

Universidade Federal do Pampa

Anderson Schwede Raugust

**SignMessenger: Um Aplicativo de Conversa
Utilizando o Sistema de Escrita de Sinais
SignWriting**

Alegrete

2015

Anderson Schwede Raugust

SignMessenger: Um Aplicativo de Conversa Utilizando o Sistema de Escrita de Sinais SignWriting

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Me. Sam da Silva Devincenzi

Coorientador: Prof. Me. Jean Felipe Patkowski Cheiran

Alegrete

2015

Anderson Schwede Raugust

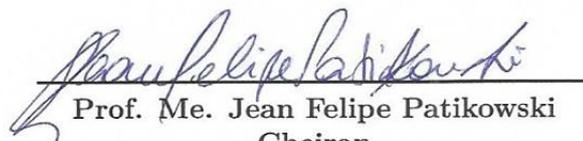
SignMessenger: Um Aplicativo de Conversa Utilizando o Sistema de Escrita de Sinais SignWriting

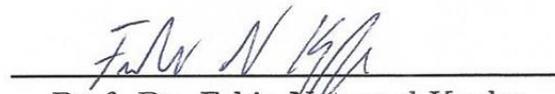
Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em 04 de dezembro de 2015.

Banca examinadora:


Prof. Me. Sam da Silva Devincenzi
Orientador


Prof. Me. Jean Felipe Patikowski
Cheiran
Co-orientador
UNIPAMPA


Prof. Dr. Fabio Natanael Kepler
UNIPAMPA


Prof^ª. Dr^ª. Aline Vieira de Mello
UNIPAMPA

Este trabalho é dedicado a Deus, primeiramente, por nunca ter deixado me faltar forças e por me conduzir até aqui, dando-me o conhecimento e sabedoria necessários para alcançar meus objetivos. Depois, dedico à minha família, mãe, pai e irmão, que sempre me apoiaram nas dificuldades, deram uma palavra de conforto nas horas difíceis e torceram pelo meu futuro. Agradeço à minha namorada pela paciência e companheirismo até aqui, durante esta difícil caminhada.

Agradecimentos

Ao orientador, professor Sam Devincenzi, pela indicação do tema do trabalho e por todo o suporte até a conclusão do mesmo.

Ao professor Jean Cheiran pelo tempo dedicado ao acompanhamento do trabalho e por todo o conhecimento que agregou, principalmente na área de Interação Humano-Computador.

À professora Ana Paula Gomes Lara e à intérprete Roberta Messa pelas sugestões e apoio na validação do sistema. Também, pela maravilhosa experiência e ensinamentos sobre o Surdo que me proporcionaram em cada encontro.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12:2)*

Resumo

Devido à necessidade humana de comunicação, convívio social, manter uma boa relação com todo o círculo de amigos e com familiares, os aplicativos de conversa continuam crescendo dia após dia. Paralelo a isso, mais de 70 milhões de pessoas no mundo são surdas (AHMED; SEONG, 2006), e possuem a língua de sinais como língua materna. Assim, precisam se comunicar na segunda língua, já que não possuem uma solução tecnológica para fazer isso de maneira prática e rápida. Para suprir a carência da escrita, a língua de sinais possui algumas formas de escrita, onde a mais popular e difundida é a representação em SignWriting, um sistema de escrita próprio para as línguas gestuais. Com o objetivo de incluir digitalmente o Surdo, possibilitando que ele se comunique na sua própria língua materna, garantindo também a acessibilidade e a igualdade de direitos, o presente trabalho propõe o desenvolvimento de um aplicativo de conversa utilizando a representação escrita de sinais SignWriting. Para a construção deste aplicativo, foi necessário um estudo aprofundado sobre Língua de Sinais, a representação SignWriting e os aplicativos de conversa atuais. A partir disso, foi definido para o desenvolvimento a metodologia Scrum. A técnica de prototipação em papel foi utilizada, para atender o apelo do produto por uma interface voltada ao Surdo. Foi definido o Android como plataforma alvo, utilizando a IDE AndroidStudio para a implementação. O resultado final obtido foi a ferramenta SignMessenger, que possibilita a conversa através de sinais, com interface intuitiva e base de dados totalmente na nuvem, melhorando a experiência do usuário Surdo. Assim, essa aplicação atende ao objetivo de prover uma forma para o Surdo se comunicar na sua língua materna.

Palavras-chave: Acessibilidade. SignWriting. Engenharia de Software. Interação Humano-Computador. Scrum.

Abstract

Due to the human need for communication, social interaction, to maintain good relationship with the whole circle of friends and family, or simply entertainment, messaging applications are continuing to grow day by day. Parallel to this, more than 70 million people in the world are deaf (AHMED; SEONG, 2006), and have the sign language as their mother tongue. So, they need to communicate in a second language as we do not have a technological solution to do this in a practical and fast way. Luckily, the sign language has a written form which is the SignWriting representation, a own writing system for sign languages. In order to digitally include the Deaf, enabling him to communicate in their own mother tongue, also ensuring accessibility and equal rights, this paper proposes the development of a instant messaging application using the written representation of SignWriting signals. For the construction of this application, an in-depth study was needed on sign language, the SignWriting representation and the current messaging applications. From this, it was decided to develop with the Scrum methodology. Paper prototyping technique was used to attend the product's appeal for an interface dedicated to the Deaf. Android was defined as target platform, using the IDE AndroidStudio for implementation. The final result was the SignMessenger tool, enabling conversation through signals with intuitive and fully cloud database, improving the user experience of the Deaf. Thus, this application serves the purpose of providing a way for the Deaf to communicate in their mother tongue.

Key-words: Accessibility. SignWriting. Software Engineering. Human-Computer Interaction. Scrum.

Lista de ilustrações

Figura 1 – Descrição de uma criança Surda (RODRIGUES, 2011)	21
Figura 2 – Posições básicas de mãos (Fonte: SignWriting.org)	22
Figura 3 – Configurações de mão (Fonte: SignWriting.org	22
Figura 4 – Sinal composto (STUMPF, 2005)	22
Figura 5 – Sinais para Surdos (STUMPF, 2005)	23
Figura 6 – Representação da palavra bonito	23
Figura 7 – Trecho de Cinderela Surda (HESSEL FABIANO ROSA, 2007)	24
Figura 8 – Volume de tráfego de dados no brasil (ERICSSON, 2014)	25
Figura 9 – Arquitetura e funcionamento de um serviço XMPP (CHANG; CHEN, 2011)	26
Figura 10 – Comunicação XMPP (Fonte: www.ibm.com)	27
Figura 11 – Processo de desenvolvimento Scrum. (Fonte: desenvolvimentoagil.com)	36
Figura 12 – Product Backlog inicial	37
Figura 13 – Caso de Uso 1 - Enviar mensagem em SignWriting	38
Figura 14 – Diagrama de classes	38
Figura 15 – Diagrama de Implantação	39
Figura 16 – Diagrama Entidade-Relacionamento	40
Figura 17 – Casos de teste para o UC1	40
Figura 18 – Product Backlog final	44
Figura 19 – Arquitetura MVC no Android. (Fonte: Programming Android, 2nd Edition by Mednieks, Dornin, Meike, and Nakamura)	45
Figura 20 – Protótipo 1	47
Figura 21 – Protótipo 2	47
Figura 22 – Hierarquia do menu de simbolos até o indicador	48
Figura 23 – Interface da aplicação - Tela de Login e de lista de usuários	49
Figura 24 – Interface da aplicação - Tela de conversa e de desenho	49
Figura 25 – Resultados das duas baterias de testes	51
Figura 26 – Diagrama de sequência	68

Sumário

1	INTRODUÇÃO	17
2	REFERENCIAL TEÓRICO	19
2.1	Língua de Sinais	19
2.2	SignWriting	20
2.3	Aplicativos de Conversa	24
2.4	Protocolo XMPP	26
2.5	Interação Humano-Computador	26
2.6	Conclusão do Capítulo	28
3	TRABALHOS RELACIONADOS	29
3.1	SignWriting	29
3.2	Aplicativos de Conversa	31
3.3	Aplicativos com SignWriting	32
3.4	Conclusões do Capítulo	33
4	METODOLOGIA	35
4.1	Descrição do Trabalho	35
4.2	Metodologia Scrum	35
4.2.1	Product Backlog	36
4.2.2	Especificação dos Casos de Uso	36
4.2.3	Projeto	37
4.2.4	Qualidade do Software	39
4.3	Prototipação e Validações com Usuário	40
4.4	Conclusão do capítulo	41
5	SIGN MESSENGER	43
5.1	Product Backlog	43
5.2	Engenharia	43
5.3	Ferramentas	44
5.3.1	Base64	45
5.3.2	Parse	45
5.3.3	Sinch	46
5.4	Interface	46
5.5	Conclusão do capítulo	48
6	TESTES E RESULTADOS	51

6.1	Casos de teste	51
6.2	Validação com usuários	51
6.3	Conclusões do capítulo	52
7	CONSIDERAÇÕES FINAIS	53
	REFERÊNCIAS	55
	APÊNDICES	59
	APÊNDICE A – CASOS DE USO EXPANDIDOS	61
	APÊNDICE B – CASOS DE TESTE	65
	APÊNDICE C – PROJETO	67

1 Introdução

A computação e a internet seguem aproximando pessoas em todo o mundo, tornando-se cada vez mais o principal meio de comunicação. Com a chegada dos smartphones esse cenário se afirmou, quebrando as barreiras físicas que impediam as pessoas de estarem mais tempo conectadas à rede. Permitir um contato contínuo com familiares e amigos torna esta missão de comunicar ainda mais importante. Uma das maneiras de garantir este objetivo é através dos aplicativos de conversa, que já recebem bilhões de mensagens diariamente, com rapidez, facilidade e de forma gratuita.

Concomitantemente, 70 milhões de pessoas no mundo são surdas ([AHMED; SE-ONG, 2006](#)), o que torna a surdez a principal deficiência no mundo. A maioria das pessoas que possuem esta deficiência utilizam a língua de sinais para se comunicar, seja com outras pessoas com a mesma deficiência ou com ouvintes. A língua de sinais é uma ferramenta não só para a comunicação, mas também para a educação, necessidade de acesso à informação, entre outros, adquirindo assim grande importância. A linguagem de sinais ainda tem seu valor cultural. Aliás, vale lembrar que o Surdo tem sua própria cultura, valores e linguagem, coisas que geralmente são esquecidas pela população em geral, portanto eles precisam de ferramentas específicas para comunicar-se entre si e com outros membros da sociedade.

O paradigma da educação bilíngue considera que duas línguas co-existem no mesmo espaço, sendo que a primeira língua (língua materna) do Surdo é a língua de sinais, enquanto a segunda língua é a língua oral da comunidade ouvinte em que estão inseridos. Portanto, eles devem ser capazes de produzir conhecimento em sua própria língua materna, e precisam de softwares que dêem o suporte necessário a essa necessidade ([ANTUNES et al., 2011](#)).

Para dar esse suporte, é necessário uma forma de representar os sinais graficamente de uma maneira padronizada e que todos os Surdos que conheçam uma língua de sinais consigam entender. O SignWriting é um sistema de escrita da língua de sinais, como se fosse o alfabeto romano em relação ao português. Com ela, esta escrita pode ser padronizada e ferramentas podem assim ser desenvolvidas ([SUTTON, 2015](#)).

O desafio de juntar ambas as áreas em uma única proposta dá a oportunidade de fazer com que todos se comuniquem utilizando a sua língua materna, o que é praticamente impossível se tratando do Surdo e a computação atual, devido à grande escassez de serviços que permitam isso na área.

O objetivo deste trabalho é a construção de uma aplicação que permita a comunicação entre pessoas que utilizem a língua de sinais via sua representação escrita

(SignWriting), garantindo acessibilidade ao Surdo, ou seja, podendo incluí-la no desenvolvimento de recursos que promovam o pleno acesso ao conhecimento, para assim permitir a eles o total exercício da cidadania. Portanto, a aplicação tem como principal objetivo promover a inclusão desta classe no novo universo da comunicação.

2 Referencial Teórico

Neste capítulo abordaremos toda a base teórica deste trabalho, que se dá a partir de questões ligadas aos conceitos de língua de sinais e da linguagem de sinal escrita SignWriting, necessárias para a compreensão e desenvolvimento da proposta.

Outro foco deste estudo são os aplicativos de conversa e sua importância para a sociedade atual, o porquê de eles serem tão populares e estarem tão ligados ao nosso dia-a-dia. Também são foco deste capítulo questões técnicas e dados de sistemas já existentes.

2.1 Língua de Sinais

Um dos principais aspectos da cultura surda é a língua de sinais. Segundo [Strobel \(2013\)](#), a língua de sinais representa a visão que o Surdo tem do mundo, e através dela ele pode criar uma ligação com outros Surdos, mesmo com os que não tem conhecimento de uma língua de sinais específica, já que os sinais buscam ser intuitivos. Ainda, a autora afirma que uma criança surda que não tem nos seus primeiros anos de vida um acompanhamento de especialista ou a convivência com uma família surda, acaba demorando muito mais para desenvolver uma linguagem, o que pode acarretar em insegurança, baixa-autoestima e, em alguns cenários, crises de identidade.

