computador como uma ferramenta para solucionar problemas. A Figura 9 ilustra este contexto.

Figura 9- Engenharia de Software na Ciência da Computação.



Fonte: Pfleeger (2007, p. 4).

As definições aqui apresentadas são importantes para compreender o papel da engenharia de software na atualidade, pois demonstra que a aplicação dessa disciplina ocorre em todas as fases do desenvolvimento – e de pós-desenvolvimento – de um programa ou sistema.

### 3.14 A LINGUAGEM DE MODELAGEM UNIFICADA (UML)

A linguagem de modelagem unificada pode ser considerada como uma peça fundamental para a boa engenharia de *software*, pois se trata de um

conjunto de diagramas de modelagem que expressam a intenção da empresa desenvolvedora em função do que será desenvolvido. Desta forma, contribui para uma boa ES no que se diz respeito ao planejamento pré-desenvolvimento.

De acordo com Guedes (2004), a UML é uma linguagem visual que almeja modelar sistemas computacionais através de vários tipos de diagramas que compõe a linguagem. A linguagem de modelagem, no entanto, possui independência da linguagem de programação e plataforma de desenvolvimento que serão utilizados no decorrer do projeto devido ao fato de que esses não são pontos considerados relevantes pela UML.

A UML (Unified Modeling Language) é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Ela poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software (BROOCH; JACOBSON; RUMBAUGH, 2006, p. 13).

A visão dos autores sobre a definição da UML é bastante pertinente por demonstrar que existe uma variedade de funcionalidades que pode ser utilizada com a linguagem em determinado contexto, além da importância de se fazer uso deste recurso sempre que houver a necessidade.

Guedes (2004) estabelece uma analogia entre a importância da UML com o processo que é utilizado por um engenheiro civil ao projetar um prédio de vários andares, apresentando o fato de que um edifício simplesmente não pode ser modelado de cabeça.

Da mesma forma que o autor demonstra a importância organizacional sobre a linguagem de modelagem unificada, também é possível levantar outros fatores decisivos que colaboram com esse ponto de vista, sendo um deles a possibilidade de se levantar estimativas de tempo e recursos que serão utilizados no decorrer do projeto partindo de seus modelos criados a partir do levantamento de requisitos.

### **3.14.1 Os diagramas de modelagem UML**

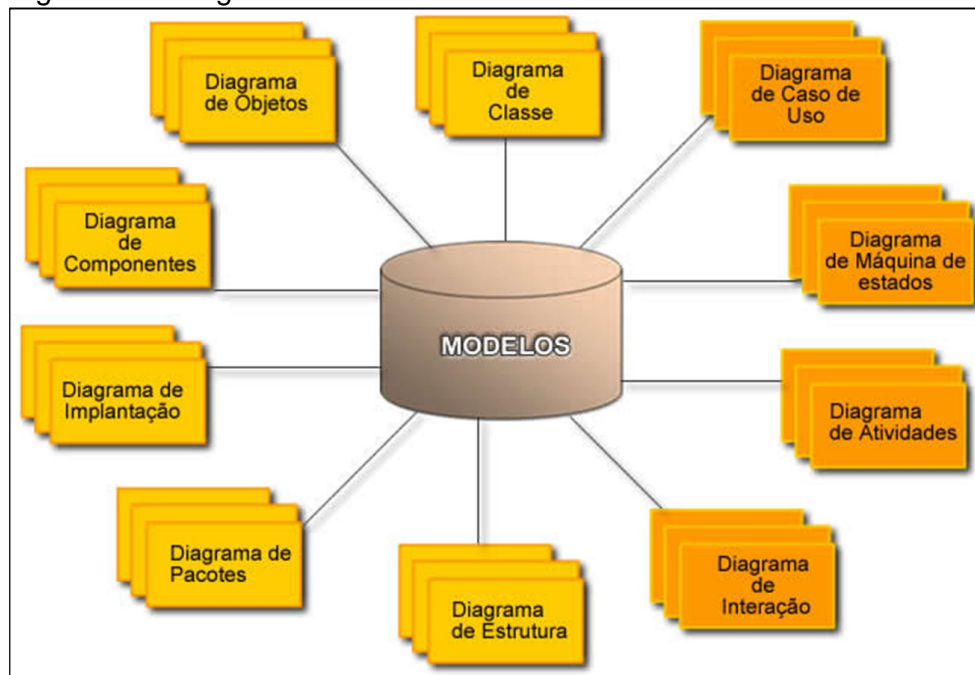
Existem 10 tipos de diagramas diferentes que compõe a linguagem de modelagem unificada, sendo cada um deles independente de qualquer outro, além de expressar uma função exclusiva e insubstituível pelos demais.

Os Diagramas da UML estão divididos em Estruturais e Comportamentais, conforme ilustra a Figura 10.

1. **Diagramas Estruturais:** de Classe; de Objeto; de Componentes; de implantação; de Pacotes; de Estrutura.
2. **Diagramas Comportamentais:** de Caso de Uso (Use Case); de Máquina de Estados; de Atividades; de Interação: Dividem-se em: de Sequência; Geral interação; de comunicação; de tempo.

Autores como Guedes (2004), Fowler (2005) e Bezerra (2015) apresentam os principais deles como sendo: Diagrama de Caso de Uso, Diagrama de Classes, Diagrama de Objetos, Diagrama de Sequência, Diagrama de Atividades, Diagrama de Estado, Diagrama de Componentes e Diagrama de Pacotes.

Figura 10 – Diagramas da UML.



Fonte: Martinez (2006-2016).

Alguns autores como Fowler (2005) ainda categorizam esses diagramas como sendo de estrutura, comportamento e interações. O autor ressalta que essa distinção é importante por deixar claro que não existe uma necessidade de se fazer uso de todos os tipos de diagramas disponíveis para todo e qualquer projeto.



### *3.14.1.1 O diagrama de casos de uso*

O Diagrama de Casos de Uso pode ser considerado como um dos mais famosos entre todos os tipos, por geralmente ser utilizado em todo tipo de projeto no decorrer de seu desenvolvimento.

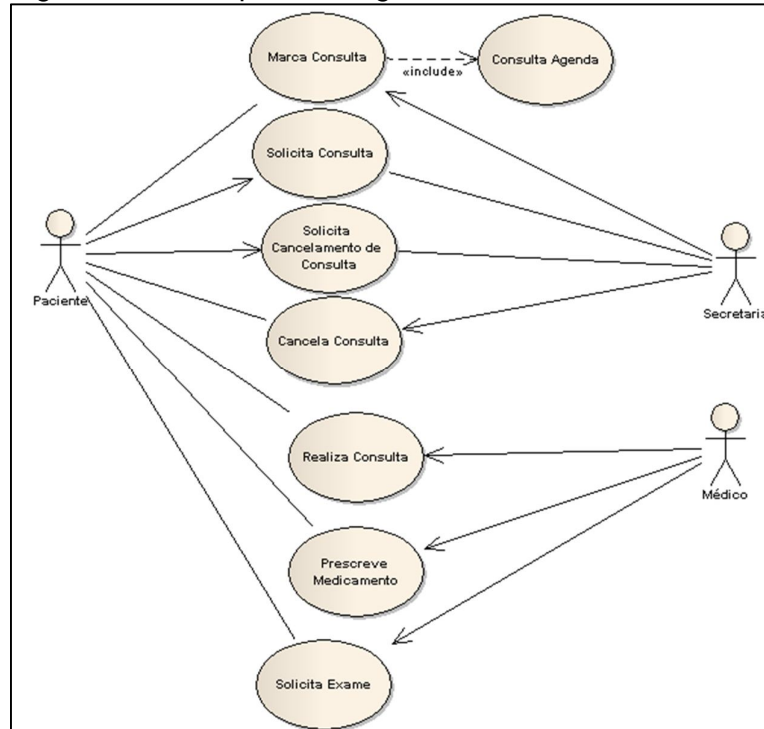
Guedes (2004) afirma que sua aplicação normalmente é feita para que seja consultada nas etapas de levantamento de requisitos, mas que também pode ser utilizado durante todas as fases do projeto.

De acordo com Bezerra (2015), a visão de casos de uso é criada nas etapas iniciais de um projeto e o direciona partindo de seu ponto de vista. O recurso é utilizado com o propósito de enxergar os componentes com uma visão externa, tal como suas interações e agentes que causam a interação.

Ambos os autores apontam para o fato de que o Diagrama de Casos de Uso é muito importante nas etapas iniciais do projeto, além do fato de servir de apoio e alinhamento para que outros diagramas sejam desenvolvidos. É importante notar que a visão externa providenciada por este modelo de diagrama tem a capacidade de demonstrar todas as interações e consequências sobre cada tentativa de tomada de decisão por parte do “ator” (indivíduo qualquer pertinente ao sistema representado pelo diagrama).

A Figura 11 ilustra um exemplo de Diagrama de Casos de Uso.

Figura 11– Exemplo de Diagrama de Casos de Uso.



Fonte: Elaborada pelo autor.

### 3.14.1.2 O diagrama de classes

O Diagrama de Classes é considerado por alguns autores como sendo o mais utilizado e o mais importante da UML. Isso se dá ao fato de que ele pode servir como apoio para ambos - o desenvolvimento do projeto e dos demais diagramas.

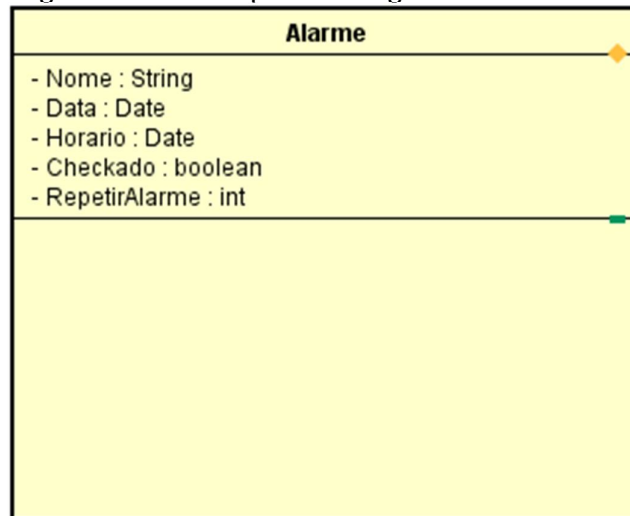
Guedes (2004, p. 27) define que “como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos possuídos por cada classe, além de estabelecer como as classes se relacionam e trocam informações entre si.”.

A definição do autor deixa claro que, embora a UML possa ser representada sobre qualquer linguagem de programação com a qual o sistema será desenvolvido, sem que haja necessidade de qualquer adaptação para implementação, ainda é necessário que haja algum conhecimento sobre o que será programado e sua estrutura.

Melo (2010) ainda alerta para o fato de que este diagrama pode ser desenvolvido após a extração dos requisitos para, que então sejam modeladas as classes do sistema.

A Figura 12 ilustra um exemplo de Diagrama de Classes.

Figura 12 – Exemplo de Diagrama de Classes.



Fonte: Elaborada pelo autor.