Uma língua de sinais é um sistema de representação classificado como de modalidade gestual-visual que apresenta uma organização, estrutura formal e gramatical própria. É utilizada pela grande maioria das pessoas surdas em todo o mundo, pelas quais a língua oral é tida como uma segunda língua. No Brasil, a língua amplamente utilizada é a Língua Brasileira de Sinais (LIBRAS), sendo assim chamada desde 1993.

No que se refere a LIBRAS no quesito legal, existem dois capítulos de extrema importância, definindo e regulamentando o seu uso. O primeiro trecho ([BRASIL, 2002](#)) é definido pela Lei N° 10.436, de 24 de abril de 2002, que determina o seguinte sobre a Língua:

Art. 1º É reconhecida como meio legal de comunicação e expressão a Língua Brasileira de Sinais - Libras e outros recursos de expressão a ela associados.

Parágrafo único. Entende-se como Língua Brasileira de Sinais - Libras a forma de comunicação e expressão, em que o sistema lingüístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema lingüístico de transmissão de idéias e fatos, oriundos de comunidades de pessoas surdas do Brasil. ([BRASIL, 2002](#))

O segundo trecho ([BRASIL, 2005](#)) acrescenta a obrigatoriedade da disciplina em

cursos associados com Pedagogia e Magistério, além da obrigação dos demais cursos de educação superior de ofertá-la como disciplina optativa. O trecho segue abaixo:

Art. 3º A Libras deve ser inserida como disciplina curricular obrigatória nos cursos de formação de professores para o exercício do magistério, em nível médio e superior, e nos cursos de Fonoaudiologia, de instituições de ensino, públicas e privadas, do sistema federal de ensino e dos sistemas de ensino dos Estados, do Distrito Federal e dos Municípios.

§ 1º Todos os cursos de licenciatura, nas diferentes áreas do conhecimento, o curso normal de nível médio, o curso normal superior, o curso de Pedagogia e o curso de Educação Especial são considerados cursos de formação de professores e profissionais da educação para o exercício do magistério.

§ 2º A Libras constituir-se-á em disciplina curricular optativa nos demais cursos de educação superior e na educação profissional, a partir de um ano da publicação deste Decreto. /citebrasil2005

No contexto educacional, como vimos neste decreto, LIBRAS está garantida por lei, pelo menos no ensino superior e no profissional. Segundo [Campos \(2001\)](#), a gramática própria que LIBRAS possui não é a única característica que dificulta o seu aprendizado, ela possui um sistema fonológico que estuda as posições das mãos, os movimentos e pontos de articulação. Além disso, possui um Sistema Morfológico que trabalha com a formação de cada sinal, considerando peculiaridades de sua composição, como gênero, tempo, grau e negação. Há ainda o Sistema Sintático, que como o próprio nome diz, trata das questões ligadas à sintaxe que, em LIBRAS, possui um padrão Objeto – Sujeito – Verbo para compor as sentenças.

LIBRAS não possui representação para artigos, grande parte das preposições e de conjunções utilizadas no português, inclusive os tempos verbais e modos são produzidos com sinais de mão. Então, como no exemplo da frase em português "eu fui para casa de carro", em LIBRAS, a frase ficaria da seguinte maneira: "EU IR CASA CARRO"([ALMEIDAL, 2012](#)). A Figura 1 representa a forma como uma criança Surda acaba montando suas sentenças em português.

Segundo [Gesser \(2009\)](#), até pouco tempo atrás, a língua de sinais era considerada uma língua sem escrita, que não possuía um sistema de representação gráfica equivalente. Nesta necessidade, surge a linguagem de sinais escrita (SignWriting), abordada na próxima seção.

2.2 SignWriting

SignWriting surgiu através de pesquisadores da Dinamarca, que, em 1974, se basearam em um sistema de escrita de danças criado por Valerie Sutton, que utilizava sinais para descrever os passos e movimentos, e a partir deste sistema iniciaram uma pesquisa

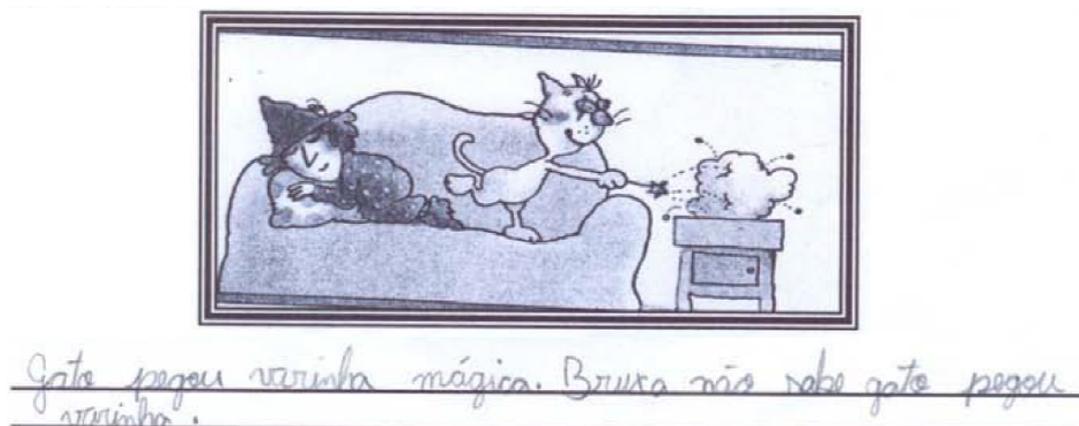


Figura 1 – Descrição de uma criança Surda (RODRIGUES, 2011)

que se espalhou por diversos países e que vem se popularizando gradativamente entre os Surdos.

Segundo a definição de Sutton (2015), SignWriting é um sistema de escrita que usa símbolos visuais para representar configurações de mãos, movimentos, e expressões faciais de línguas de sinais. Uma espécie de alfabeto que pode ser usado para escrever qualquer língua de sinal existente. Uma comparação interessante é entre o alfabeto romano, que usamos para escrever a grande maioria das línguas orais, e o SignWriting, para línguas de sinais.

A forma como os símbolos são apresentados graficamente são descritos por Boutora (2003) :

"Retângulos virtuais compreendem um conjunto de símbolos alinhados verticalmente. Um conjunto de símbolos representa um signo, quer dizer, uma unidade lexical eventualmente associada a um complemento de informações gramaticais, os signos são separados por um espaço. No interior de um "signo etiqueta" ou "pilha", os símbolos são colocados verticalmente segundo a lógica do corpo humano. Assim, o círculo que configura a cabeça suporta os símbolos que representam os elementos manuais. Os elementos não manuais (essencialmente os movimentos da face e o olhar) são inscritos dentro do círculo cabeça." (BOUTORA, 2003, 80)

Nesta estrutura da representação, como citado pela autora, um conjunto de símbolos representa um signo, que é a representação equivalente a um sinal completo da Língua de Sinais. O símbolo mais básico de um signo é a mão que, como visto na Figura 2, pode ter muitas configurações, que representam sua posição em três dimensões. Um signo também é composto de expressões faciais e movimentos, os quais seguem nas figuras.

Existem dez categorias de símbolos: mãos, contato das mãos, faces, ombro, localização, inclinação da cabeça, movimentos do corpo e da cabeça, membros, movimento de dinâmicas e pontuação. Dentro de cada categoria, ainda existem grupos.

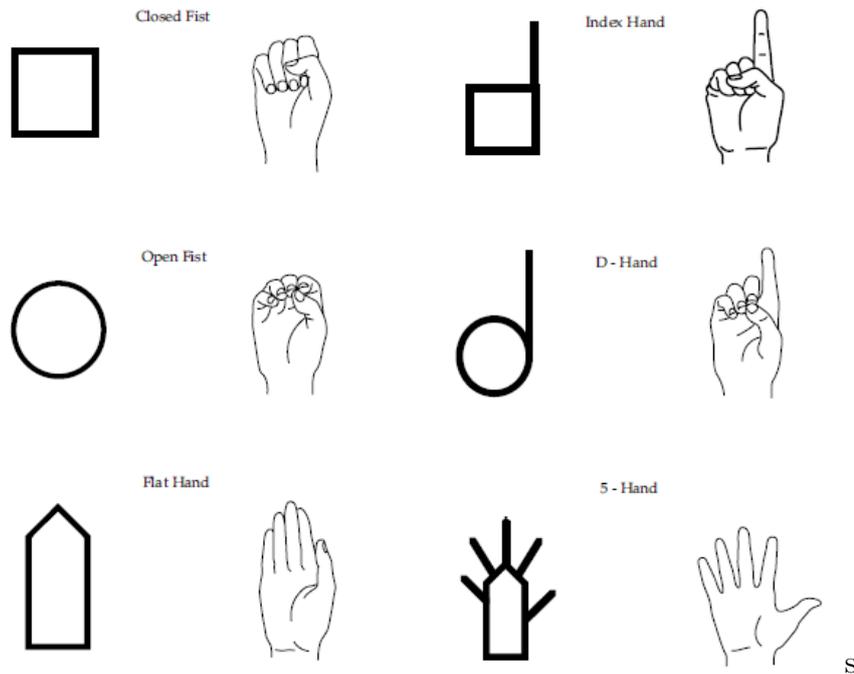


Figura 2 – Posições básicas de mãos (Fonte: SignWriting.org)

Posições de mãos e sua localização no espaço são representadas em SignWriting como na Figura 3, onde a forma de preenchimento do signo representa o ângulo da mão em relação ao indivíduo, o que colabora com a clareza do sinal.



Figura 3 – Configurações de mão (Fonte: SignWriting.org)

Por questões de economia, alguns sinais podem representar duas configurações diferentes num mesmo conjunto (chamado de "pilha"), como no exemplo da figura 4. São os chamados sinais compostos.

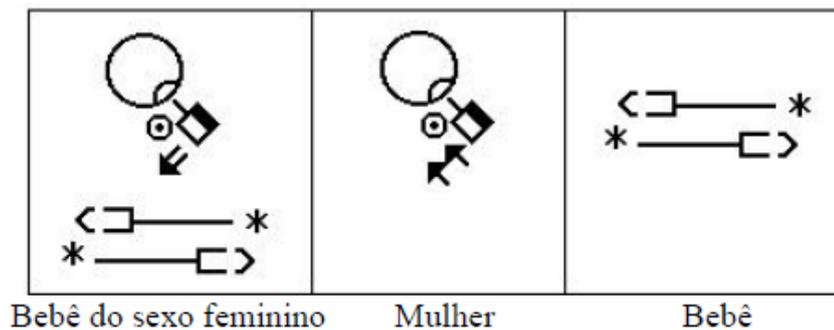


Figura 4 – Sinal composto (STUMPF, 2005)

Outra característica do SignWriting é a não padronização de como uma palavra é representada, como no exemplo de [Stumpf \(2005\)](#), visto na Figura 5, em que a palavra surdo pode ser representada de cinco maneiras diferentes, e todas estão corretas. O escritor tem a liberdade de grafar da forma como achar conveniente, geralmente como costuma usar no dia-a-dia na sua língua de sinais.



Figura 5 – Sinais para Surdos ([STUMPF, 2005](#))

Para comparar o signo em LIBRAS e em SignWriting, a figura 6 apresenta a palavra "bonito" representada por ambas.

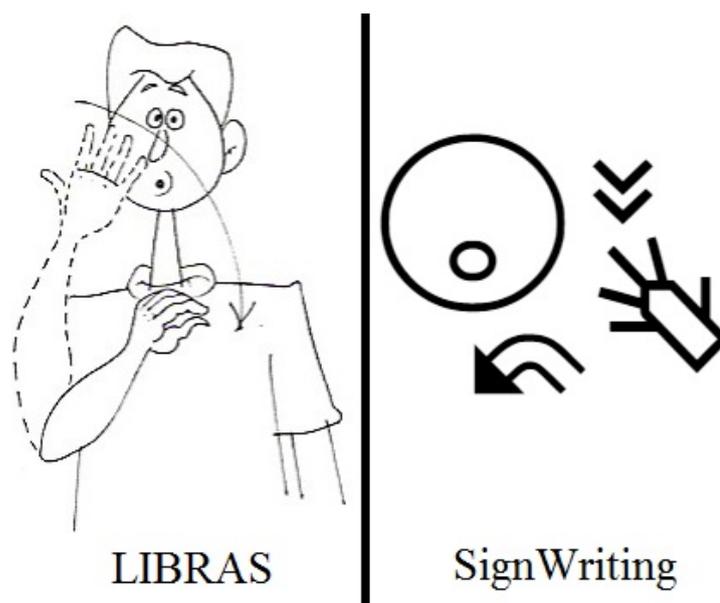
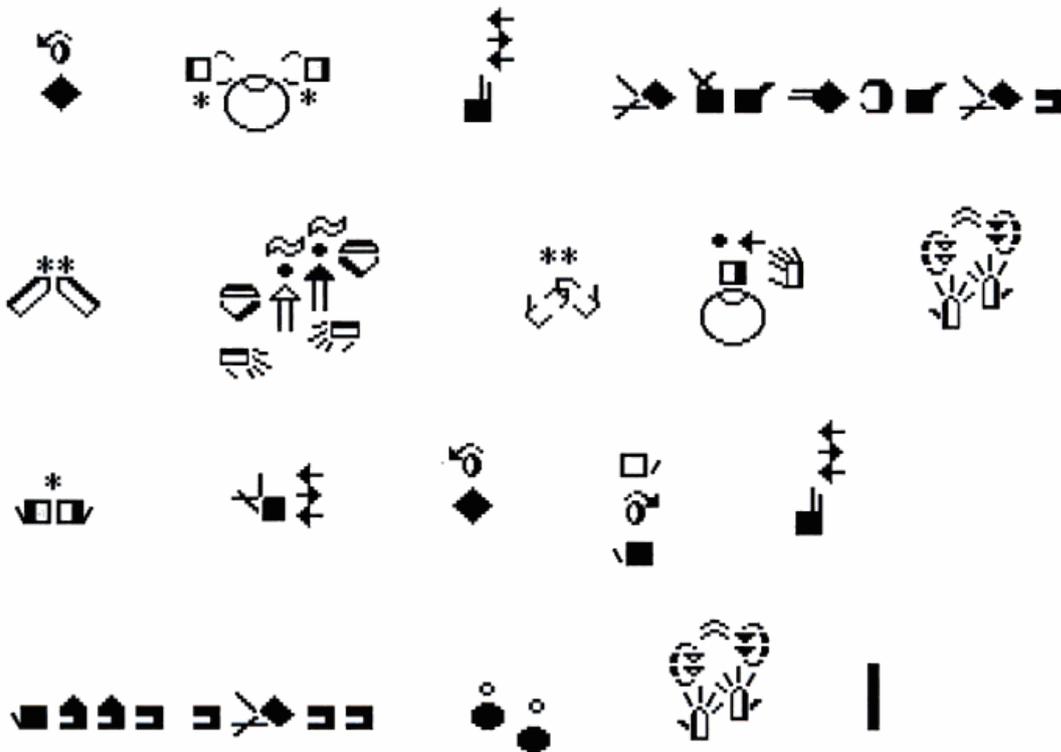


Figura 6 – Representação da palavra bonito

Para exemplificar o material apresentado até agora nesta seção, a Figura 7 representa um trecho da história de Cinderela Surda, uma adaptação do clássico da literatura, por [Hessel Fabiano Rosa \(2007\)](#).

[Strobel \(2013\)](#) afirma que a criação e adoção do sistema de escrita SignWriting foi um marco para o Surdo, pois antes disto, a língua de sinais era considerada ágrafa, ou seja, que não podia ser escrita. Hoje, este sistema é conhecido no Brasil como Escrita em Língua de Sinais (ELS), e algumas universidades já ofertam esta disciplina. Esta é uma forma de mostrar que o surdo pode vir a ter uma alfabetização escrita na sua própria língua materna.



Cinderela era filha de nobres franceses e aprendeu a Língua de Sinais Francesa com a comunidade de surdos, nas ruas de Paris.

O rei e a rainha contrataram o mestre L'Epeé para ensinar a Língua de Sinais Francesa ao Príncipe herdeiro do trono.

Figura 7 – Trecho de Cinderela Surda (HESSEL FABIANO ROSA, 2007)

2.3 Aplicativos de Conversa

O sucesso das redes sociais nos últimos anos é uma das revoluções da era da internet. Mais recentemente, com a chegada e popularização dos Smartphones, os aplicativos de conversa se tornaram uma força nesses novos tempos da comunicação.

Bueno (2014) afirma que em 2013, mesmo com a crise financeira no Brasil, a venda de smartphones cresceu cerca de 122%. Com o acesso aos aparelhos aumentando, este tipo de serviço tende a se popularizar ainda mais.

Neste tipo de plataforma, muitas aplicações já se firmaram no mercado, como Facebook Messenger, Viber, Skype, WeChat, Telegram, incluindo até o clássico ICQ. Cada uma possui características próprias pelas quais são preferidas por certos usuários, como o Skype, referência para conversa através de vídeo.

Entre os mais populares, encontra-se o WhatsApp, um dos ícones dessa categoria. Segundo Reis (2014), em um único dia, o WhatsApp chegou a registrar 27 bilhões de mensagens enviadas. Um marco para a história da comunicação.

O site oficial do WhatsApp descreve o serviço da seguinte maneira:

(...) um aplicativo de mensagens multiplataforma que permite trocar mensagens pelo celular. Não há custo para enviar mensagens e ficar em contato com seus amigos. Além das mensagens básicas, os usuários do WhatsApp podem criar grupos, enviar imagens, vídeos, local, contatos e áudio.

No Brasil, segundo levantamento de Ericsson (2014), o tráfego de dados produzido pelo WhatsApp é o quarto maior. Se levarmos em conta que, mesmo com o suporte a imagens, áudio e vídeo, as mensagens de texto são o que realmente dominam o tráfego, podemos ter uma ideia da quantidade de mensagens que transitam na rede. A Figura 8 representa bem a classificação do levantamento.

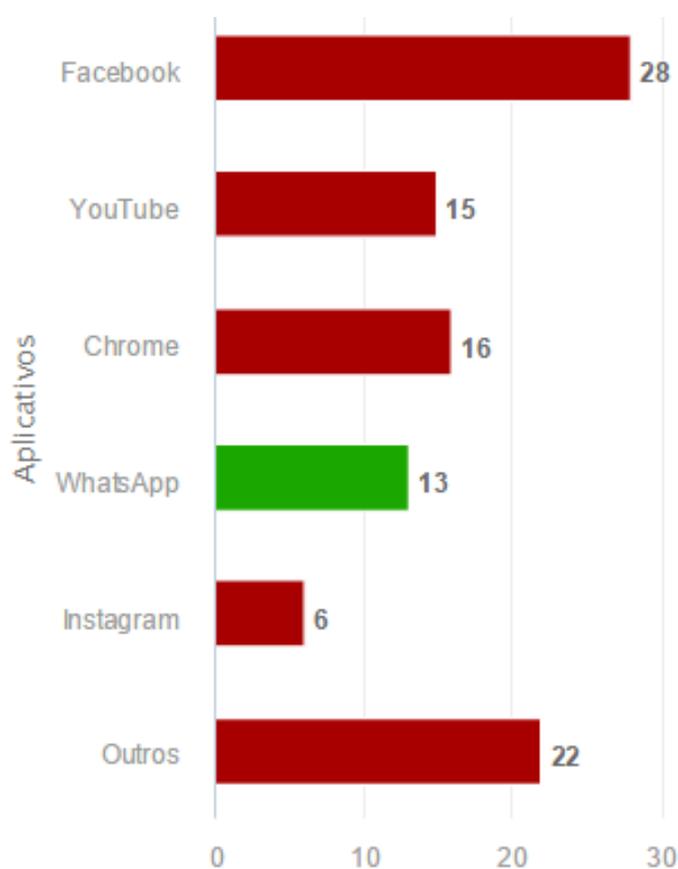


Figura 8 – Volume de tráfego de dados no Brasil (ERICSSON, 2014)

2.4 Protocolo XMPP

Do ponto de vista técnico, os aplicativos de conversa possuem em sua arquitetura a implementação de XMPP (protocolo de presença e mensagem extensível) que é utilizado também em salas de bate-papo, e agora também nos novos aplicativos de conversa. Ele foi desenvolvido originalmente para mensagens instantâneas e informações de presença. Segundo [Chang e Chen \(2011\)](#), ele permite o transporte de qualquer tipo de dados XML estruturados, inclusive. É comumente usado em aplicações críticas por sua confiabilidade.

Como vemos na Figura 9, a rede XMPP é composta na prática por clientes e servidores, onde cada cliente ganha um endereço único na rede. Como consequência, a comunicação fim-a-fim (de um cliente até o outro) funciona como ponto-a-ponto, embora esteja disposta fisicamente como cliente-servidor.

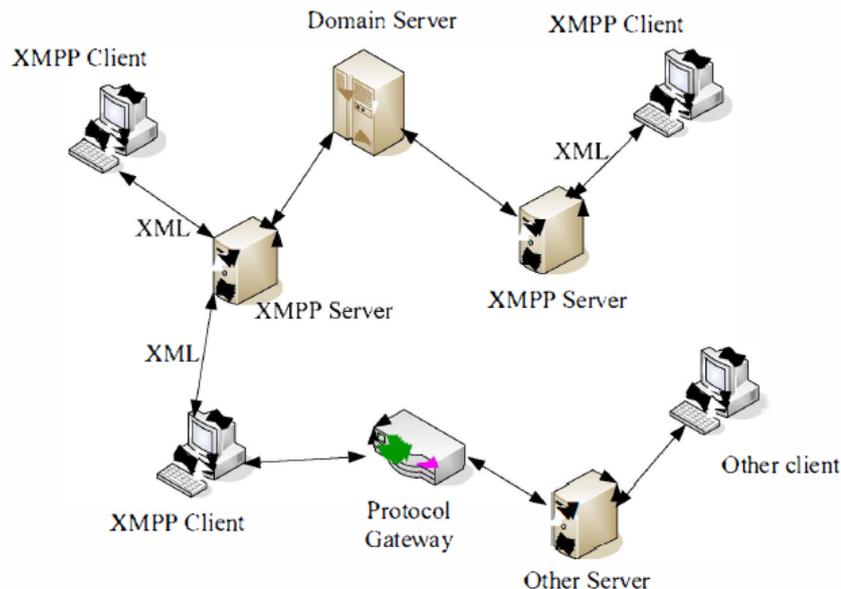


Figura 9 – Arquitetura e funcionamento de um serviço XMPP ([CHANG; CHEN, 2011](#))

XMPP funciona sobre os sockets do protocolo TCP (*transport control protocol*) usando mensagens XML (*extensible markup language*). A Figura 10 mostra uma conversa simples entre duas entidades, onde o cliente que inicia o contato manda uma solicitação de stream em XML, o receptor recebe e responde também com um XML stream. Então eles trocam mensagens e, após, a conexão é terminada.

2.5 Interação Humano-Computador

A análise da Interação Humano-Computador no contexto de smartphones já inicia com um desafio: projetar as interfaces para os dispositivos móveis. Devido ao espaço

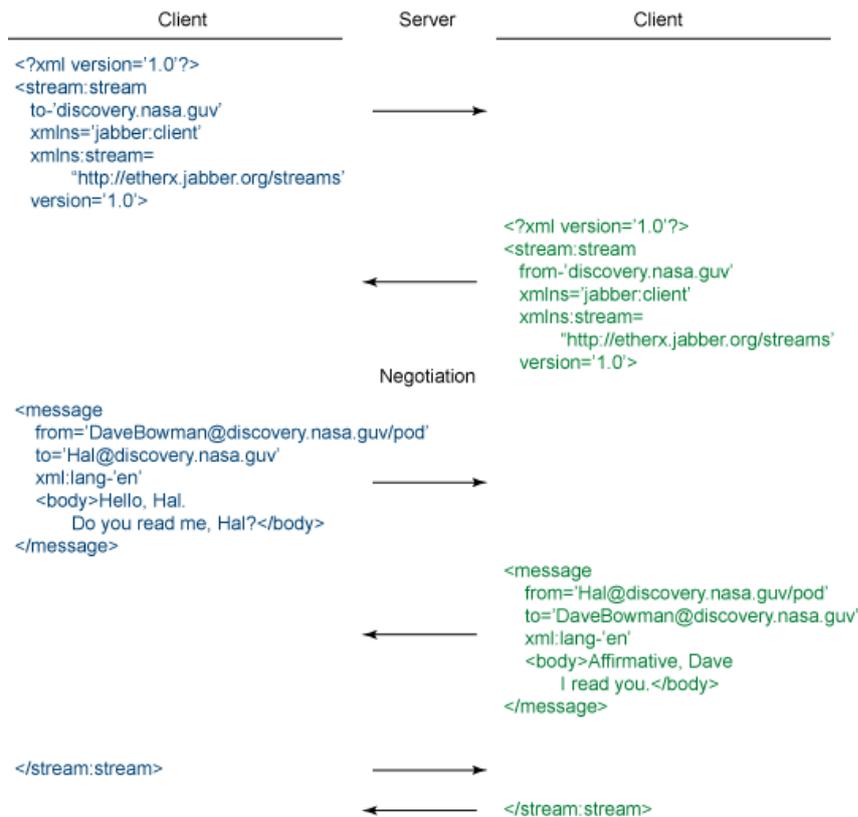


Figura 10 – Comunicação XMPP (Fonte: www.ibm.com)

reduzido para a exibição de informações e ao toque não ser tão preciso quanto um clique em um cenário desktop.

Para isso, é necessário evitar o que Nielsen (2014) chama de síndrome do dedo gordo, onde o usuário precisa lutar para tocar em áreas minúsculas na tela para concluir sua ação. Isso pode ser garantido com um design mais simples, exibindo menos informação em cada tela, mesmo que seja necessário aumentar o número de telas para garantir que sejam mais enxutas. Com menos informação, elas podem ser maiores, facilitando a manipulação pelo usuário.