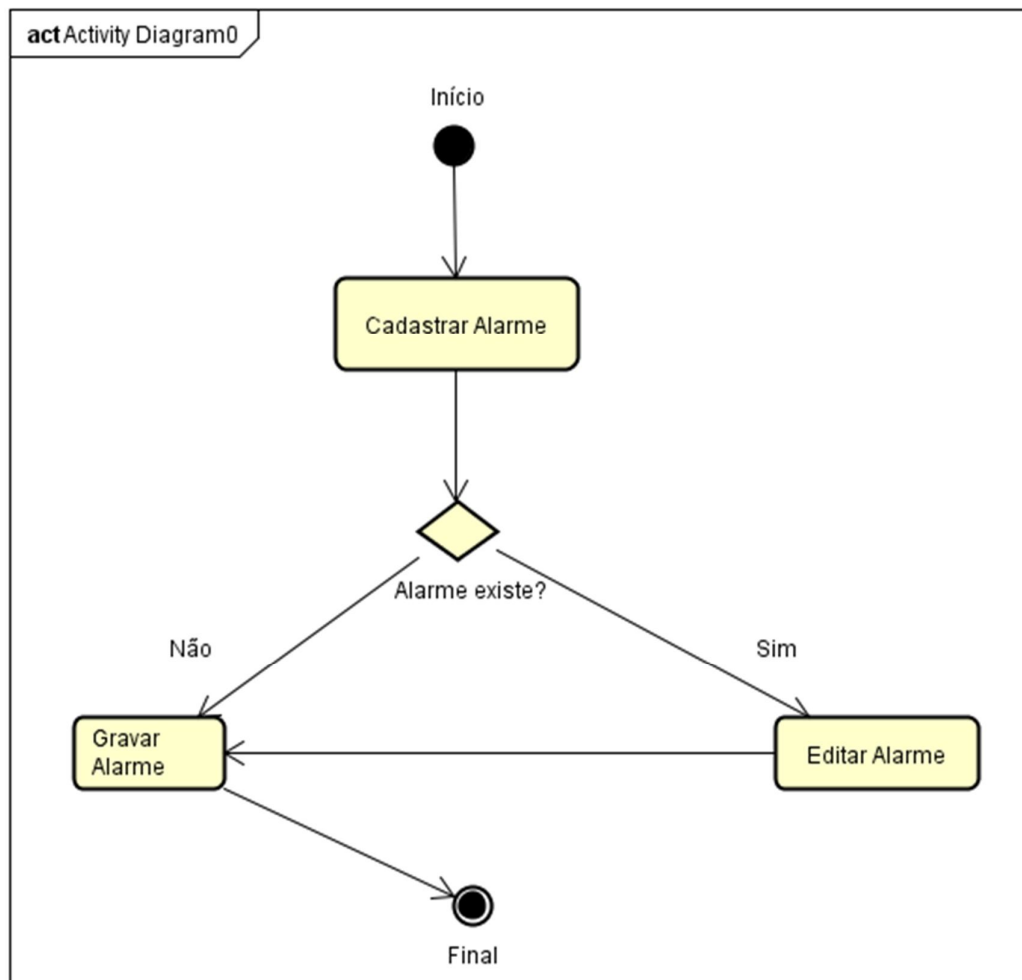
### 3.14.1.3 O diagrama de atividades

O Diagrama de Atividades se comporta como um fluxograma que representa detalhadamente as fases para que uma determinada atividade aconteça dentro do contexto de um sistema.

De acordo com o portal Microsoft Developer Network (MSDN), o Diagrama de Atividades é um fluxo de atividades que compõe uma atividade. No entanto, também é possível descrever fluxos de dados e esquemas mais complexos utilizando-o, embora não seja uma prática comum.

Guedes (2004) cita que o diagrama de atividades tem como principal foco descrever os passos de execução para que uma determinada atividade seja concluída. O autor também complementa que, embora não seja prática comum da engenharia de *software*, também é possível de se utilizá-lo para representar o sistema como um todo, indo além da premissa de materializar uma classe de atividade complexa. A Figura 13 ilustra um exemplo de Diagrama de Atividades.

Figura 13 – Exemplo de Diagrama de Atividades.



Fonte: Elaborada pelo autor.

### 3.15 ANÁLISE E LEVANTAMENTO DE REQUISITOS

Embora seja uma visão contra intuitiva, a fase de levantamento de requisitos vai muito além de obter as informações necessárias do ponto de vista do usuário para que o sistema seja projetado conforme as expectativas.

De acordo com Bezerra (2015), o objetivo de se realizar um levantamento dos requisitos é alinhar as metas dos desenvolvedores com os interesses dos usuários, para que então ambas as partes possam definir as necessidades que serão satisfeitas pelo sistema que será desenvolvido.

Assim, fica claro que essa etapa da construção de um projeto pode ser considerada como uma das primeiras que deve ser abordada, alinhando a

visão do desenvolvedor com o que é esperado pelos clientes na entrega do produto final.

Guedes (2004) se refere ao levantamento de requisitos como a fase em que o engenheiro de *software* tenta se aproximar da mentalidade do cliente, dessa forma compreendendo-o. De acordo com o autor, esse levantamento deve ser realizado a partir de uma quantia indeterminada de entrevistas nas quais a equipe de desenvolvimento deve auxiliar seus clientes com a definição de suas necessidades.

O autor ainda aponta para o fato de que a análise dos requisitos levantados é um subprocesso de extrema importância, pois deve ser levado em consideração todos os apontamentos feitos pelos usuários com o propósito de examinar, e verificar se não houve falhas no decorrer da comunicação nessa etapa inicial.

Outras questões de peso que surgem durante a fase de levantamento de requisitos são a importância de uma boa comunicação entre todas as partes envolvidas no projeto, bem como, o subjetivismo de alguns critérios que são levantados pelos clientes. Todos esses aspectos devem necessariamente ser tratados antes de que qualquer etapa de desenvolvimento prático comece a ser realizada sobre o que foi levantado.

### 3.16 TESTE DE SOFTWARE

Antes de iniciarmos uma discussão sobre teste de software precisamos esclarecer alguns conceitos relacionados a essa atividade. Inicialmente, precisamos conhecer a diferença entre defeitos, erros e falhas. As definições que iremos usar aqui seguem a terminologia padrão para Engenharia de Software do IEEE – Institute of Electrical and Electronics Engineers – (IEEE 610, 1990 citado por CLAUDIO ([201-])).

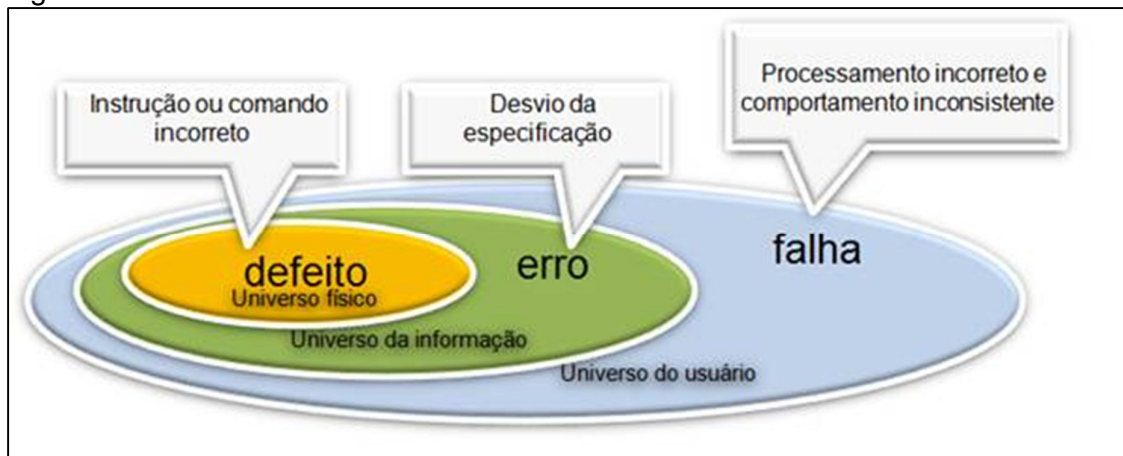
- a) **Defeito** é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto.
- b) **Erro** é uma manifestação concreta de um defeito num artefato de software. Diferença entre o valor obtido e o valor esperado, ou seja,

qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.

- c) **Falha** é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.

A Figura 14 expressa a diferença entre esses conceitos.

Figura 14– Defeito x erro x falha.



Fonte: Claudio ([201-]).

Como pode ser observado na Figura 14, os defeitos fazem parte do universo físico (a aplicação propriamente dita), e são causados por pessoas, por exemplo, através do mal uso de uma tecnologia, podendo ocasionar a manifestação de erros em um produto, ou seja, a construção de um *software* de forma diferente ao que foi especificado (universo de informação). Por fim, os erros geram falhas, que são comportamentos inesperados em um *software*, que afetam diretamente o usuário final da aplicação (universo do usuário), e podem inviabilizar a utilização de um software.

A etapa de teste de *software* (TS) é caracterizada pela revisão, de certo modo, de algum dos aspectos que foram desenvolvidos anteriormente. É essencial para a engenharia de *software* que o TS seja regulamentado de alguma forma com o objetivo de padronizar o controle de qualidade que é feito sobre o produto.

Segundo Claudio ([201-]), teste de *software* é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou

corretamente no ambiente para o qual foi projetado, ao mesmo tempo em que faz parte de todo o processo de engenharia de *software*. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, dizemos que sua natureza é “destrutiva”, e não “construtiva”, pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final.

Ainda segundo o autor, o conceito de teste de *software* pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. Existem atualmente várias definições para esse conceito. De uma forma simples, testar um *software* significa verificar através de uma execução controlada se o seu comportamento ocorre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispendo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos.

Para Pressman (2011), o processo de testes visa encontrar falhas no sistema para corrigi-las antes de distribuir o sistema, ou de atualizar o *software* com novos recursos, portanto o sistema deve ser projetado e implementado pensando na facilidade da realização dos testes. Por sua vez os testes devem ser feitos levando em consideração certas características para que nada seja deixado para trás.

Em conformidade com o autor, o conceito de testabilidade consiste em se medir o quão simples é o ato de testar um *software*. As características apresentadas seguir caracterizam um *software* a ser testável:

- a) **Operabilidade:** um sistema projetado e implementado tendo em mente a qualidade, terá poucas falhas quando os testes forem realizados.
- b) **Observabilidade:** quando é possível ver com clareza as entradas, saídas e variáveis do sistema fica mais fácil de detectar possíveis falhas.
- c) **Controlabilidade:** entradas geram saídas específicas, e para cada tipo de saída existirá um tipo de entrada específica. Se o engenheiro puder controlar essas entradas ficará mais fácil realizar os testes.
- d) **Decomponibilidade:** o sistema é construído a partir de módulos e pode ser testado em partes.

- e) **Simplicidade:** Quanto mais simples um sistema for, atingindo o objetivo, mais simples serão os testes.
- f) **Estabilidade:** Quanto menos alterações o software tiver, menos testes precisarão ser feitos.
- g) **Compreensibilidade:** Quanto mais informações estiverem disponíveis para o entendimento do software, mais eficazes serão os testes, isso inclui manuais organizados, detalhados e especificados.

A atividade de teste é composta por alguns elementos essenciais que auxiliam na formalização desta atividade, os quais são apresentados a seguir.

- a) **Caso de Teste:** descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado (CRAIG; JASKIEL, 2002).
- b) **Procedimento de Teste:** é uma descrição dos passos necessários para executar um caso (ou um grupo de casos) de teste (CRAIG; JASKIEL, 2002).
- c) **Critério de Teste:** serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto (ROCHA et al., 2001). Os critérios de teste são apresentados a seguir:
  - a. **Critério de cobertura dos testes:** permite a identificação de partes do programa que devem ser executadas para garantir a qualidade do *software* e indicar quando o mesmo foi suficientemente testado (RAPPS; WEYUKER, 1982). Ou seja, determinar o percentual de elementos necessários por um critério de teste que foram executados pelo conjunto de casos de teste.
  - b. **Critério de adequação de casos de teste:** Quando, a partir de um conjunto de casos de teste T qualquer, ele é utilizado para verificar se T satisfaz os requisitos de teste estabelecidos pelo critério. Ou seja, este critério avalia se os casos de teste definidos são suficientes ou não para



avaliação de um produto ou uma função (ROCHA et al., 2001).

- c. **Critério de geração de casos de teste:** quando o critério é utilizado para gerar um conjunto de casos de teste T adequado para um produto ou função, ou seja, este critério define as regras e diretrizes para geração dos casos de teste de um produto que esteja de acordo com o critério de adequação definido anteriormente (ROCHA et al., 2001).