Esse apelo por simplicidade ganha força quando o público alvo é o Surdo, que é extremamente visual. A poluição da tela com a exibição de um grande número de informações pode tornar a experiência desse usuário difícil e angustiante. Portanto, Nielsen afirma que em dispositivos que possuem tela pequena, os recursos visuais precisam ser limitados aos que tem maior importância ao caso do usuário.

Um dado importante que Nielsen destaca, que auxilia na definição de uma plataforma para desenvolvimento, é que usuários de aplicações desenvolvidos para uma plataforma específica são mais felizes que os usuários de sites móveis. Isso se explica pela otimização da experiência proporcionada pelo design para uma plataforma específica, diferente de sistemas que precisam ser transformados para funcionarem em muitas plata-

formas diferentes.

2.6 Conclusão do Capítulo

Neste capítulo foi possível conhecer mais sobre a cultura surda, suas peculiaridades e necessidades no quesito linguístico, analisando a Língua de Sinais e o que referências no assunto afirmam ao seu respeito. Após, a representação da língua de sinais foi o foco, onde vimos a história por trás de SignWriting, como se popularizou no mundo, qual a relação desses símbolos com a língua de sinais e detalhes sobre o seu funcionamento. Nas últimas sessões tratamos dos aplicativos de conversa, que são tão populares hoje em dia. Tratamos de sua concepção, como são construídos, seu funcionamento, funcionalidades interessantes à proposta deste trabalho e, principalmente, a importância destas aplicações nas relações pessoais e profissionais, ou seja, quais benefícios eles nos trazem e poderão também levar ao Surdo.

3 Trabalhos Relacionados

O capítulo apresentará o que já existe de pesquisa e desenvolvimento sobre aplicativos com SignWriting e demais áreas relacionadas com esta pesquisa, analisando sua contribuição para o estudo e desenvolvimento do presente trabalho e fortalecendo os pontos pelos quais ele se destaca.

3.1 SignWriting

A proposta de [Guimarães, Guardazi e Fernandes \(2014\)](#) trata de um framework que auxilia no aprendizado de pessoas com deficiência, em específico crianças que não nasceram em uma família surda, tendo assim o aprendizado prejudicado pela falta de informação dos pais sobre a importância de uma língua de sinais escrita no desenvolvimento pedagógico da criança. Portanto, o framework tem como objetivo servir de base para o desenvolvimento de ferramentas para alfabetização escrita em língua de sinais.

Esse trabalho se destaca pela criação do framework com uma interface intuitiva para rápido aprendizado do usuário, seja ele surdo, ouvinte, adulto ou criança. Para tanto, realizou um estudo sobre os métodos e a arquitetura pedagógica e, para servir de base para a construção do framework, foi realizado um estudo de caso com dezesseis crianças e quatro professores surdos que não tiveram contato com o SignWriting. Os professores consideraram a ferramenta importante para fazer com que os pais (ouvintes) das crianças também aprendam e se interessem pela língua de sinais, para assim aumentar o interesse dos filhos nela, o que aumentaria a velocidade do aprendizado e facilitaria a comunicação da criança surda com o resto dos Surdos.

Tal estudo serviu como base para destacar a importância de ferramentas com SignWriting para o aprendizado das crianças, pois, como em qualquer outra língua, na língua de sinais a prática também é necessária.

Por outro lado, a proposta de [Bianchini, Borgia e Marsico \(2012\)](#) apresenta um editor online de SignWriting, que pode ser visto como uma alternativa à linguagem vocal, como o próprio autor define. Esse editor tem como argumento a promessa de ser anexável em outros editores padrões, auxiliando principalmente no aprendizado eletrônico dos usuários (e-learning).

Bianchini utiliza uma boa base de IHC para a construção da ferramenta, destacando os princípios de usabilidade e fazendo uma avaliação da mesma com usuários Surdos. Dez usuários participaram da avaliação e detectaram alguns erros nas funcionalidades, que foram corrigidos. Outro problema encontrado foi a grande quantidade de elementos

necessários para construir um símbolo, os quais pareciam não estar muito intuitivos para os usuários.

Essa construção de interface para construção e envio de sinais, e os resultados desses testes de usabilidade nessa interface são o ponto onde a proposta de Bianchini se relaciona com a proposta do presente trabalho, contribuindo para a futura construção da interface de envio. Porém, a atual proposta se diferencia por levar em conta a opinião do usuário durante o processo de desenvolvimento, com validações periódicas, o que garante um software com a cara do usuário.

Já o software criado por [Matsumoto, Kato e Ikeda \(2009\)](#) pode ser definido como a escrita da Língua de Sinais Japonesa em SignWriting, permitindo que os usuários construam as sentenças em um tempo menor, já que utiliza símbolos prontos e evita que o usuário tenha que construir cada símbolo.

Nesse software, escrever símbolos japoneses e caracteres auxiliares são uma maneira rápida de chegar ao símbolo pretendido. A estimativa de Matsumoto é de ter mais de três mil símbolos no software.

O ganho de tempo com os símbolos prontos tem como contrapartida um software com uma interface menos intuitiva e mais estática, tirando a liberdade do usuário em montar os próprios símbolos, como as demais ferramentas apresentaram. O estudo desse artigo contribui com essa nova hipótese de construção, colocando o desempenho no lugar de liberdade do usuário e simplicidade da interface.

O trabalho de [Almasoud e Al-Khalifa \(2011\)](#) consiste em um sistema que faz a tradução do árabe escrito para a Língua de Sinais Árabe (ArSL), representada em SignWriting. O grande destaque é o enfoque pela semântica da tradução, pois como já vimos, uma língua de sinais tem uma semântica peculiar em cada símbolo, por isso esse sistema leva em conta todas as regras de tradução e utiliza uma ontologia de domínio para chegar ao objetivo da notação em ArSL.

A arquitetura do sistema de Almasoud é constituída de um grupo de processos sequenciais: Análise Morfológica, Transformação Gramatical e Tradução Semântica. A Análise Morfológica consiste em receber o texto em árabe e fazer a classificação gramatical em cada sentença. A Transformação Gramatical utiliza o resultado da fase anterior e aplica as regras da Língua de Sinais Árabe em cada palavra dependendo de sua função na frase. Por fim, a Tradução Semântica procura cada palavra na Ontologia de Domínio para atribuir-lhe o símbolo correspondente, ou de um sinônimo, caso a palavra não tenha um símbolo atribuído no banco de dados da aplicação, ou, ainda, soletrar a palavra com o alfabeto manual, na pior das hipóteses.

Esse artigo sobre o sistema de tradução para ArSL contribui para a proposta atual através do estudo feito sobre língua de sinais, que foi necessário para construir a proposta,

e também sobre SignWriting e a construção de uma ferramenta que suporte todos os símbolos necessários, conseguindo posteriormente identificar o símbolo construído.

3.2 Aplicativos de Conversa

Do ponto de vista da interface, que é um dos focos do desenvolvimento, o artigo escrito por [Choi e Lee \(2012\)](#) apresenta uma abordagem sobre o impacto da simplicidade na aprovação dos usuários no contexto de smartphones. Utilizando medidas para chegar a um denominador da simplicidade de uma tela, o autor avalia os impactos da cultura e do nível de conhecimento dos usuários nesse levantamento.

Antes de pensar em resultados, foi necessário criar uma relação que chegasse até esse denominador comum, e para isso foi necessário relacionar seis componentes: redução, organização, complexidade dos componentes, complexidade de coordenação, complexidade dinâmica e estética visual. A partir disso, foi possível avaliar a interface de aplicações para smartphones e comparar essa avaliação com a opinião de usuários.

Esta avaliação criada por Choi e sua revisão sobre IHC são elementos que contribuem para o presente projeto. Porém, o próprio autor afirma que usuários mais instruídos não teriam problema em interpretar um número maior de informações na tela do smartphone, assim como pessoas que lidam com grande quantidade de informações todos os dias, ou seja, a amostra de usuários pode influenciar no resultado dessa análise, o que demonstra certa fraqueza da medida.

O artigo de [Chang e Chen \(2011\)](#) traz uma abordagem sobre o protocolo de comunicação XMPP (extensible messaging and presence protocol), protocolo que é usado de base para os aplicativos de conversa populares. Esse trabalho descreve o projeto de uma aplicação do tipo SOA (Arquitetura baseada em serviços) utilizando como base o protocolo XMPP.

O objetivo do trabalho de Chang é mostrar essa alternativa que empresas de grande porte e que se preocupam com a segurança e privacidade de suas conversas encontram: construir o seu próprio servidor de mensagens. Segundo ele, também, essa alternativa geralmente é adotada por empresas que trabalham com grandes quantidades de dinheiro por questões de segurança.

A alternativa de um servidor XMPP levantada por Chang e o estudo feito em cima dele são uma contribuição para a atual proposta, visto as qualidades esperadas dos aplicativos de conversa, como o desempenho.

3.3 Aplicativos com SignWriting

No artigo de [Ahmed e Seong \(2006\)](#), os autores estudam a viabilidade de um aplicativo que se torne alternativa ao tradicional SMS, escrevendo e lendo mensagens em SignWriting e texto, com enfoque para a criação de sinais. Avaliar a aceitação dessa proposta também é um dos objetivos do trabalho de Ahmed, avaliando se os Surdos usariam uma ferramenta como essa para se comunicar entre si e também com a comunidade ouvinte.

Para essa avaliação foram convidados deficientes auditivos, dos quais 9 pessoas que nasceram com a surdez aceitaram participar. Nesse estudo, os voluntários foram submetidos a um ambiente controlado de testes em que o protótipo do sistema estava implantado, onde eles precisavam manipular o sistema por duas horas, realizando alguns fluxos pré-definidos, e responder um questionário ao final da sessão de testes. O nível de aceitação dos usuários foi de 100%, sendo que 88,80% consideraram como fortemente aceitável.

Devido à semelhança na tecnologia e público alvo, esse estudo de viabilidade acaba contribuindo muito com a proposta do presente trabalho, pois possui uma boa revisão de conceitos em todas as áreas, que acabam envolvendo também a proposta atual. Além de esse estudo de viabilidade reforçar a importância dessa proposta, principalmente quanto ao interesse do público alvo no produto final, ele ajuda destacar a necessidade que os Surdos possuem de mais recursos voltados a eles em seu dia-a-dia.

Porém, por ser alternativa ao SMS, o aplicativo de Ahmed fica limitado a esse serviço que entrou em decadência nos últimos anos, justamente por causa dos aplicativos de conversa. O fato de a proposta atual se tratar de um serviço de mensagem instantânea se torna um grande diferencial.

A ferramenta desenvolvida por [Krause e Cunha \(2013\)](#) tem como objetivo a construção de um protótipo de um aplicativo que funcione como um Serviço de Mensagem Curta (SMS) ou correio eletrônico, porém baseado em Língua de Sinais, onde o usuário poderá mandar e receber mensagem com SignWriting.

Esse projeto leva em conta questões importantes da área de Interação Humano-Computador (IHC), como princípios de usabilidade, frizando em especial o desenvolvimento de sistemas interativos, com o auxílio de padrões de interface para dispositivos móveis. O sistema é construído exclusivamente para o sistema operacional Android e possui em sua composição um banco completo de signos para compor sinais da Língua Brasileira de Sinais (LIBRAS) através do SignWriting. Outro recurso utilizado é a série de padrões do Consórcio World Wide Web (W3C) para a construção de conteúdo para a Web, que leva em considerações questões de acessibilidade, qualidade e estabilidade das aplicações.

O desenvolvimento da presente proposta se relaciona ao trabalho de Krause em aspectos diversos, como o foco em dispositivos móveis e na língua de sinais utilizando SignWriting, o apelo por boas práticas e a priorização e revisão de IHC, porém acaba sendo mais abrangente levando em consideração que será um aplicativo de mensagem instantânea e não de SMS, ou seja, garante mais recursos, melhor desempenho e praticidade no uso. Isso se deve a grande queda da popularidade dos serviços de SMS, causada justamente pelos mensageiros instantâneos. Também, a possibilidade de suporte ao envio de texto garante um grupo de possíveis usuários ainda maior.

3.4 Conclusões do Capítulo

Levando em conta os trabalhos aqui analisados, a presente proposta se destaca nos dois ramos da pesquisa: na língua de sinais por possuir o auxílio de um usuário durante todo o processo de desenvolvimento, e na parte de desenvolvimento por aplicar todos os conceitos de engenharia de software que forem necessários para o pleno desenvolvimento desta aplicação. Assim, pode prover ao usuário um melhor desempenho, melhor usabilidade possibilitada por uma interface limpa e completamente intuitiva, e segurança no uso.

4 Metodologia

Neste capítulo veremos detalhes da proposta do presente trabalho, como características do produto e de que forma ele foi desenvolvido. Aqui são detalhados os métodos utilizados.

4.1 Descrição do Trabalho

Este trabalho teve como objetivo desenvolver um aplicativo de conversa que tenha suporte a SignWriting, motivado pela necessidade do Surdo de se comunicar de maneira mais ágil e de uma forma na qual está mais familiarizado (Língua de Sinais). A interface também recebe o foco durante a construção, tendo constante prototipação e validação durante o processo de desenvolvimento, com o objetivo de ser altamente intuitiva e voltada ao usuário Surdo.

Para isto, é utilizada a metodologia Scrum, que gera pacotes evolutivos por prioridade, ou seja, versões executáveis que incluem as funções mais essenciais a cada novo lançamento. Mais detalhes sobre a metodologia são dados a seguir.

4.2 Metodologia Scrum

Scrum é uma metodologia ágil de desenvolvimento de software, caracterizada por ser iterativa, ou seja, o tempo é dividido em ciclos (geralmente mensais), onde um conjunto de atividades precisa ser realizado. No Scrum, temos o *Product Backlog* que determina tudo o que o sistema precisa ter quanto a funcionalidades. Ali, estas atividades vão ser priorizadas e divididas entre os *Sprints* (Iterações).

Pensando sequencialmente, no início de cada mês é realizada uma reunião para definir as atividades do *Sprint*, de acordo com o quanto a equipe será capaz de realizar neste período e com a prioridade do que precisa ser feito. Então, estas atividades são divididas entre o time de desenvolvimento e o seu andamento é checado diariamente em um *Daily Meeting*, para relatarmos o andamento do projeto, dificuldades e planejamentos. A figura 11 dá uma melhor perspectiva do processo.

Este processo de desenvolvimento é constituído de três papéis distintos: O *time de desenvolvimento*, composto de 3 a 9 desenvolvedores; O *Scrum Master*, responsável por garantir o funcionamento do processo, tirando dúvidas e auxiliando o time em qualquer impedimento; *Product Owner*, que gerencia a construção do produto de acordo com os interesses do cliente (SCHWABER, 2004).

Como a nossa realidade para a execução deste projeto é diferente da configuração ideal de uma equipe Scrum, iremos adaptar o processo à relação de orientação: As iterações se darão de 30 a 45 dias, e o Daily Meeting se tornará uma reunião semanal. Com relação aos papéis, caberá ao orientador o papel de *Scrum Master* enquanto ao aluno cabe os papéis de *Product Owner* e do *time de desenvolvimento*.

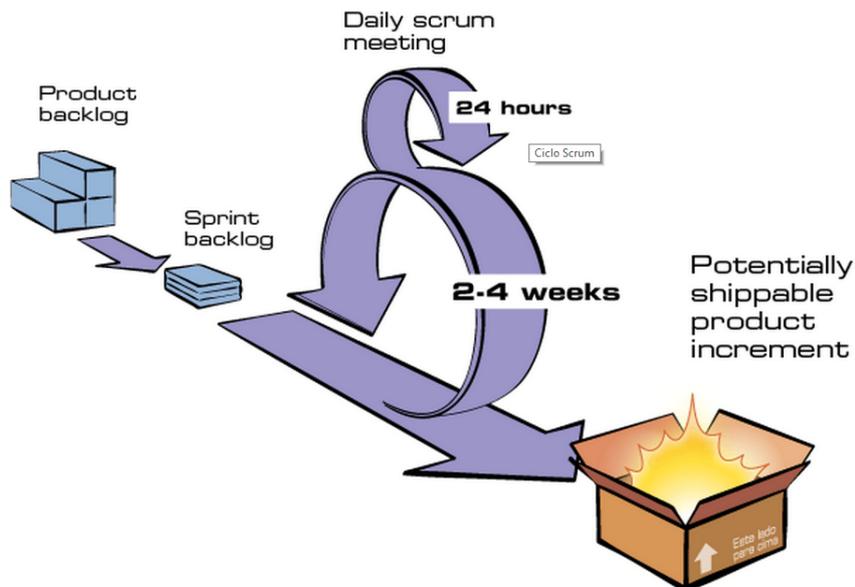


Figura 11 – Processo de desenvolvimento Scrum. (Fonte: desenvolvimentoagil.com)

4.2.1 Product Backlog

Na metodologia Scrum, o *Product Backlog* contém todas as funcionalidades que o sistema irá ter. Ele dá origem ao *Sprint Backlog*, gerando assim os ciclos de iteração. Cada funcionalidade nesta lista recebe uma prioridade, e a partir desta informação o *Product Backlog* será separado nesses ciclos de iteração. Esse é um documento volátil, pois como o Scrum também é recomendado para o desenvolvimento de projetos com o escopo não tão bem definido, ele costuma sofrer alterações a cada Sprint devido a novas informações que surgem do cliente, repriorizando todos os seus itens em decorrência disto. O *backlog* inicial do projeto é apresentado na figura 12.

Para o desenvolvimento de cada item do *backlog*, são utilizados os artefatos de desenvolvimento detalhados abaixo.

4.2.2 Especificação dos Casos de Uso

Os casos de uso são especificados através do documento expandido, como o detalhado a seguir, na figura 13. Ele contém os passos que o usuário deve seguir para executar a ação, o que já concede uma boa base para o início da fase de projeto.

ID	Funcionalidade	Descrição	Tipo	Sprint
01	Enviar sinais	Construção e envio da mensagem escrita totalmente em SignWriting.	Cliente	2
02	Enviar texto	Envio de texto normal para dar suporte aos testes iniciais de comunicação do aplicativo.	Cliente	1
03	Gerenciar contatos	O usuário poderá adicionar novos contatos através do nome de usuário ou email, também poderá excluir contatos indesejados.	Cliente	2
04	Enviar fotos	O Sistema permitirá que o usuário envie fotos para um contato ou grupo.	Cliente	2
05	Personalizar perfil	O usuário poderá personalizar o seu perfil, compartilhando assim as informações pessoais que quiser com seus contatos.	Cliente	3
06	Manter grupos	O servidor deverá conter os grupos cadastrados e gerenciar a conversação entre os usuários presentes neles.	Servidor	3
07	Manter cadastros	O servidor deverá receber todos os novos cadastros no sistema e armazená-los de maneira em que o usuário possa autenticar-se no sistema.	Servidor	1
08	Encaminhar mensagem	O servidor deverá encaminhar a mensagem enviada pelo remetente ao seu destinatário (Usuário ou Grupo).	Servidor	1
09	Gerenciar grupos	O usuário poderá gerenciar seus grupos de conversa com amigos, adicionando mais amigos à conversa, também deixando de participar de grupos indesejados.	Cliente	3
10	Realizar Login	O usuário deverá autenticar-se no sistema.	Cliente	1
11	Realizar Cadastro	O usuário poderá realizar o cadastro no sistema.	Cliente	1

Figura 12 – Product Backlog inicial

4.2.3 Projeto

O projeto de software é a descrição da estrutura que vai ser implementada, podendo representar vários níveis da aplicação, dando visões diferentes ao desenvolvedor (SOMMERVILLE, 2007). Para a concepção da solução, foram analisadas os diagramas disponibilizadas pela linguagem UML (Linguagem de Modelagem Unificada), descritas por Booch, Rumbaugh e Jacobson (2006), para saber quais colaboram mais para especificar a solução para o sistema proposto. Como resultado desse estudo sobre UML, são utilizados os seguintes diagramas:

- Diagrama de classes: O diagrama de classes reflete a estrutura estática do software, especificando as classes que servirão de base para os objetos do sistema e seus atributos, métodos e relacionamentos. Ela serve de base para o diagrama de sequência, descrito a seguir. O diagrama de classes da aplicação encontra-se na figura 14, onde o usuário tem uma lista de contatos, e com cada contato ele possui uma lista de mensagens;

ID:	1	Caso de Uso:	Enviar Mensagem em SignWriting
Ator Principal	Usuário	Atores Auxiliares	
Pré-condição	Realizar Caso de Uso 10		
Pós-condição	Fluxo segue para o Caso de Uso 8.		
Fluxo Básico			
1. Usuário seleciona um contato para o qual quer enviar a mensagem.			
2. Sistema exibe a tela da conversa com o respectivo contato.			
3. Usuário clica no botão de criar nova mensagem.			
4. Sistema exibe a tela de construção de mensagem.			
5. Usuário cria a mensagem arrastando os elementos do menu de símbolos.			
6. Usuário clica no botão de enviar mensagem.			
7. Sistema envia a mensagem.			
8. Sistema redireciona para a tela de conversa, já exibindo a nova mensagem no fluxo.			
Fluxo Alternativo			
Fluxo de Exceção			
3.1 - Usuário não está conectado à internet			
3.1.1 - Sistema exibe mensagem de erro com o texto "Você não está conectado à internet, conecte-se e tente novamente!"			
Requisito Não Funcional			
1. Sistema deverá informar ao usuário o feedback das operações realizadas.			

Figura 13 – Caso de Uso 1 - Enviar mensagem em SignWriting

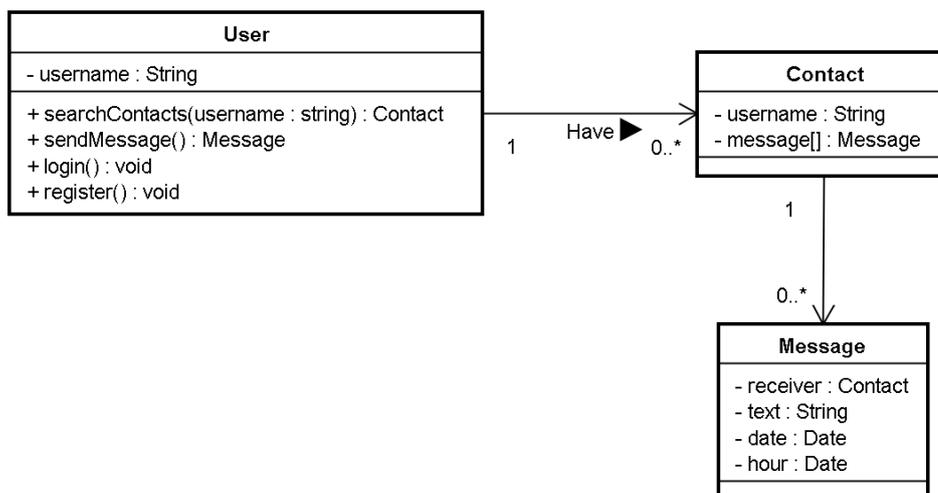


Figura 14 – Diagrama de classes

- Diagrama de sequência: Este diagrama reflete a sequência de mensagens trocadas pelos objetos do sistema para a conclusão de determinada tarefa. Neste projeto, ele é importante para especificar a comunicação, desde a construção da mensagem até como o sistema reage ao receber uma nova. Os diagramas de sequência encontram-se na seção de apêndices do trabalho.
- Diagrama de Implantação: Utilizado para especificar os componentes de hardware e software, e a forma como interagem com os demais elementos. Apresenta a infraestrutura necessária, além dos elementos de software presentes em cada nó. Essa

representação foi escolhida por dar uma boa perspectiva da organização de software distribuídos. O diagrama de implantação da aplicação encontra-se na figura 15, onde o destaque vai para a representação dos dois serviços utilizados, um para mensagens instantâneas e outro para a persistência de dados (Sinch e Parse, respectivamente).

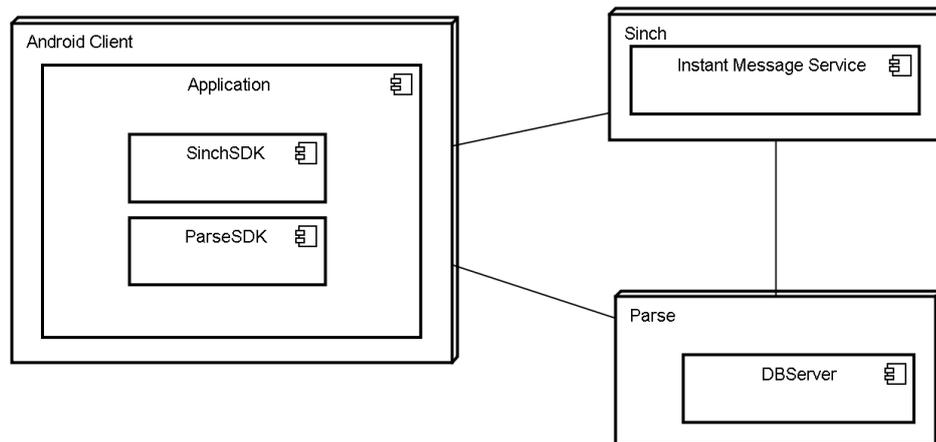


Figura 15 – Diagrama de Implantação

Além dos elementos UML, foi utilizado o modelo entidade-relacionamento para definir a forma de persistência dos dados. O modelo Entidade-Relacionamento representa a estrutura dos dados que são persistidos no servidor da aplicação. A figura 16 representa o diagrama ER projetado.

4.2.4 Qualidade do Software

Além da validação do usuário para analisar se o software atende às suas expectativas, o software precisa apresentar garantias de que ele não apresente erros em situações adversas de uso, tornando a experiência do usuário desagradável.