### 3.17.1.1 Níveis de teste de software

O planejamento dos testes deve ocorrer em diferentes níveis e em paralelo ao desenvolvimento do software (Figura 15). Segundo Rocha et al. (2001) definimos que os principais níveis de teste de software são:

- a) **Teste de Unidade:** também conhecido como teste unitário. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.
- b) **Teste de Integração:** visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do *software* que foi estabelecida na fase de projeto.
- c) **Teste de Sistema:** avalia o *software* em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do *software*. Verifica se o produto satisfaz seus requisitos.
- d) **Teste de Aceitação:** são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações

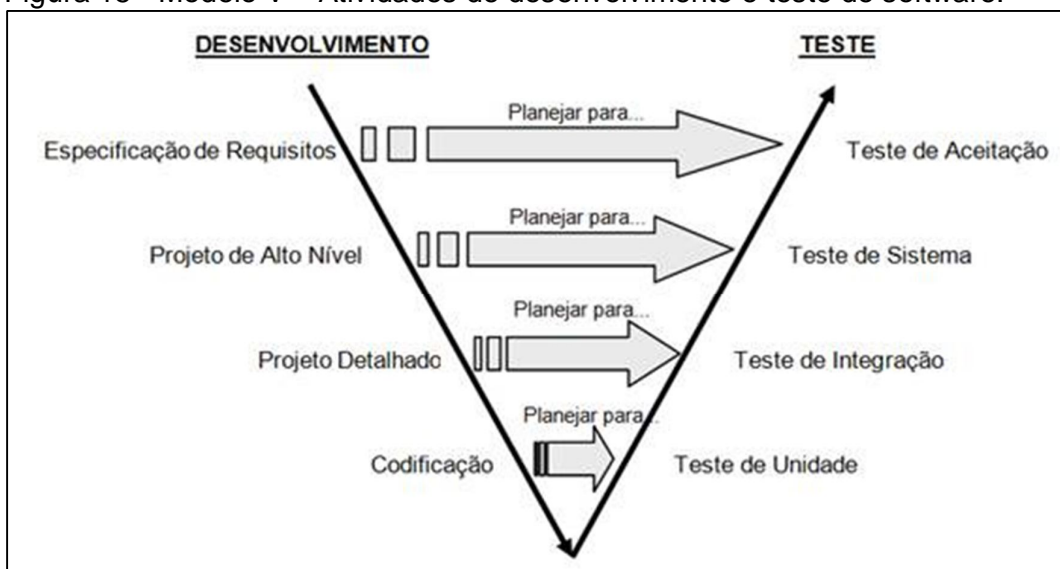
de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.

- e) **Teste de Regressão:** Teste de regressão não corresponde a um nível de teste, mas é uma estratégia importante para redução de “efeitos colaterais”. Consiste em se aplicar, a cada nova versão do *software* ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

Dessa forma, em conformidade com a Figura 15, o planejamento e projeto dos testes devem ocorrer de cima para baixo, ou seja:

- a) Inicialmente é planejado o teste de aceitação a partir do documento de requisitos;
- b) após isso é planejado o teste de sistema a partir do projeto de alto nível do software;
- c) em seguida ocorre o planejamento dos testes de integração a partir o projeto detalhado;
- d) e por fim, o planejamento dos testes a partir da codificação.

Figura 15 - Modelo V – Atividades de desenvolvimento e teste de software.



Fonte: (Craig e Jaskiel, 2002).

### *3.17.1.2 Técnicas de teste de software*

Atualmente existem muitas maneiras de se testar um *software*. Mesmo assim, existem as técnicas que sempre foram muito utilizadas em sistemas desenvolvidos sobre linguagens estruturadas que ainda hoje tem grande valia para os sistemas orientados a objeto. Apesar dos paradigmas de desenvolvimento serem diferentes, o objetivo principal destas técnicas continua a ser o mesmo: encontrar falhas no *software*.

De acordo com Claudio ([201-]), as técnicas de teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do *software*, e impõe-se a necessidade de se estabelecer uma estratégia de teste que contemple as vantagens e os aspectos complementares dessas técnicas.

#### 3.17.1.2.1 Teste de Carga

Engholm (2010) considera que os testes de carga devem conter os seguintes tópicos:

- i) Verificar a resposta do sistema com 200 usuários simultâneos;
- ii) Verificar a resposta do sistema com 500 usuários simultâneos;
- iii) Verificar a resposta do sistema com 1000 assinantes simultâneos.

#### 3.17.1.2.2 Teste de Funcional

O mesmo autor define também que o teste funcional apresenta os seguintes objetivos:

- a) Verificar se os visitantes conseguem ver as informações sobre as quais solicitaram mensagens;
- b) verificar se as mensagens do administrador são endereçadas aos visitantes quando ocorre problema durante cadastramento;
- c) verificar se a inserção de conteúdo automático está funcionando;
- d) verificar se a aprovação do editor provoca a inserção de conteúdo não-automático.

#### 4 TRABALHOS CORRELATOS

O uso da tecnologia na área fonoaudiologia vem crescendo com a evolução natural da tecnologia. A seguir podemos verificar alguns *softwares* que estão sendo utilizados nessa área.

Dentre os trabalhos podem-se citar: “Recursos de informática na terapia fonoaudiológica de crianças do espectro autístico”, o qual tem como objetivo verificar a interferência do uso de computadores e programas específicos na terapia fonoaudiológica de crianças autistas em seu perfil comunicativo e desempenho sócio-cognitivo. Como resultado as seguintes características foram descritas pelas terapeutas para as situações com o uso do computador: as crianças se apresentaram mais atentas, com mais iniciativas de comunicação, mais contato ocular, com maior nível de interação, bem como, mais verbalizações. Além de possíveis progressos qualitativos e quantitativos, esses progressos foram observados num período de tempo menor do que o usualmente utilizado para esse tipo de comparação, isso o torna um resultado promissor (FERNANDES; SANTOS; AMATO; MOLINI, 2010).

Outro trabalho que mostra o estudo da tecnologia na fonoaudiologia é o trabalho intitulado “Utilização de Softwares por Fonoaudiólogos no Rio Grande do Sul”, que teve como objetivo averiguar a utilização de *softwares* na prática dos fonoaudiólogos no Rio Grande do Sul. A partir deste trabalho verificou-se que 62,5% dos profissionais utilizavam *softwares*, sendo a maioria localizada na região metropolitana e Porto Alegre, também foram levantadas as especialidades mais citadas na utilização de *softwares*, as quais se destacam: linguagem, motricidade orofacial, disfagia e audiologia, com principal finalidade terapêutica, havendo uma boa aceitação pelos pacientes (SANTOS; FERNANDES; FERREIRA; TRINDADE; VIDOR, 2015).

## 5 METODOLOGIA

Uma pesquisa exploratória ajuda o pesquisador, a saber, quais das várias opções se aplicam ao problema de pesquisa. As prioridades poderão ser estabelecidas porque uma particular hipótese explicativa surgida durante a pesquisa exploratória parecerá mais promissora do que outras (MATTAR, 2012).

O trabalho foi desenvolvido em três etapas, inicialmente foi realizada uma pesquisa com fonoaudiólogos, para obter informações sobre o processo de terapia intensiva de fonoaudiologia e assim definir os objetivos do aplicativo e o público-alvo, em seguida foram estudados materiais teóricos de diversos assuntos, os quais estão relacionados com o escopo desta pesquisa, envolvendo áreas como: Informática aplicada a fonoaudiologia, engenharia de *software*, modelagem de banco de dados, Interação Humano-Computador, linguagem de programação Java, entre outros.

Esta etapa também abrangeu a escolha da plataforma coerente com o projeto a ser executado. Para cada sistema operacional, seja no *smartphone* ou *tablet*, existe uma plataforma específica para o desenvolvimento, e é a definição desta plataforma que determina o custo e prazo do projeto de criação do aplicativo, além de sua manutenção e evolução. A escolha da tecnologia é fundamental, porque ela define como o aplicativo será desenvolvido e como os usuários irão interagir com ele. Optar por uma tecnologia errada pode ser um erro caro, resultando em inacessibilidade ou mau desempenho.

As ferramentas certas simplificam todo o processo; por isso, não se pode esquecer no processo de escolha de uma plataforma, por exemplo, de identificar se ela tem a funcionalidade de Gerenciamento de Aplicativos Móveis (MAM), ou as características de Gerenciamento de Dispositivos Móveis (MDM). Essas características tratam de questões como controle de acesso centralizado e gerenciamento de versão do aplicativo.

Ainda foi preciso certificar-se se a plataforma inclui outras características importantes, como: recursos abrangentes de testes, codificação amigável móvel, protocolos de transferência de dados, armazenamento seguro *on-device*, capacidades de identificação de rede, entre outros.

Na segunda etapa foi feita a modelagem e o desenvolvimento do aplicativo mobile para a plataforma Android, bem como, sua disponibilização no Google Play. O **Google Play**, ou **Google Play Store** é uma loja virtual da **Google**, nela é possível fazer o *download* de músicas, filmes, livros e, é claro, aplicativos.

A terceira e última etapa, constituiu em analisar *feedbacks* dos profissionais de fonoaudiologia que utilizaram o aplicativo em relação à interface e funcionalidades. Esta interação entre desenvolver e profissionais da área foi de extrema importância para estabelecer propostas de melhorias, bem como, trabalhos futuros.

## 5.1 FERRAMENTAS UTILIZADAS

A seguir, serão apresentadas as ferramentas que foram utilizadas durante o desenvolvimento do projeto considerando todas as suas etapas.

### 5.1.1 Hardware

Seguindo a estrutura estabelecida, a aplicação foi desenvolvida utilizando como ferramenta principal um notebook da marca DELL inspiron 14r, processador i5, 3337U, CPU@ 1.80GHz, memória RAM de 6 GB, sistema operacional Windows 10 - 64 bits. Como ferramenta de teste, foi utilizado um *smartphone* Motorola Moto G2, com sistema operacional Android 6.0.. Este aparelho auxiliar foi utilizado no decorrer do projeto com o propósito de testar desempenho e interatividade em função do aplicativo desenvolvido.

A escolha de ambos os aparelhos se dá pelo fato de que são propriedades do autor, além de serem suficientes para suprir todas as necessidades que surgiram no decorrer do desenvolvimento.

### 5.1.2 Softwares

Em nível de *software*, foram utilizadas ferramentas de apoio ao desenvolvedor, programas de modelagem UML e gerenciadores de banco de dados.

- a) **Astah Community**: esta ferramenta oferece ao programador uma forma simples e rápida de desenvolver diagramas de modelagem UML, além de apresentar uma interface bem intuitiva (Figura 16), ao mesmo tempo em que facilita a criação de diagramas mais complexos, e pode ser adquirida no *site*: <[https://members.change-vision.com/files/astah\\_community](https://members.change-vision.com/files/astah_community)>. A escolha pela ferramenta se dá pelo da mesma ser disponibilizada gratuitamente pela própria empresa Astah, além de ter sido utilizada em sala de aula durante o aprendizado sobre a modelagem de diagramas no decorrer da disciplina de Engenharia de *Software II*;
- b) **SQLite**: foi utilizado como banco de dados interno, por se tratar de uma poderosa ferramenta, além de ser totalmente nativo do Android, sem necessitar de bibliotecas de terceiros;
- c) **Java Development Kit (JDK)**: O Java Development Kit (JDK) contém um ambiente integrado de desenvolvimento para a criação de aplicações, mini-aplicativos e componentes utilizando a tecnologia Java;
- d) **Android SDK**: SDK são as iniciais para Software Development Kit e é a linguagem de programação do **Android**. A partir dela podem-se emular *softwares* desenvolvidos para outras plataformas no Android;
- e) **Android Studio versão 2.2.2**: O Android Studio é a IDE oficial da Google para desenvolvimento para Android.

## 5.2 LINGUAGENS DE PROGRAMAÇÃO E MODELAGEM

As linguagens de programação utilizadas neste projeto foram implementadas de forma a complementarem uma a outra pelo bem da aplicação.

- a) **A UML:** o padrão UML foi a primeira linguagem a ser utilizada e desempenhou o papel de modelar a estrutura do sistema, sobre a qual o mesmo foi desenvolvido. Assim, trata-se de uma escolha fundamental para as etapas seguintes que se apoiam em como o sistema funcionaria. A linguagem de modelagem foi aplicada com o *software* Astah Community para criar os Diagramas de Casos de Uso, de Classes e de Atividades;
- b) **A linguagem Java:** foi utilizada com o propósito de criar toda a estrutura e lógica de programação do aplicativo que foi desenvolvido. O Java foi escolhido por ser uma linguagem moderna e crescente no mercado. Além de oferecer todos os recursos que serão necessários a nível de programação, Java também possui excelente suporte e uma vasta comunidade de desenvolvedores disponíveis para consulta.

## 5.3 MODELAGEM DO APLICATIVO

Na área de desenvolvimento de *softwares*, muitas vezes os programas acabam sendo considerados inadequados para a empresa/usuário, e não atendem às necessidades dos usuários, devido à produtividade e facilidade oferecidas pelas linguagens de programação visuais, e quanto mais complexo for o sistema, maior será a probabilidade de ocorrência de erros, no caso de ter sido feito sem nenhum tipo de modelagem.

Para a modelagem dos diagramas utilizados no projeto foi utilizada a ferramenta Astah Community, com o propósito de expressar a estrutura do aplicativo, tal como de elaborar o planejamento da parte prática do trabalho.

Esses diagramas foram posteriormente utilizados com fins de consulta no decorrer da implementação da ideia inicial.



Com o propósito de cobrir todas as áreas pertinentes deste projeto foram utilizados os diagramas de Atividades, Casos de Uso e de Classes.

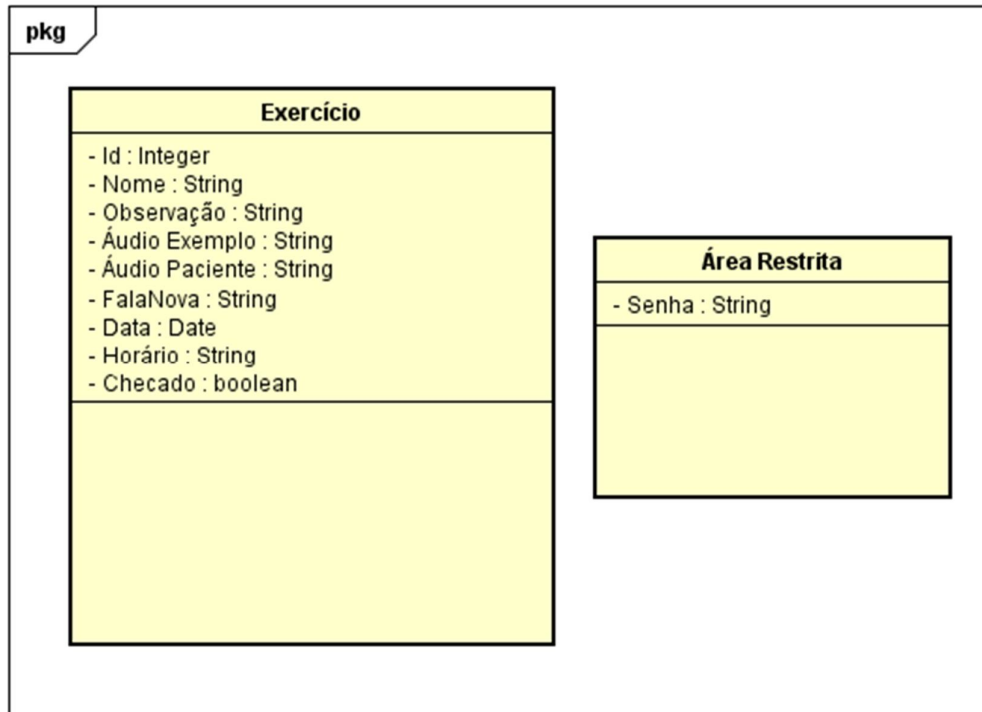
### 5.3.1 Diagrama de classes

O Diagrama de Classes do aplicativo possui duas classes, a classe “Exercício” e a classe “Área Restrita”, como pode ser observado na Figura 16. A classe “Exercício” tem como objetivo armazenar as seguintes informações:

- a) **Id:** Id é o número único do exercício, cada exercício possui apenas um único id e esse id é único no aplicativo inteiro, ele é gerado automaticamente através do banco de dados e é utilizado para referenciar os exercícios no banco de dados.
- b) **Nome:** Nome/descrição do exercício a ser realizado.
- c) **Observação:** Esse campo é utilizado pelo usuário para descrever para o fonoaudiólogo se encontrou alguma dificuldade ou facilidade com o exercício realizado.
- d) **Áudio Exemplo:** O áudio exemplo é gravado pelo fonoaudiólogo para orientar o paciente ao realizar o exercício proposto.
- e) **Áudio Paciente:** O áudio paciente é gravado pelo paciente para que o fonoaudiólogo tenha o registro de como foi realizado o exercício.
- f) **FalaNova:** Armazena se o paciente utilizou a fala nova sempre, quase sempre ou se não utilizou.
- g) **Data:** Data em que o exercício será disparado.
- h) **Horário:** Horário em que o exercício será disparado.
- i) **Checado:** Armazena o *status* concluído toda vez que um exercício é realizado.

Essas informações são armazenadas para auxiliar o profissional de fonoaudiologia no controle da evolução do tratamento de seus pacientes, ao mesmo tempo em que são utilizadas para traçar estatísticas de evolução, podendo também ser enviadas por e-mail. A Figura 16 ilustra o Diagrama de Classes do Aplicativo.

Figura 16 – Diagrama de Classes do aplicativo.

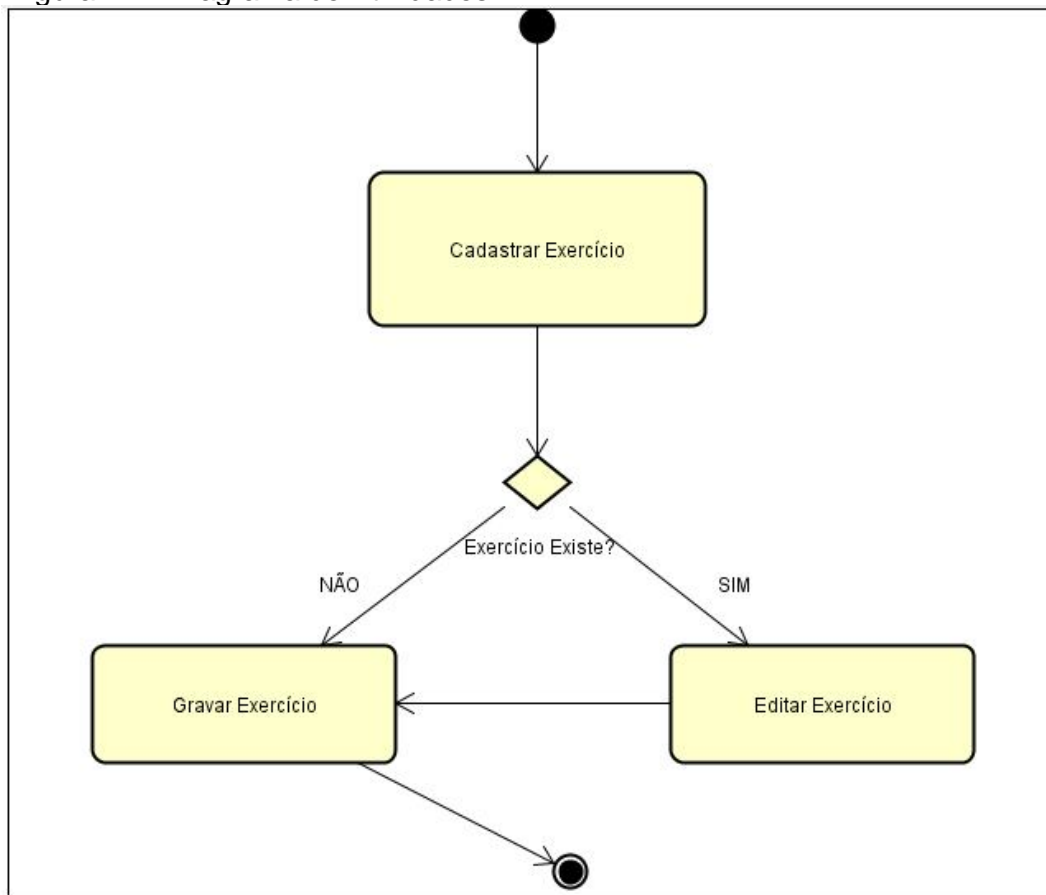


Fonte: Elaborada pelo autor.

### 5.3.2 Diagrama de atividades

O diagrama de Atividades representado na Figura 17 demonstra como o exercício é gravado no banco de dados. Em um primeiro momento é feito o cadastro com todas as informações apresentadas na Figura 16, logo após é feita uma validação com o Banco de dados SQLite que retorna se o cadastro realizado já existe ou não no, caso exista, ele é editado e posteriormente gravado, caso contrário, ele é gravado diretamente.

Figura 17 – Diagrama de Atividades.

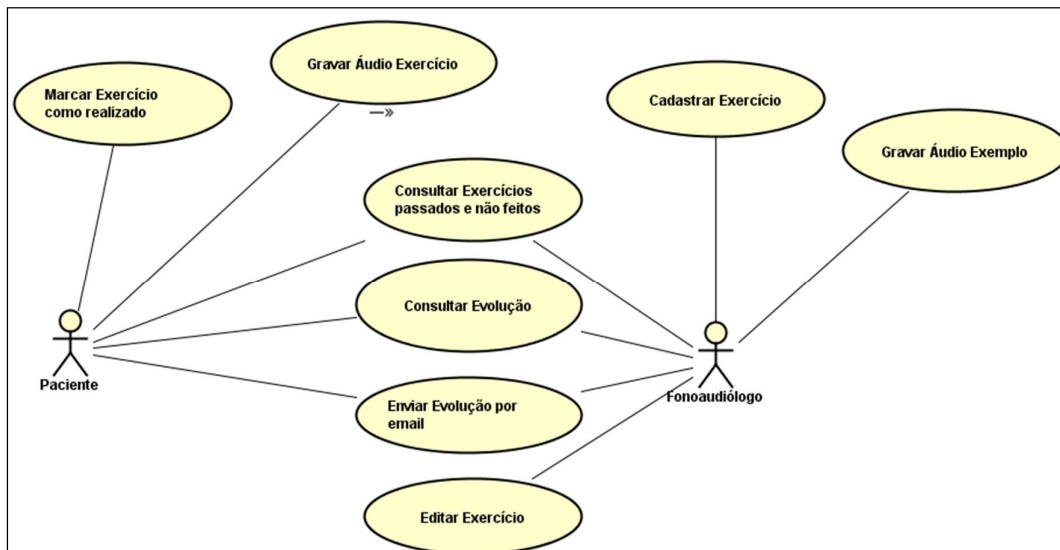


Fonte: Elaborada pelo autor.

### 5.3.3 Diagrama de casos de uso

O Diagrama de Casos de Uso do aplicativo (Figura 18) tem como atores: paciente - o qual pode consultar exercícios passados e não feitos, marcar exercícios como realizados, gravar áudio na realização do exercício, além de poder consultar sua evolução, ou seja, verificar os exercícios já realizados. O outro ator é o fonoaudiólogo, que tem como papel consultar exercícios passados e não feitos pelos pacientes, consultar a evolução dos pacientes, cadastrar exercício, além de também poder editá-lo, gravar áudio exemplo para realização do exercício, tanto o fonoaudiólogo quanto o paciente podem enviar a evolução por e-mail.

Figura 18 – Diagrama de Casos de Uso.



Fonte: Elaborada pelo autor.

## 5.4 DESENVOLVIMENTO DO APLICATIVO

A seguir será apresentada a plataforma escolhida para receber o aplicativo.