Para garantir que o software seja robusto, tratando o maior número possível de situações adversas causadas tanto por usuários quanto por fatores computacionais, a especificação de casos de teste é utilizada. Criada a partir dos casos de uso expandidos, ele descreve uma condição peculiar a ser testada. É composto por valores de entrada, limitações para a sua execução, e por fim um resultado ou comportamento resultante daquela operação.

Para a construção dos casos de teste, [Bartié \(2002\)](#) descreve a técnica de variação textual e visual, onde cada fluxo do caso de uso deve ser mapeado e se transforma em um requisito de teste. Após isso, de cada caminho são identificadas todas as possibilidades de variação. Cada variação de cada fluxo constitui um cenário, e é em cima desses cenários resultantes dessa variação que os casos de teste são especificados.

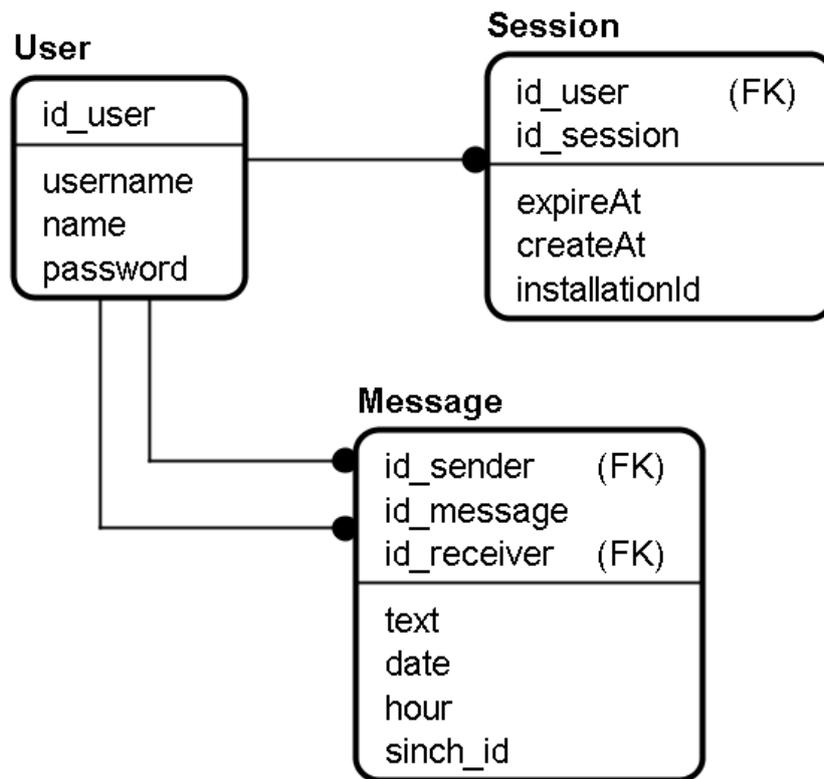


Figura 16 – Diagrama Entidade-Relacionamento

Para a aplicação, foi utilizado o seguinte modelo de caso de teste representado pela figura 17, que representa os especificados e executados para o Caso de Uso 1.

ID Caso de Uso : 01			
CT	Valores de entrada:	Valores de Saída esperados;	Resultado:
1	Usuário cria um sinal na caixa de desenho e clica em "enviar"	O sistema deve exibir o status da mensagem e notificar se foi enviada. O sistema deve exibir a mensagem na tela de conversa.	Passou
2	Usuário clica para no botão de "nova mensagem", a partir da tela de conversa, e após cancela a ação, clicando em voltar.	Sistema deve exibir a tela de conversa.	Passou
3	Usuário cria uma mensagem na caixa de desenho e clica em "enviar", porém sem estar conectada à internet.	O sistema deve exibir o alerta "Mensagem não enviada, você não está conectada à internet."	Passou

Figura 17 – Casos de teste para o UC1

4.3 Prototipação e Validações com Usuário

Para garantir que o produto gerado seja de fácil uso e tenha uma organização que facilite o entendimento e adaptação do usuário Surdo, a técnica de prototipação em papel foi escolhida. Esta técnica definida por Snyder (2003) como uma variação dos testes de usabilidade, onde usuários figurativos "executam" ações interagindo com a versão em papel da interface, que é manipulado por alguém que realiza as transições e ações do sistema no protótipo.

Esta técnica não consiste somente na simulação, ela possui sete passos sequencias bem definidos para alcançar um protótipo que será a base da interface final:

- Brainstorm do contexto da aplicação, entendendo as necessidades e características do projeto e dos usuários;
- Listar funcionalidades que precisam estar incluídas, definindo qual será o escopo do protótipo;
- Listar objetivos que o usuário deve alcançar, ou seja, quais a principais funcionalidades e como elas podem ser atingidas com o menor esforço. Pode levar em consideração a frequência com que o usuário irá realizar cada atividade;
- Desenvolver a primeira versão do protótipo, elaborando as telas na ordem que irão aparecer ao usuário. Deve levar em conta os itens definidos nos passos anteriores;
- Teste da primeira versão com usuários, simulando o fluxo de execução até alcançar os objetivos. Questionar usuário sobre os passos para realizar a ação, e quais deveriam ser os passos em suas opiniões;
- Criar segundo protótipo baseado nos resultados da avaliação. Analisar opiniões e desempenho dos usuários para realizar as tarefas, a fim de otimizar o protótipo;
- Realizar última validação com usuários, comparando os resultados e analisando se as alterações realmente surtiram efeito.

4.4 Conclusão do capítulo

Nesse capítulo foi apresentada a metodologia de desenvolvimento do sistema, assim como as estruturas e artefatos gerados nas fases iniciais dessa metodologia. No próximo capítulo será apresentada a estrutura que foi implementada no produto.

5 Sign Messenger

Neste capítulo será apresentado o produto do trabalho que foi desenvolvido, apresentando o escopo final, ferramentas e alguns resultados.

5.1 Product Backlog

No processo de desenvolvimento Scrum, o Product Backlog apresenta como característica ser um documento volátil, estando sempre sujeito a alterações proporcionadas por novos cenários. O Product Backlog na figura ?? apresenta o panorama final do desenvolvimento, com todas as alterações em relação a prioridade dos itens e com a adição dos novos, em amarelo.

Essas alterações se deram em decorrência da opinião de usuários e consenso entre os autores, nos papéis de Scrum Master e Product Owner. O surgimento de novos requisitos identificados no decorrer do projeto também influenciaram na alteração de prioridades.

A principal mudança foi a inclusão do item 13, referente ao armazenamento das mensagens na nuvem, que se deu pelo cenário do usuário perder todas as mensagens de suas conversas caso precise reinstalar a aplicação, seja por mudança do dispositivo ou por qualquer outro motivo. Igualmente, o item 12 se mostrou de grande valia na experiência de usuários de aplicações móveis, tornando-se uma agregação importante no backlog do ponto de vista do usuário. O item 14 foi implementado com o auxílio do serviço Sinch, que monitora as alterações no banco de dados enquanto a aplicação está ativa, mantendo assim o usuário sempre atualizado.

O item 2, referente ao envio de mensagens de texto, foi implementado mas removido da aplicação ao final do desenvolvimento por não ser um dos focos do trabalho, que visa ser um incentivador e facilitador do uso da representação escrita da língua de sinais. Sua implementação foi muito útil para os testes em relação ao envio de mensagens e configurações de servidor. Porém, manter esse recurso no produto final utilizaria um espaço valioso da interface, para uma função com baixo valor ao usuário Surdo e que vai na contramão do objetivo de incentivar o uso da escrita de sinais.

5.2 Engenharia

A plataforma escolhida para o desenvolvimento foi Android, por ser a mais popular entre os dispositivos móveis. Para isso, o sistema foi desenvolvido na IDE Android Studio, sendo a mais popular e mais completa para essa plataforma. A versão mínima definida

ID	Funcionalidade	Descrição	Tipo	Prioridade	Status
2	Enviar texto	Envio de texto normal para dar suporte aos testes iniciais de comunicação do aplicativo.	Cliente	1	Completado
10	Realizar Login	O usuário deverá autenticar-se no sistema.	Cliente	1	Completado
11	Realizar Cadastro	O usuário poderá realizar o cadastro no sistema.	Cliente	1	Completado
7	Manter cadastros	O servidor deverá receber todos os novos cadastros no sistema e armazená-los de maneira em que o usuário possa autenticar-se no sistema.	Servidor	1	Completado
13	Manter mensagens	O sistema deverá manter todas as mensagens trocadas e armazená-las a fim de evitar perdas por parte do usuário.	Servidor	1	Completado
12	Manter sessão	O sistema deve armazenar o login feito por um usuário em um determinado dispositivo, evitando autenticação repetitiva.	Servidor	2	Completado
1	Enviar sinais	Construção e envio da mensagem escrita totalmente em SignWriting.	Cliente	2	Completado
8	Encaminhar mensagem	O servidor deverá encaminhar a mensagem enviada pelo remetente ao seu destinatário (Usuário ou Grupo).	Servidor	2	Pendente
3	Gerenciar contatos	O usuário poderá adicionar novos contatos através do nome de usuário ou email, também poderá excluir contatos indesejados.	Cliente	2	Pendente
4	Enviar fotos	O Sistema permitirá que o usuário envie fotos para um contato ou grupo.	Cliente	3	Pendente
5	Personalizar perfil	O usuário poderá personalizar o seu perfil, compartilhando assim as informações pessoais que quiser com seus contatos.	Cliente	3	Pendente
6	Manter grupos	O servidor deverá conter os grupos cadastrados e gerenciar a conversação entre os usuários presentes neles.	Servidor	3	Pendente
9	Gerenciar grupos	O usuário poderá gerenciar seus grupos de conversa com amigos, adicionando mais amigos à conversa, também deixando de participar de grupos indesejados.	Cliente	3	Pendente

Figura 18 – Product Backlog final

para o suporte à aplicação foi a 4.0.4, o que garante acesso a maioria dos recursos da plataforma e abrange cerca de 92% dos usuários de Android, segundo dados divulgados pelo Google no site oficial da plataforma. Para os testes e execuções durante o desenvolvimento, foi utilizado um dispositivo Samsung Galaxy Trend Lite, com versão 4.1.2 do Android.

Para a construção da estrutura do projeto foi definido o modelo MVC, conforme a figura 19. O modelo MVC (Modelo-Visualização-Controle) é favorecido pelo planejamento do Android, que já utiliza na sua essência a separação entre a camada de visualização e a de controle. A figura 19 também representa muito bem a interação entre as camadas.

5.3 Ferramentas

Para o desenvolvimento da solução, foram estudadas uma série de ferramentas para auxiliar na troca de mensagens entre os dispositivos e para gerenciar a base de dados. Entre as estudadas e que foram descartadas, destaca-se a API oficial do Skype, que permite

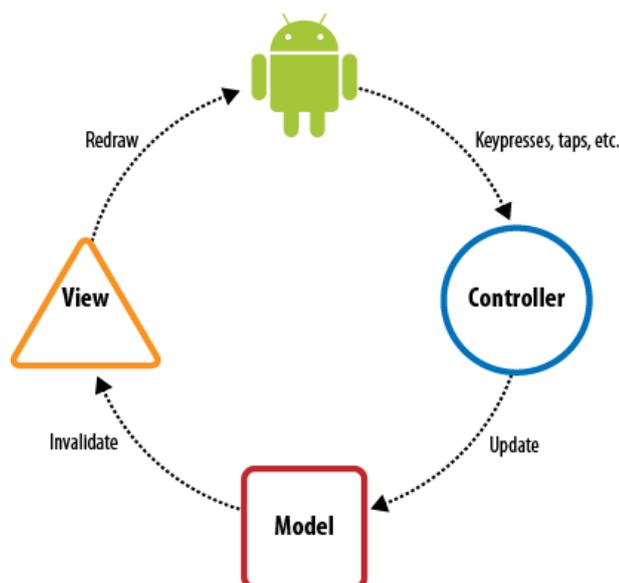


Figura 19 – Arquitetura MVC no Android. (Fonte: Programming Android, 2nd Edition by Mednieks, Dornin, Meike, and Nakamura)

apenas a troca de mensagem através dos clientes Skype instalados nos dispositivos. Com a sua utilização, a aplicação tornaria-se dependente da instalação do Skype, obrigando o usuário a tê-lo em seu dispositivo também. Essa dependência poderia trazer uma resistência ao seu uso por parte dos usuários, o que vai contra o objetivo do trabalho de construir uma aplicação de fácil uso com boa aceitação pelo público Surdo.

As ferramentas que mais se adequaram e que foram incorporadas na aplicação estão detalhadas abaixo.

5.3.1 Base64

A biblioteca Base64 consiste em um conversor de dados que são usados para transferência na Internet. Frequentemente é utilizado para transformar dados binários em texto, para que possa ser enviado pela rede. Um exemplo de utilização é para mandar arquivos anexados no email ou, no nosso caso, transmitir a escrita do sinal ao outro usuário (JOSEFSSON, 2006).

5.3.2 Parse

Parse é uma ferramenta Baas (*Backend as a Service*), que oferece uma forma de armazenar os dados na nuvem. Além disso, fornece mais uma série de serviços na nuvem, como notificações push, integração com redes sociais, e análise de uso de dados da sua aplicação.