### 5.4.1 Escolha da plataforma

O aplicativo recebeu o nome de “Fala Nova, Treino Fácil” e foi projetado para funcionar em *smartphones* que utilizem a plataforma Android, superior a versão 4.0. Foi escolhida a plataforma Android para o desenvolvimento desse aplicativo por ser a plataforma mobile com o maior número de usuários ativos, seguida do iOS e Windows Phone, como pode ser observado na pesquisa realizada em 2014 e retratada anteriormente na Figura 2.

## 6 RESULTADOS E DISCUSSÕES

A seguir serão apresentados os resultados e discussões desta pesquisa.

### 6.1 FUNCIONAMENTO

Ao abrir o aplicativo “Fala Nova Treino Fácil”, o usuário se depara com uma tela inicial (Figura 19) que possui quatro botões:

- a) **Listar Exercícios Passados:** onde é possível listar todos os exercícios do dia que o usuário não marcou como concluídos;
- b) **Evolução:** no qual mostra como está ocorrendo a evolução do paciente;
- c) **Área da Fono:** na qual se encontra uma área dedicada a cadastros, edição de exercícios.
- d) **Sair:** sai do aplicativo.

Figura 19 – Tela principal do aplicativo.



Fonte: Elaborada pelo autor.

Como pode ser observado na Figura 19, o fonoaudiólogo tem sua área restrita, onde existe a possibilidade de cadastrar, alterar e excluir todos os exercícios, além de uma área de evolução, através da qual pode acompanhar como está evoluindo o tratamento de seu paciente.

Ao clicar pela primeira vez no botão “**Área da Fono**”, é aberta uma tela solicitando para que o profissional de fonoaudiologia cadastre uma senha para ter acesso a essa área, conforme pode ser visualizado na Figura 20.

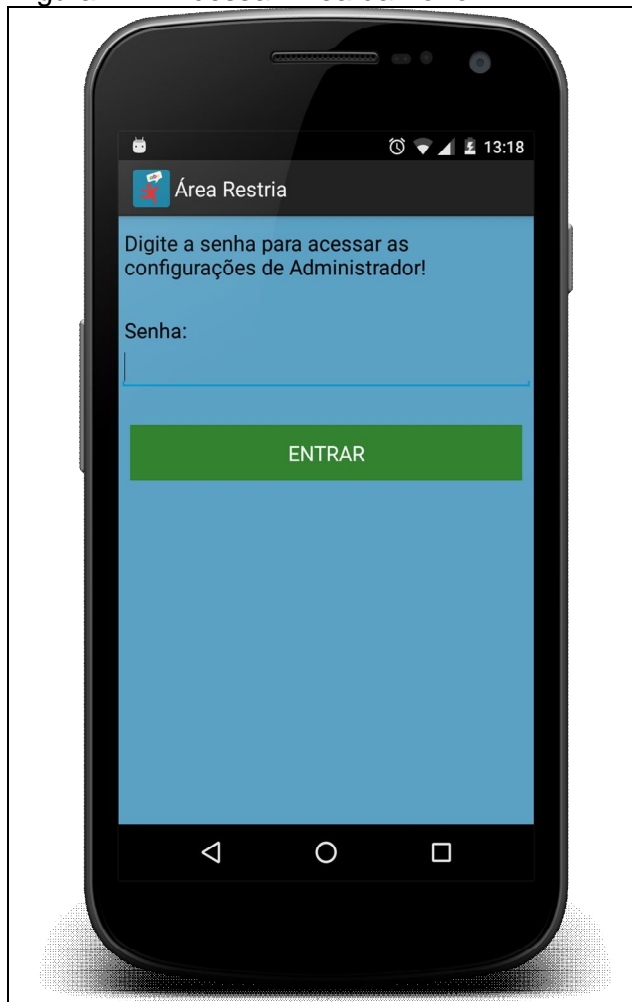
Figura 20 – Cadastrar Senha - Área da Fono.



Fonte: Elaborada pelo autor.

Uma vez cadastrada a senha, nas próximas utilizações a mesma será solicitada, conforme mostra a Figura 21.

Figura 21 – Acessar Área da Fono.



Fonte: Elaborada pelo autor.

Ao digitar a senha correta é aberta uma nova tela, com botões para “cadastrar”, “listar” e “adicionar” exercícios de fala nova, e também, verificar a evolução do paciente. A Figura 22 ilustra este contexto.



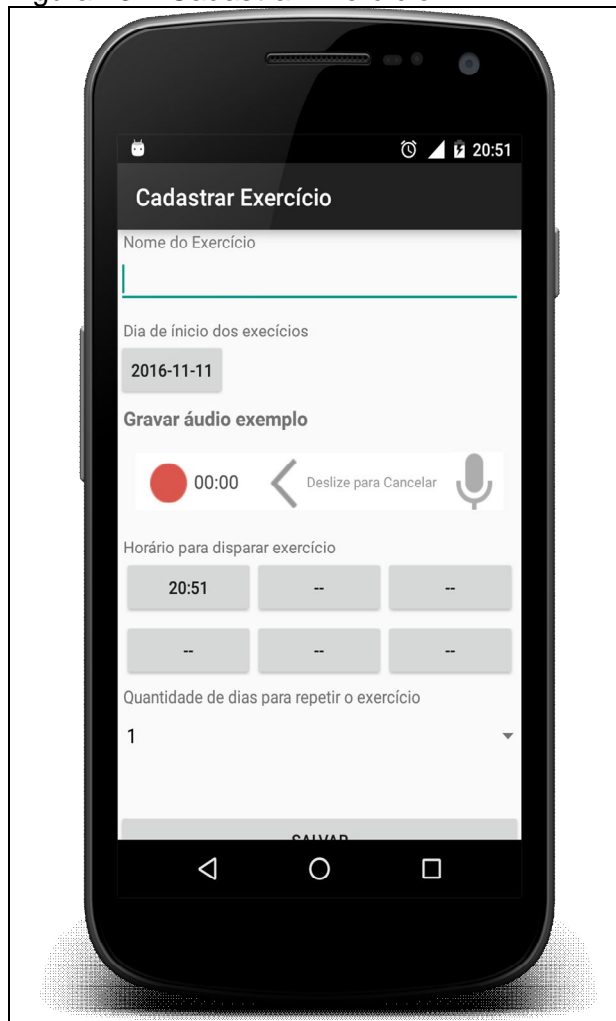
Figura 22 – Área restrita da Fonoaudióloga.



Fonte: Elaborada pelo autor.

Ao clicar no botão “Cadastrar Exercício” (Figura 22) é aberta uma tela no qual o profissional de fonoaudiologia deve inserir o nome do exercício, a data de início, os horários para disparar o exercício, bem como, por quantos dias irá se repetir, além da possibilidade de gravar um áudio exemplo de como deve ser realizado o exercício, conforme pode ser observado na Figura 23.

Figura 23 – Cadastrar Exercício.



Fonte: Elaborada pelo autor.

Ao clicar no botão “**Cadastrar Fala Nova**” (Figura 22) é aberta uma tela onde o profissional de fonoaudiologia deve escolher a data de início, o horário inicial e final para disparar os lembretes, além de definir qual será o intervalo em minutos (de um lembrete para o outro), bem como, por quantos dias irá se repetir. A única diferença dessa tela com a tela “Cadastrar Exercício”, é que nessa não é contabilizado exercício para o paciente realizar, ela serve apenas para lembrar o paciente que ele deve utilizar a fala nova, como pode ser observado na Figura 24.

Figura 24 – Cadastrar Fala Nova.



Fonte: Elaborada pelo autor.

O botão “**Evolução**” apresentado na Figura 22 pode ser acessado tanto pelo usuário quanto pelo profissional de fonoaudiologia, e ao ser clicado, é aberta uma tela onde pode ser visualizada a evolução durante o tratamento. Esta tela exibe a quantidade de exercícios que foram propostos, quantidade realizada, e a quantidade não realizada. Com base nestes dados é gerado em forma de percentual um “Aproveitamento”, referente à quantidade de exercícios propostos e realizados. Ainda nessa tela é possível observar as respostas dadas pelo paciente em cada exercício, onde é questionado se o mesmo realizou o exercício “sempre”, “quase sempre” ou “não usou”. A Figura 25 apresenta este contexto.

Figura 25 – Evolução do tratamento.



Fonte: Elaborada pelo autor.

Existe a possibilidade de enviar os dados da evolução (Figura 25) por *e-mail*. Para isso basta clicar no botão “**Enviar por e-mail**”, e será aberta uma nova tela solicitando o *e-mail* do profissional de fonoaudiologia, o nome do paciente e uma observação que é opcional, conforme pode ser visto na Figura 26.

Figura 26 – Enviando e-mail.



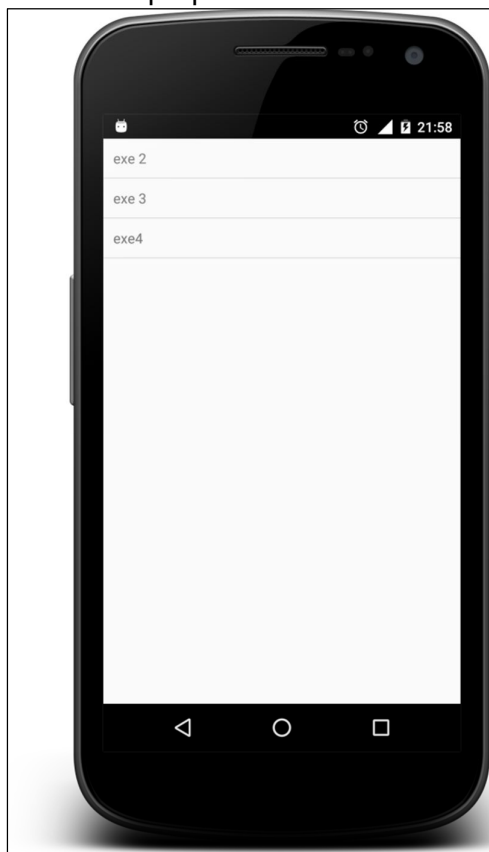
Fonte: Elaborada pelo autor.

As telas abertas ao clicar nos botões “Listar Todos Exercícios Feitos” e “Listar Todos Exercícios Propostos” são praticamente iguais, a diferença entre elas são apenas os resultados mostrados.

- a) **Listar Todos Exercícios Feitos:** são mostrados apenas os exercícios que o paciente marcou como Feito;
- b) **Listar Todos Exercícios Propostos:** são mostrados todos os exercícios propostos pelo fonoaudiólogo.

Esta situação pode ser observada nas Figuras 27 e 28.

Figura 27 – Mostrar todos os exercícios propostos.



Fonte: Elaborada pelo autor.

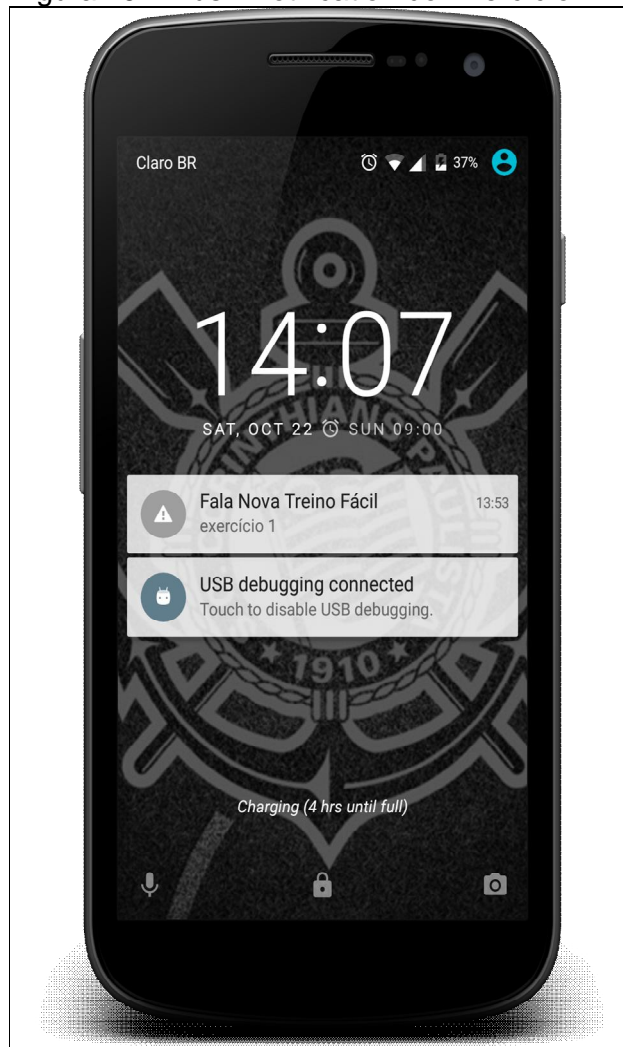
Figura 28 – Mostrar todos os exercícios feitos.



Fonte: Elaborada pelo autor.

No horário programado pelo fonoaudiólogo para disparar o exercício, é enviado um *push notification* para o *smartphone* do paciente, através do qual é possível verificar o exercício a ser realizado. Para acessar o exercício basta o paciente clicar no ***push***. Esta situação pode ser visualizada na Figura 29.

Figura 29 – Push Notification do Exercício.



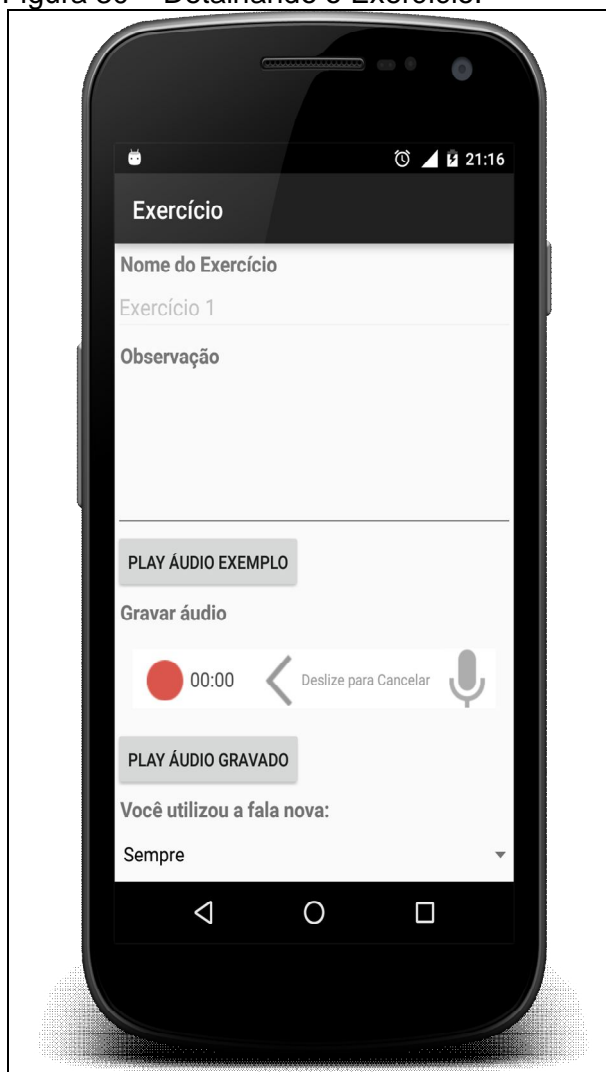
Fonte: Elaborada pelo autor.

Assim que o usuário acessa o exercício é possível verificar o **nome do exercício**, porém com o campo bloqueado, para que o paciente não consiga alterá-lo. Também são apresentados os seguintes itens:

- a) Um campo de **observação**: onde o paciente pode relatar algo durante a realização daquele exercício, como por exemplo, se foi complicado ou fácil de realizar o exercício.
- b) Um botão com o nome "**PLAY ÁUDIO EXEMPLO**". Ao clicar neste botão o usuário pode ouvir o áudio de exemplo (modelo) que o fonoaudiólogo cadastrou no momento da criação do exercício.
- c) Um botão para gravar o áudio do paciente realizando o exercício.
- d) Um botão para que o paciente escute seu próprio áudio.

Abaixo do botão “**Play Áudio**” existe a seguinte pergunta: “Você utilizou a fala nova:”. O paciente deve escolher uma dentre as três opções: “Quase Sempre”, “Sempre”, “Não Usei” (Figura 30). Também se apresenta o campo data e o campo horário, porém ambos também bloqueados, e por fim, um campo do tipo *checkbox* para que o paciente marque como exercício concluído. Realizado os passos anteriores o paciente tem a opção de clicar no botão **Salvar**, conforme ilustra a Figura 31.

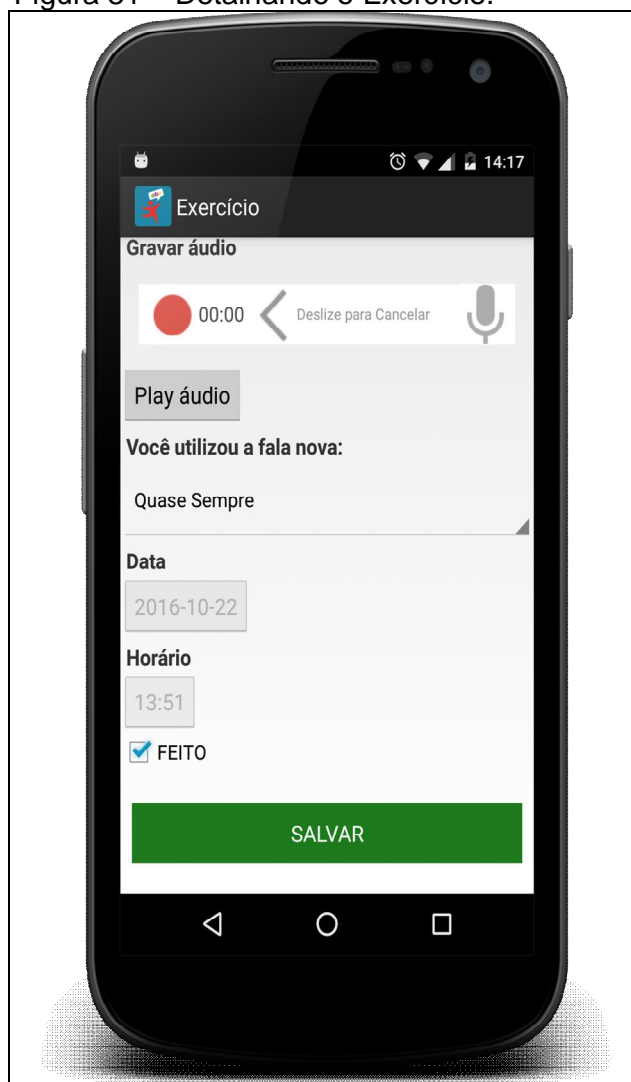
Figura 30 – Detalhando o Exercício.



Fonte: Elaborada pelo autor.



Figura 31 – Detalhando o Exercício.



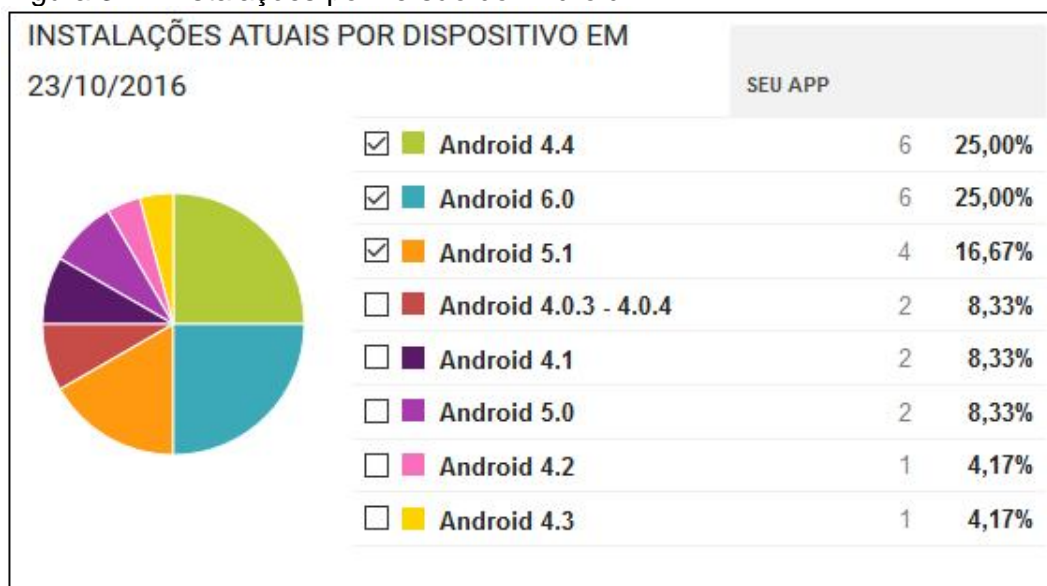
Fonte: Elaborada pelo autor.

## 6.2 BUSCA E DOWNLOAD DO APLICATIVO

O aplicativo Fala Nova Treino Fácil foi publicado no Google Play em 29 de Junho de 2016 de forma gratuita. Em 25 de Outubro de 2016 o aplicativo contava com 97 *downloads* feitos.

Podemos visualizar na Figura 32 o gráfico das instalações por versão do Android.

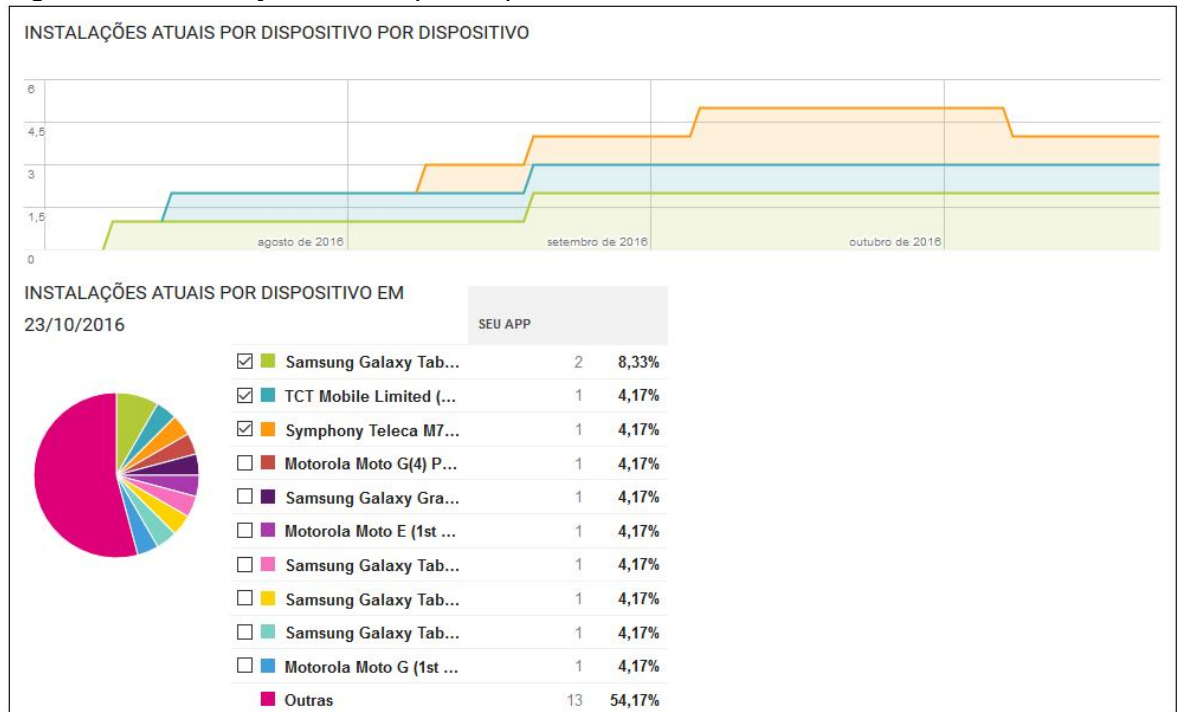
Figura 32 – Instalações por versão do Android.