Foi adquirida pelo Facebook em 2013 para ser usada como backend de um SDK (*Software Development Kit*) que oferece uma série de ferramentas para desenvolvedores,

permitindo assim que o desenvolvedor possa focar no frontend da aplicação, para melhorar a experiência do usuário, enquanto essa ferramenta melhora a experiência no backend (PARSE, 2015).

5.3.3 Sinch

Sinch é uma plataforma de comunicação móvel, totalmente cloud-based, que oferece recursos para adicionar comunicação por voz, autenticação, video, SMS, e mensagens instantâneas às aplicações. Ele visa remover a complexidade de construir e manter uma complexa infraestrutura de comunicação para dar suporte à aplicação. O recurso que foi utilizado para a atual aplicação foi o de mensagem instantânea, que permite utilizar mais plataformas, adiciona criptografia nas mensagens, mantém um histórico de 30 dias de mensagens enviadas, e disponibiliza notificações do tipo push. Já é utilizado por aplicações populares como Uber, Tango, Easy Taxi, Nimbuzz, entre outros (SINCH, 2015).

5.4 Interface

O processo de prototipação se concentrou na tela de criação dos sinais, pois tinha como desafio organizar a enorme quantidade de sinais, de maneira em que fosse possível garantir o melhor desempenho do usuário no processo de criação da sua mensagem.

Inicialmente, o protótipo de tela da figura 20 foi criado. Ele possui uma grande inspiração no design do WhatsApp, o aplicativo de conversa mais utilizado. Essa estratégia visa utilizar a experiência que o usuário já possui para facilitar o uso de novas aplicações, reduzindo o tempo de adaptação.

Após a criação do primeiro protótipo, a primeira avaliação com usuários foi realizada. Nela, foram obtidas as seguintes informações:

- Os usuários indicaram e concordaram com a melhor forma de agrupar os símbolos. A exibição preferida foi em 5 grandes grupos, agrupados por proximidade no uso, com 4 níveis de navegação;
- Um menu vertical seria mais interessante ao usuário, por tornar o espaço de construção do símbolo maior e mais adequado ao SignWriting, que possui sua escrita na vertical;
- Uma barra inferior com ferramentas também foi sugerida, para manipular os símbolos na área de desenho, com as funções mais necessárias como girar, espelhar, duplicar e excluir.

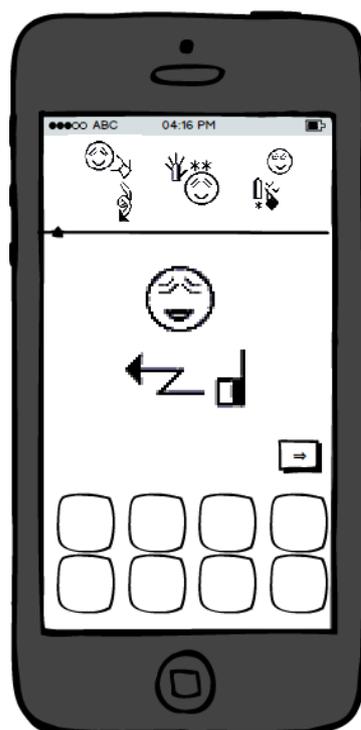


Figura 20 – Protótipo 1

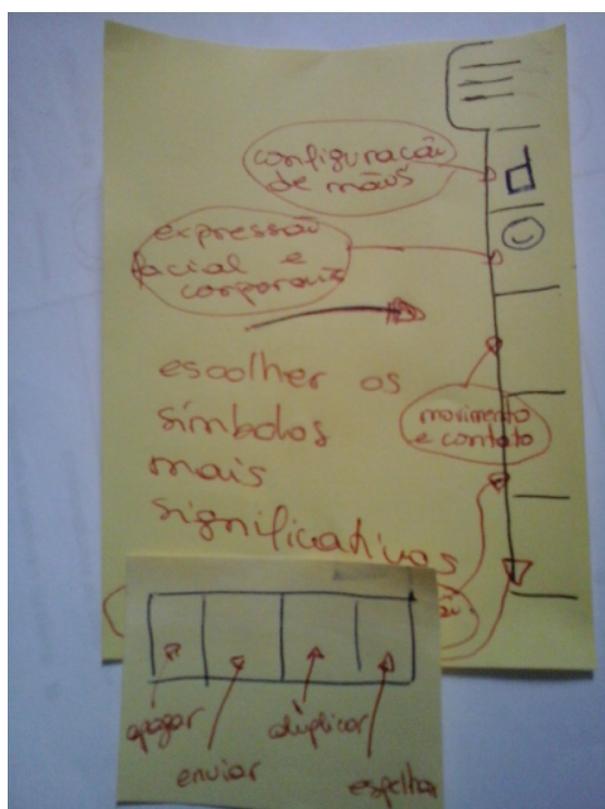


Figura 21 – Protótipo 2

As informações obtidas foram levadas em consideração, junto com princípios de usabilidade móvel, para elaborar o protótipo da figura 21. Ele contempla a organização

dos símbolos da forma como foi sugerida na avaliação, tendo um menu vertical de 4 níveis, apresentando no primeiro nível cinco grandes grupos de sinais. Do segundo ao quarto, o menu apresenta símbolos mais particulares de cada categoria, como no exemplo das configurações de mão, que no segundo nível apresenta os tipos categorizados pelos dedos apresentados na configuração. Já no terceiro nível, a partir dessa categorização pelos dedos, apresenta a forma como a própria mão se mostra, sendo com os demais dedos fechados ou um pouco mais abertos, por exemplo. Por último, no quarto nível, a orientação da mão é definida. Todo esse fluxo está representado visualmente na figura 22, onde o quadrado representa o local onde a ação (toque) é executada, e a chave o resultado da ação, com o novo estado da lista de símbolos.

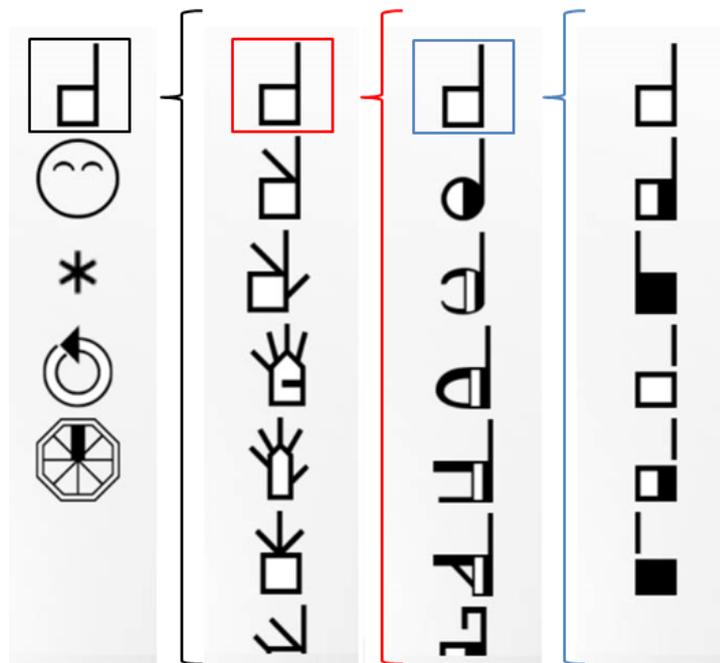


Figura 22 – Hierarquia do menu de símbolos até o indicador

O resultado final da técnica de prototipação em papel foi implementado conforme a figura 23 e a figura 24. Com elas, pode-se ter um panorama geral da execução dos casos de uso, sendo essas quatro telas suficientes para cobrir todas as funcionalidades sem tornar a exibição poluída e piorar a experiência do usuário. Esse resultado foi levado para a validação final com os usuários, que avaliaram a organização final da interface já funcional e realizaram testes de usabilidade, executando os fluxos dos casos de uso no sistema. Os resultados das execução serão discutidos no próximo capítulo.

5.5 Conclusão do capítulo

Este capítulo descreveu o software que foi gerado neste trabalho. No próximo capítulo iremos analisar o produto e os seus resultados nas perspectivas da qualidade do

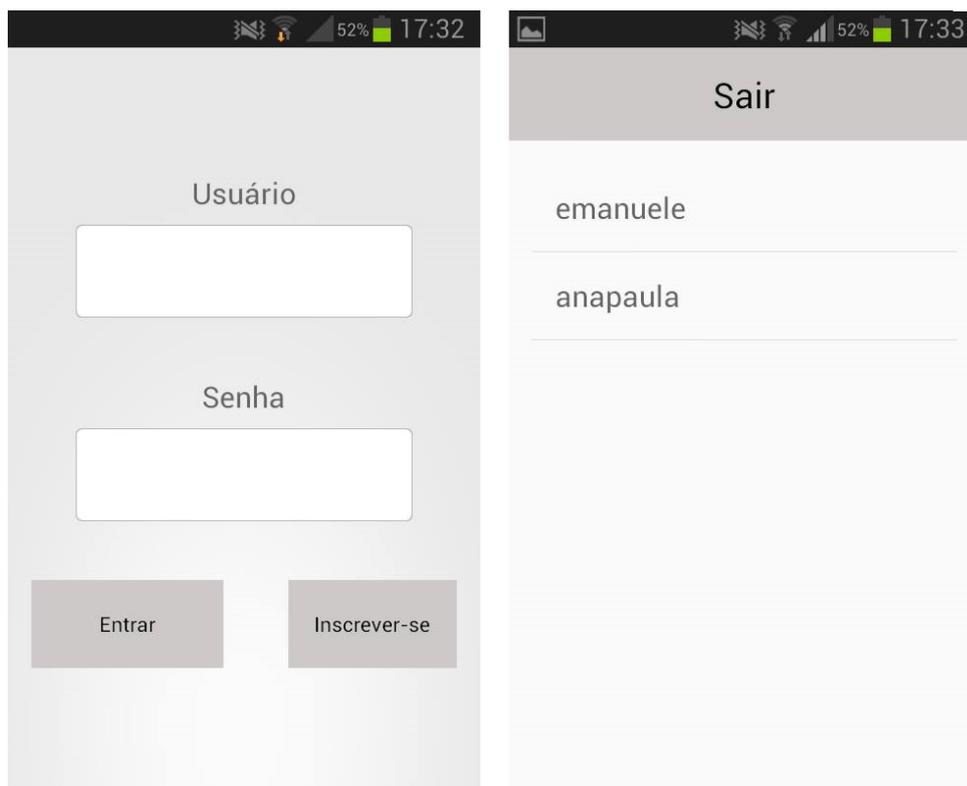


Figura 23 – Interface da aplicação - Tela de Login e de lista de usuários

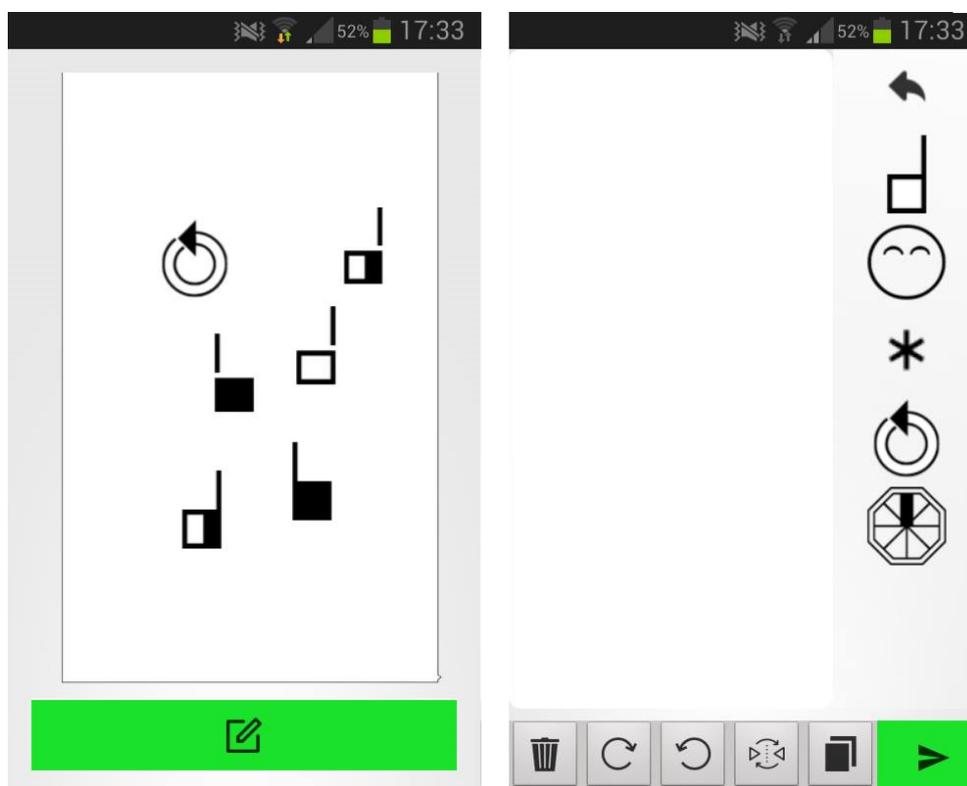


Figura 24 – Interface da aplicação - Tela de conversa e de desenho

software e dos usuários.