Fonte: Developer Console – Google Play (2016).

A Figura 33, a instalação do “Fala Nova Treino Fácil” por modelo de dispositivo, bem como, o acompanhamento da evolução de instalação através do mapa de linhas.

Figura 33 – Instalações atuais por dispositivo x modelo.



Fonte: Developer Console – Google Play (2016).

## 7 CONSIDERAÇÕES FINAIS

O Hospital de Reabilitação de Anomalias Craniofaciais - HRAC/USP, Serviço de Prótese de Palato e a Universidade do Sagrado Coração - USC deram um passo importante no desenvolvimento de novas tecnologias no tratamento de fala de pacientes com fissura labiopalatina. O aplicativo já está em uso no HRAC/USP.

A ideia deste projeto surgiu a partir de uma proposta de parceria em um projeto de Doutorado de duas alunas do HRAC/USP com a USC, por intermédio dos professores do curso – prof. Elvio Gilberto da Silva e prof. Patrick Pedreira Silva. A professora Dra. Jeniffer C. R. Dutka (HRAC/USP) procurou os professores acima citados solicitando a possibilidade da parceria, bem como, o desenvolvimento do aplicativo. A partir disso, várias reuniões foram feitas com as doutorandas e os professores supracitados visando levantar todos os requisitos e funcionalidades necessárias para o desenvolvimento do aplicativo.

O aplicativo cumpriu com o proposto inicialmente, conforme demonstrado nos resultados. Além do profissional de fonoaudiologia obter o controle de quais exercícios foram realizados pelos seus pacientes, também é possível ouvir o áudio de seu paciente realizando o exercício, possibilitando assim, que verifique de forma mais eficaz a deficiência que o paciente venha a apresentar no momento da execução do exercício.

O prazo estimado para o desenvolvimento do aplicativo foi de oito meses, esse resultado foi encurtado para um prazo de seis meses de desenvolvimento, assim restou um tempo para realizar melhorias na interface, e implementação de novas funcionalidades sugeridas pelos profissionais de fonoaudiologia durante a fase de testes do aplicativo.

A informática, utilizada em terapia fonoaudiológica, mostra-se um recurso viável, criativo e enriquecedor. Além de oferecer uma nova gama de estratégias para o fonoaudiólogo, a informática também propicia diferentes possibilidades para um trabalho clínico, ao situar o terapeuta como mediador deste processo.

Para o paciente, a informática contribui de forma significativa ao proporcionar uma nova perspectiva para que a aprendizagem aconteça. Os

recursos multimídia, com as várias modalidades de informação que as Tecnologias da Informação e Comunicação incorporam, ampliam o repertório simbólico do paciente, auxiliando a prática e o aprimoramento das linguagens oral e auditiva. Estimula, ainda, a autonomia do paciente, ao colocá-lo como sujeito ativo dentro do processo terapêutico, ao mesmo tempo em que desperta no paciente a vontade da descoberta em um ambiente que favorece a livre exploração.

## 8 TRABALHOS FUTUROS

Sugere-se como trabalhos futuros, a possibilidade do aplicativo trabalhar de forma *online*, no qual o paciente possa realizar o exercício, e por meio de um Webservice gravá-lo em um banco de dados externo (MySQL), permitindo assim que o profissional de fonoaudiologia consiga visualizar em tempo real a evolução de seu paciente durante o decorrer do tratamento.

Também se sugere o desenvolvimento do aplicativo para versão iOS (iPhone - Apple), visto que o número de usuários desta plataforma vem crescendo exponencialmente nos últimos anos, ficando atrás apenas do Android.

## 9 SUBMISSÕES E PUBLICAÇÕES

O aplicativo foi submetido ao XXIV Congresso Brasileiro de Fonoaudiologia sob o título “**Projeto de inovação no tratamento de fala de pacientes com fissura labiopalatina: aplicativo móvel**”, e aprovado para apresentação em forma de pôster.

Existe a pretensão de submetê-lo em forma de artigo para a Revista Ciência & Desenvolvimento - Revista Eletrônica da FAINOR. A Revista Ciência & Desenvolvimento é um espaço de divulgação de práticas pedagógicas desenvolvidas no âmbito acadêmico. A revista fomenta a produção científica e a disseminação de conhecimento multidisciplinar, objetivando a troca de informações, a reflexão e o debate, provendo assim o desenvolvimento social e científico.

Ciência & Desenvolvimento é um periódico interdisciplinar da Faculdade Independente do Nordeste – Fainor, e está avaliado na base de dados Qualis da CAPES, referência 2013/2014, como: B2 - Ensino; B5 - Ciências Ambientais, Enfermagem. C – Educação.

## 10 REFERÊNCIAS

AQUINO, J. F. S. **Plataformas de desenvolvimento para dispositivos móveis**. 2007. 14 f. Monografia (Programa de Pós Graduação em Informática) - Pontifícia Universidade Católica do Rio De Janeiro, Departamento de Informática, 2007. Disponível em: <<http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/07/Android-Juliana-Mono.pdf>>. Acesso em: 5 nov. 2016.

BAPTISTA, E. V. P. **Malformações congênitas associadas à fissura labial e/ou palatal em pacientes atendidos em um serviço de referência para tratamento de defeitos da face: Um estudo de série de casos**. 2007. 83 f. Dissertação (Mestrado em Ciências Biológicas) – Instituto Materno Infantil Professor Fernando Figueira, 2007.

BASTOS, T. R. **Aplicativos para dispositivos móveis e seu uso em bibliotecas**: uma visão das experiências em âmbito internacional [Trabalho de Conclusão de Curso- Graduação]. Goiânia (GO): Faculdade de Informação e Comunicação, Universidade Federal de Goiás; 2014. Disponível em: <<https://repositorio.bc.ufg.br/handle/ri/10806?mode=full>>. Acesso em: 18 nov. 2015.

BAYLIS, A. L.; MUNSON, B.; MOLLER K. T. Perceptions of audible nasal emission in speakers with cleft palate: a comparative study of listener judgments. *Cleft Palate Craniofac J* 2011; 48(4):399-411.

BISPO, N. H. M.; WHITAKER, M. E.; AFERRI, H. C.; NEVES, J. D. A; DUTKA, J. C. R.; PEGORARO-KROOK, M. I. Speech therapy for compensatory articulations and velopharyngeal function: a case report. *J Appl Oral Sci*. 2011;19(6):679-84.

BUDEL, C.; MOLOSSI, R. **Arquiteturas para Aplicações Móveis**, Paraná, Novembro 2011. Disponível em: <<http://zip.net/bltkJg>>. Acesso em: 28 abr. 2016.

CAPOVILLA, F. C. e colaboradores. **Desenvolvimento linguístico na criança dos dois anos aos seis anos**: Ciência cognitiva: teoria, pesquisa e aplicação, v. 1, n.1, 1997.

CLAUDIO, A. Introdução a teste de software. **DEV MEDIA**, Rio de Janeiro, 201-. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>>. Acesso em: 26 maio. 2016.

COSTA, C. E. R. **A Influência das redes sociais na comunicação organizacional**. 2011. 1 f. Monografia (Tecnologia em Processamento de Dados) – Faculdade de Tecnologia de São Paulo – FATEC/SP, São Paulo, 2011.



CRAIG, R. D., JASKIEL, S. P. **Systematic Software Testing**. Boston: Artech House Publishers, 2002.

DANTAS, M.A.L. **Avaliação de desvios fonéticos e/ou fonológicos em crianças**: contribuições para a criação de um software aplicativo. Dissertação (Mestrado em Ciências da Linguagem) - Universidade Católica de Pernambuco – UNICAP, 2006.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 9. reimp. Rio de Janeiro: Elsevier, 2003.

ENGHOLM, H. J. **Engenharia de Software na Prática**. São Paulo: Novatec, 2010.

FERRARI, F. R. **Banco de Dados I**. 2010. 66 p. Apostila. Disponível em: <<https://pt.scribd.com/doc/37452635/Banco-de-Dados-I-Apostila>>. Acesso em: 5 nov. 2016.

FERNANDES, F. D.; SANTOS, T. H.; AMATO, C. A.; MOLINI, D. R. **Recursos de informática na terapia fonoaudiológica de crianças do espectro artístico**, São Paulo, 2010, Disponível em: <[www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0104-56872010000400009](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-56872010000400009)>. Acesso em: 12 maio 2016.

FERNANDES, R. R.; L. M.; SILVA, A. C. M.; ARAÚJO, M. O. O uso da tecnologia da informação em dispositivos móveis. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 7., 2010. Disponível em: <[http://www.aedb.br/seget/arquivos/artigos10/22\\_Seget\\_2010\\_TI.pdf](http://www.aedb.br/seget/arquivos/artigos10/22_Seget_2010_TI.pdf)>. Acesso em: 5 set. 2016.

FERNANDES, G. G.; **Interface Humano Computador**: prática pedagógica para ambientes virtuais, Teresina, 2009, Disponível em: <[http://cead.ufpi.br/conteudo/material\\_online/disciplinas/video/livro\\_gildasio.pdf](http://cead.ufpi.br/conteudo/material_online/disciplinas/video/livro_gildasio.pdf)>. Acesso em: 6 jan. 2015.

FORTE, L. K. **A tecnologia informática e suas aplicações clínicas**. In: FOZ, F. B., PICCARONE, M. L., BURSZTYN, C. S. A tecnologia informática na fonoaudiologia. São Paulo: Plexus Editora, 1998. 222 p.

FUKUSHIRO, A. P.; GENARO, K. F.; SUGUIMOTO, M. F. F. C. P. **Fissuras Labiopalatinas**: Uma abordagem Interdisciplinar. São Paulo: Santos, 2007.

FURGERI, S. **Java 2 ensino didático**: desenvolvendo e implementando aplicações. In: Silva E.G. Uma ferramenta de apoio à avaliação de disciplinas em cursos de graduação. Bauru: EDUSC, 2006.

GEREMIA, J. **Tutorial de Introdução a Banco de Dados**. 2010. Tutorial de Introdução a Banco de Dados, Escola de Engenharia, Universidade Federal Fluminense, Niterói – Rio de Janeiro, 2010. Disponível em:

<[http://www.telecom.uff.br/pet/petws/downloads/tutoriais/db/Tut\\_DB.pdf](http://www.telecom.uff.br/pet/petws/downloads/tutoriais/db/Tut_DB.pdf)>. Acesso em: 5 nov. 2016.

GOMES, P. J. **A Ação da fonoaudiologia junto ao portador de fissura lábio palatina**. Disponível em <<http://www.webartigos.com/artigos/a-acao-da-fonoaudiologia-junto-ao-portador-de-fissura-labio-palatina/28285/>>. Acesso em: 30 out.. 2016.

GONÇALVES, E. C. **SQLite, Muito Prazer!**. Disponível em <<http://www.devmedia.com.br/post-7100-SQLite-Muito-Prazer.html>>. Acesso em: 7 maio. 2016.

HAMANN, R. **iOS, Android e Windows Phone: números dos gigantes comparados**. Disponível em < <http://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm>>. Acesso em: 7 maio. 2016.

HEUSER, C. A. **Projeto de Banco de Dados**. 6. ed. Porto Alegre: Bookman, 2009.

HUTH, G. **Um modelo para o gerenciamento de bancos de dados SQL através de Stored Procedures**. 2002. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2002.

KESKE, K. de L., BERNARDIR, G. **Fonoware: Software para auxílio na avaliação e terapia da fonoaudiologia**. Disponível em: <[www.inf.unifra.br/sis-info\\_tfg2002.php](http://www.inf.unifra.br/sis-info_tfg2002.php)>. Acesso em: 7 maio. 2016.

KUMMER, A. W. Speech evaluation for patients with cleft palate. Clin Plast Surg. 2014;41:241-51.

LAET, L. G. L.; Utilização da plataforma Android no desenvolvimento de um aplicativo para o setor sucroalcooleiro. **Coordenação do Curso de Sistemas de Informação Universidade Estadual do Mato Grosso do Sul**, Mato Grosso do Sul, 2010, Disponível em: <[http://www.uems.br/portal/biblioteca/repositorio/2012-06-26\\_18-08-13.pdf](http://www.uems.br/portal/biblioteca/repositorio/2012-06-26_18-08-13.pdf)>. Acesso em: 3 jan. 2015.

LAUDON, K. C.; LAUDON, J. P. Sistemas de informação. Rio de Janeiro: LTC, 1999.

LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. Ed. Novatec. 2. ed. São Paulo, 2010.

LEE, V.; SCHNEIDER, H.; SCHELL, R. Aplicações Móveis: Arquitetura, Projeto e Desenvolvimento. Ed. Pearson. 2005.

MARINO, V. C. C.; DUTKA, J. C. R.; KROOK, M. I. P.; GREGIO, A. M. L. **Articulação Compensatória Associada à Fissura de Palato ou Disfunção Velofaríngea: Revisão de Literatura**: Marília, 2011.

MATTAR, F. N. **Pesquisa de Marketing**. Rio de Janeiro: Elsevier, 5. ed., 2012.

MEDEIROS, E. **Desenvolvendo Software com UML Definitivo**. São Paulo: Pearson – Makron Books, 2006.

MELO, A. C. **Desenvolvendo Aplicações com UML 2.0**. Rio de Janeiro: Brasport, 2. ed., 2004.

OLIVEIRA, I. A.; **Interface de usuário**: A interação Homem-Computador através dos tempos. Revista Olhar Científico – Faculdades Associadas de Ariquemes, 2010, Disponível em: <<http://www.olharcientifico.kinghost.net/index.php/olhar/article/viewFile/28/29>>. Acesso em: 5 jan. 2015.

PASSOS, T. S. **Android, Arquitetura e Desenvolvimento**. 2009. 05 f. Monografia (Bacharel em Ciência da Computação) – UNINOVE, Poços de Caldas, 2009.

PEGORARO-KROOK MI, AFERRI HC, UEMEOKA E. Prótese de palato e obturadores faríngeos. In: Di Ninno CQMS, Jesus MV. Fissura Palatina: Fundamento para a prática fonoaudiológica. 1th ed. São Paulo: Roca; 2009. p. 113-24.

PEGORARO-KROOK MI, DUTKA-SOUZA JC, MAGALHÃES LCT, FENIMAN MR. Intervenção fonoaudiológica na fissura palatina. In: Ferreira LP, Befi-Lopes DM, Limongi SCO. Tratado de Fonoaudiologia. 2. ed. São Paulo: Roca; 2010. p. 439-55.

PINHO, S. M. R., CAMARGO, Z. Laboratório de voz e fala. In: FOZ, F. B., PICCARONE, M. L., BURSZTYN, C. S. **A tecnologia informática na fonoaudiologia**. São Paulo: Plexus Editora, 1998.

PRESSMAN, R. S.; **Engenharia de Software**. São Paulo: Makron Books, 2011.

RABELLO, R. R.; **Android: um novo paradigma de desenvolvimento móvel**, Pernambuco, 2000?, Disponível em: <[http://www.cesar.org.br/site/files/file/WM18\\_Android.pdf](http://www.cesar.org.br/site/files/file/WM18_Android.pdf)>. Acesso em: 3 jan. 2015.

RAPPS, S.; WEYUKER, E.J. **Data Flow analysis techniques for test data selection** In: International Conference on Software Engineering, p. 272-278, Tokio, Sep. 1982.

ROCHA, A. R. C. et al. **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

ROCHA, H. V.; BARANAUSKAS, M. C. C. **Design e avaliação de Interfaces Humano-Computador**. Campinas: NIED/UNICAM, 2003.

ROMEIRO, B. G. B. A. **Desenvolvimento de aplicativos para dispositivos móveis na plataforma j2me**. 2005. Monografia (Bacharelado em Engenharia de Computação) - Universidade de Pernambuco, Escola Politécnica de Pernambuco, 2005.

SANTOS, S. L.; TEIXEIRA, F. G. **Design de uma Interface de Interação Tridimensional com Foco na Usabilidade e no Desempenho Gráfico**. PgDesign. Porto Alegre, p 39-50, janeiro, 2010.

SANTOS, K. W.; FERNANDES, R. A.; VIDOR, D. C. G. M.; TRINDADE, C. S.; VIDOR, D. C. **Utilização de Softwares em Pesquisas Científicas de Fonoaudiologia**. Porto Alegre, 2012, Disponível em: < <http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/viewFile/185/116>>. Acesso em: 31 out. 2016.

SANTOS, K. W.; FERNANDES, R. A.; FERREIRA, G. G. TRINDADE, C. S.; VIDOR, D. C. **Utilização de Softwares por Fonoaudiólogos no Rio Grande do Sul**. Porto Alegre, 2015, Disponível em: < <http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/334>>. Acesso em: 3 abril. 2015.

SCHAEFER, C. **Protótipo de aplicativo para transmissão de dados a partir de dispositivos móveis aplicados a uma empresa de transporte**. 2004. 53 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, 2004.

SCHNEIDER, E.; SHPRINTZEN, R. J. A survey of speech pathologists: current trends in the diagnosis and management of velopharyngeal insufficiency. *Cleft Palate J*. 1980;17:249–253.

SILVA, E. G. **Desenvolvimento de um portal para análise da fala por avaliadores múltiplos**. 2016. 30 f. Projeto (Pós-doutorado em Ciências da Reabilitação) - Universidade de São Paulo, Hospital de Reabilitação de Anomalias Craniofaciais, 2016.

SILVA, R. S. S. **Fissuras Labiopalatinas**. 1999. 36 f. Trabalho de Conclusão de Curso (Graduação em Fonoaudiologia) – CEFAC – Centro de Especialização em Fonoaudiologia Clínica, 1999.

SOMMERVILLE, I. **Engenharia de Software**. Pearson – Addison Wesley, 3. ed., 2012.

SWEENEY, T.; SELL, D. Relationship between perceptual ratings of nasality and nasometry in children/adolescents with cleft palate and/or velopharyngeal dysfunction. *Int J Lang Comm Disord*. 2008; 43 (3): 265-82.

TANENBAUM, A. S.; **Sistemas Operacionais Modernos**. 2003 Pearson Prentice Hall, 2. ed., 2003.

VOGEL, L. **Android SQLite Database** - Tutorial. 2011. Disponível em <<http://www.vogella.de/articles/AndroidSQLite/article.html>>. Acesso em: 26 set. 2011.

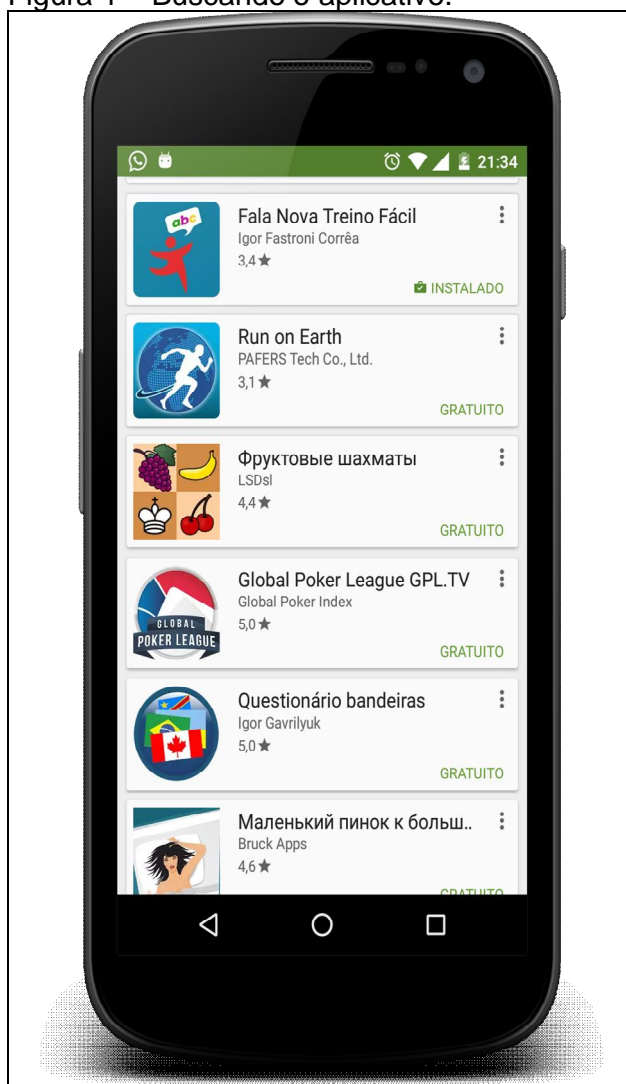
## APÊNDICE A – DOWNLOAD E INSTALAÇÃO DO APLICATIVO

A seguir nesta seção serão apresentados os passos para efetuar o *download* e instalação do aplicativo.

### Download

O aplicativo “Fala Nova Treino Fácil” está disponível para download de forma gratuita no Google Play para smartphones com versão do Android superior a 4.4. Para realizar o *download* do aplicativo basta acessar o aplicativo Google Play disponível no celular, e digitar como palavra-chave na busca “Fala Nova Treino Fácil”, conforme pode ser visualizado na Figura 1.

Figura 1 – Buscando o aplicativo.



Fonte: Elaborada pelo autor.

Após ter encontrado o aplicativo “Fala Nova Treino Fácil” no resultado da pesquisa, deve-se clicar em cima do logotipo do aplicativo, e imediatamente será aberta uma com a explicação (apresentação) do aplicativo, como pode ser observado na Figura 2.

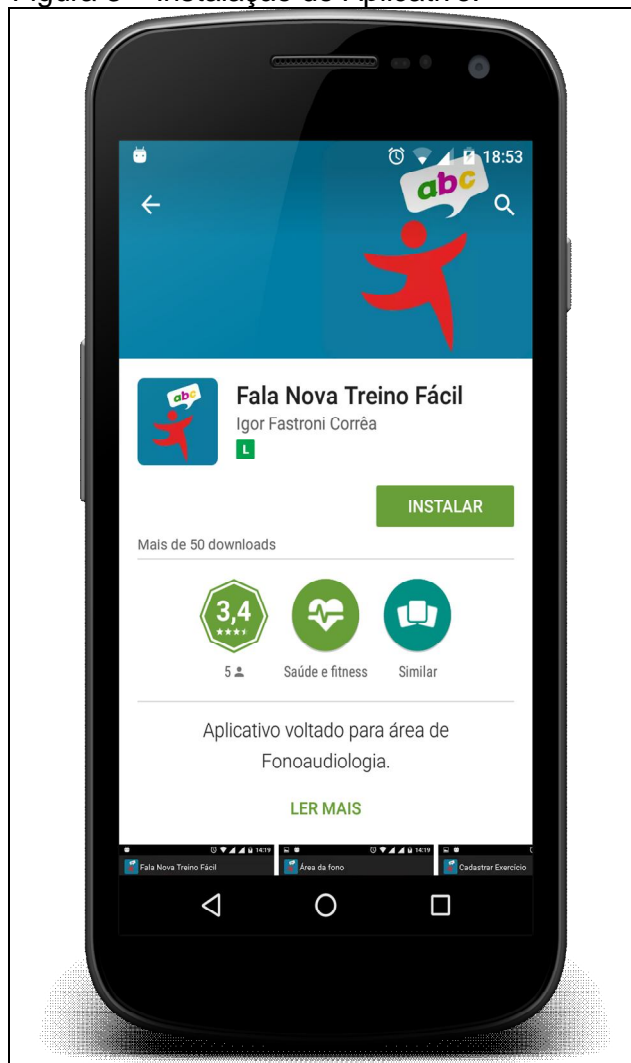
Figura 2 – Descrição do aplicativo.



Fonte: Elaborada pelo autor.

Para Instalar o aplicativo basta clicar no botão **instalar**, como pode ser visualizado na Figura 3, e o mesmo já estará pronto para utilização.

Figura 3 – Instalação do Aplicativo.



Fonte: Elaborada pelo autor.