6 Testes e Resultados

Neste capítulo os resultados dos testes do software e da validação com os usuários são apresentados e discutidos.

6.1 Casos de teste

Para garantir a qualidade do software, foram utilizados os casos de teste elaborados para cobrir cada cenário na utilização do software. Algumas premissas definidas para a elaboração dos casos de teste são: todas as ações realizadas necessitam de um feedback do sistema; todos os dados inválidos inseridos devem ser informados ao usuário; todos os problemas na execução devido a falhas técnicas (não causadas pelo usuário) também devem ser informados precisamente.

O resultado da primeira execução dos casos de teste encontra-se no gráfico da figura 25, tendo 52,6% de aprovações. Após os erros na execução dos fluxos serem detectados na primeira bateria de testes, eles foram corrigidos e passaram para a segunda bateria de testes, onde todos os casos de teste foram novamente executados, para garantir que nenhuma alteração durante as correções tenha invalidado um caso de teste que já foi aprovado. A taxa de aprovação na última execução foi de 100,00%.

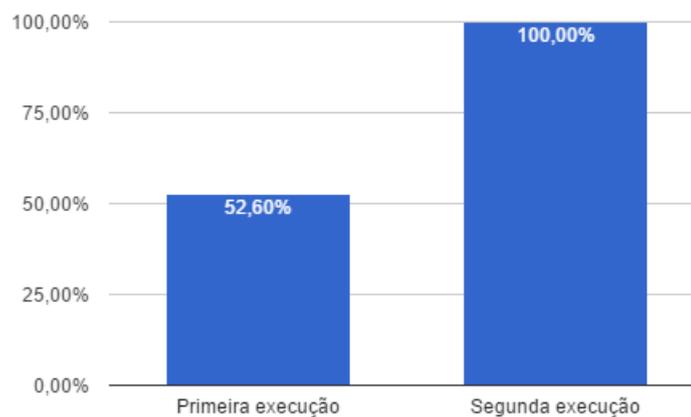


Figura 25 – Resultados das duas baterias de testes

6.2 Validação com usuários

A validação final da aplicação, já funcional e em ambiente controlado, foi realizada pelos usuários. Nela, os usuários foram induzidos a executar tarefas no sistema, como

os fluxos dos casos de uso, a fim de relatarmos se o software e as suas funcionalidades atenderam ou não as suas expectativas iniciais.

O retorno dos usuários foi extremamente positivo, sendo que em relação ao fluxo de execução dos casos de uso implementados, todas as expectativas foram alcançadas, assim como em relação aos recursos de interface e suas disposições. Também como resultado da reunião, muitas sugestões para trabalhos futuros foram levantadas, o que será apresentado no próximo capítulo.

6.3 Conclusões do capítulo

A partir dos resultados aqui obtidos pode-se concluir que o sistema está de acordo com os padrões pré-definidos para a qualidade do software e, principalmente, que está de acordo com o que os usuários exigem quanto ao uso, havendo uma grande demanda por um software com essas características entre o seu público alvo.

7 Considerações Finais

O objetivo do presente trabalho foi o desenvolvimento de um aplicativo de conversa com suporte à representação escrita da língua de sinais através de SignWriting, trazendo a usabilidade dos aplicativos populares até o Surdo, resultando na inclusão digital e na possibilidade de comunicação na sua língua materna. Objetivo esse que foi atendido com a aplicação produzida.

Como desafios e aprendizados deste trabalho, pode constar como destaque o projeto de todo o sistema na perspectiva de um usuário Surdo, garantindo sempre uma interface limpa e com informações que agreguem muito valor, utilizando poucos recursos. A interação com os usuários que puderam fazer as avaliações e dar as sugestões foram muito importantes para o desenvolvimento do trabalho e para o crescimento pessoal dos envolvidos.

Outro desafio, do ponto de vista tecnológico, foi encontrar ferramentas que auxiliassem na comunicação e possibilitassem ao sistema manter seus dados totalmente na nuvem. Para esse último, foi necessário muitos testes com ferramentas diferentes até encontrar aquela que melhor se encaixa com a necessidade da aplicação. Através desse serviço, todas as mensagens enviadas são criptografadas, garantindo a segurança no uso ao usuário.

O SignWriting está para a Língua de Sinais como o alfabeto grego está para as línguas faladas, portanto a ferramenta de comunicação bilingue construída não é restrita à LIBRAS, possibilitando a qualquer usuário Surdo a comunicação na sua língua materna, seja ela qual for.

Por fim, destacam-se alguns pontos de melhoria que podem dar continuidade ao trabalho no futuro. Essas oportunidades de melhoria seguem abaixo, como limitações do trabalho e sugestões para trabalhos futuros:

- Estender a aplicação para abranger outras plataformas, como iOS, Web ou Desktop, com o intuito de diminuir as restrições ao uso;
- Desenvolver os casos de uso não desenvolvidos que constam no product backlog, para possibilitar mais recursos que agreguem ao usuário;
- Foi prevista a utilização do XMPP, para possibilitar notificações do tipo push. Adicionar esse e outros recursos que aproximem cada vez mais a aplicação dos mensageiros mais populares, dando melhor desempenho ao sistema e melhorando a experiência do usuário;

- Sugerido pelo usuário na última validação, criar uma espécie de dicionário com alguns sinais em LIBRAS já prontos, para melhorar o desempenho na escrita e auxiliar quem é novo no sistema SignWriting;
- Apesar da validação do sistema ter considerado testes de interface, realizar esses testes com uma amostra considerável de usuários finais, para avaliar mais profundamente sua interface e funcionalidades na visão do Surdo.

Referências

- AHMED, A. S.; SEONG, D. S. K. Signwriting on mobile phones for the deaf. In: ACM. *Proceedings of the 3rd international conference on Mobile technology, applications & systems*. [S.l.], 2006. p. 28. Citado 4 vezes nas páginas 9, 11, 17 e 32.
- ALMASOUD, A. M.; AL-KHALIFA, H. S. A proposed semantic machine translation system for translating arabic text to arabic sign language. In: ACM. *Proceedings of the Second Kuwait Conference on e-Services and e-Systems*. [S.l.], 2011. p. 23. Citado na página 30.
- ALMEIDAL, E. O. C. de. *Leitura e surdez: um estudo com adultos não oralizados*. Rio de Janeiro: Revinter, 2012. Citado na página 20.
- ANTUNES, D. R. et al. A framework to support development of sign language human-computer interaction: Building tools for effective information access and inclusion of the deaf. In: IEEE. *Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on*. [S.l.], 2011. p. 1–12. Citado na página 17.
- BARTIÉ, A. *Garantia da qualidade de software: adquirindo maturidade organizacional*. [S.l.]: Elsevier, 2002. Citado na página 39.
- BIANCHINI, C. S.; BORGIA, F.; MARSICO, M. D. Swift-a signwriting editor to bridge between deaf world and e-learning. In: IEEE. *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*. [S.l.], 2012. p. 526–530. Citado na página 29.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Elsevier Brasil, 2006. Citado na página 37.
- BOUTORA, L. *Étude des systèmes d'écriture des langues vocales et des langues signées*. Tese (Doutorado) — Université Paris, Paris, 2003. Citado na página 21.
- BRASIL. Lei nº 10.436, de 24 de abril de 2002. dispõe sobre a língua brasileira de sinais - libras e dá outras providências. 2002. Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/2002/L10436.htm>. Citado na página 19.
- BRASIL. Decreto nº 5.626, de 22 de dezembro de 2005. regulamenta a lei no 10.436, de 24 de abril de 2002, que dispõe sobre a língua brasileira de sinais - libras, e o art. 18 da lei no 10.098, de 19 de dezembro de 2000. 2005. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2005/Decreto/D5626.htm#art1>. Citado na página 19.
- BUENO, A. V. B. Utilização de aplicativos messenger em resoluções de conflitos. *XXXVII Congresso Brasileiro de Ciências da Comunicação*, 2014. Citado na página 24.
- CAMPOS, M. de B. *Ambiente Telemático de Interação e Comunicação para Suporte à Educação Bilíngue de Surdos*. Tese (Doutorado) — Centro Interdisciplinar de Novas Tecnologias na Educação, UFRGS, Porto Alegre, 2001. Disponível em: <<http://seer.ufrgs.br/InfEducTeoriaPratica/article/view/5278>>. Acesso em: 6.21.2015. Citado na página 20.

- CHANG, F.-C.; CHEN, D.-K. The design of an xmpp-based service integration scheme. In: IEEE. *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*. [S.l.], 2011. p. 33–36. Citado 3 vezes nas páginas 13, 26 e 31.
- CHOI, J. H.; LEE, H.-J. Facets of simplicity for the smartphone interface: A structural model. *International Journal of Human-Computer Studies*, Elsevier, v. 70, n. 2, p. 129–142, 2012. Citado na página 31.
- ERICSSON. Mobility report. 2014. Disponível em: <<http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf>>. Citado 2 vezes nas páginas 13 e 25.
- GESSER, A. *LIBRAS? : Que língua é essa? : crenças e preconceitos em torno da língua de sinais e da realidade surda*. São Paulo: Parábola Editorial, 2009. Citado na página 20.
- GUIMARÃES, C.; GUARDEZI, J. F.; FERNANDES, S. Sign language writing acquisition–technology for a writing system. In: IEEE. *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. [S.l.], 2014. p. 120–129. Citado na página 29.
- HESSEL FABIANO ROSA, L. K. C. *Cinderela surda*. Canoas: Ed. ULBRA, 2007. Citado 3 vezes nas páginas 13, 23 e 24.
- JOSEFSSON, S. The base16, base32, and base64 data encodings. 2006. Citado na página 45.
- KRAUSE, F. J. S.; CUNHA, I. S. D. Signsms-sistema para escrita de mensagens em língua de sinais. *Revista da Graduação*, v. 6, n. 1, 2013. Citado na página 32.
- MATSUMOTO, T.; KATO, M.; IKEDA, T. Jspad: a sign language writing tool using signwriting. In: ACM. *Proceedings of the 3rd International Universal Communication Symposium*. [S.l.], 2009. p. 363–367. Citado na página 30.
- NIELSEN, R. B. J. *Usabilidade Móvel*. Rio de Janeiro: Elsevier, 2014. Citado na página 27.
- PARSE. Parse. 2015. Disponível em: <<http://www.parse.com/>>. Acesso em: 11.19.2015. Citado na página 46.
- REIS, B. S. S. d. Você tem whatsapp?: um estudo sobre a apropriação do aplicativo de celular por jovens universitários de Brasília. 2014. Citado na página 25.
- RODRIGUES, M. L. A. *Contos de fadas em libras*. 2011. Citado 2 vezes nas páginas 13 e 21.
- SCHWABER, K. *Agile project management with Scrum*. [S.l.]: Microsoft Press, 2004. Citado na página 35.
- SINCH. Sinch. 2015. Disponível em: <<http://www.sinch.com/>>. Acesso em: 11.19.2015. Citado na página 46.
- SNYDER, C. *Paper prototyping: The fast and easy way to design and refine user interfaces*. [S.l.]: Morgan Kaufmann, 2003. Citado na página 40.

- SOMMERVILLE, I. *Engenharia de software. Tradução: Selma Shin Shimuzu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa*. [S.l.]: São Paulo: Pearson Addison Wesley, 2007. Citado na página 37.
- STROBEL, K. *As imagens do outro sobre a cultura surda*. Florianópolis: Editora UFSC, 2013. Citado 2 vezes nas páginas 19 e 23.
- STUMPF, M. R. *Aprendizagem de Escrita de Língua de Sinais pelo Sistema SignWriting: Línguas de Sinais no Papel e no Computador*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005. Citado 3 vezes nas páginas 13, 22 e 23.
- SUTTON, V. Signwriting. 2015. Disponível em: <<http://www.signwriting.org/>>. Acesso em: 6.17.2015. Citado 2 vezes nas páginas 17 e 21.

Apêndices

APÊNDICE A – CASOS DE USO EXPANDIDOS

ID:	2	Caso de Uso:	Enviar Mensagem de Texto
Ator Principal	Usuário	Atores Auxiliares	
Pré-condição	Realizar Caso de Uso 10		
Pós-condição	Fluxo segue para o Caso de Uso 8.		

Fluxo Básico

1. Usuário seleciona um contato para o qual quer enviar a mensagem.
2. Usuário escreve a mensagem na caixa de texto
3. Usuário toca em "Enviar".

Fluxo Alternativo

Fluxo de Exceção

- 3.1 - Usuário não está conectado à internet
- 3.1.1 - Sistema exibe mensagem de erro com o texto "Você não está conectado à internet, conecte-se e tente novamente!"

Requisito Nao Funcional

1. Sistema deverá informar ao usuário o feedback do envio da mensagem.

ID:	13	Caso de Uso:	Manter Mensagens
Ator Principal	Sistema	Atores Auxiliares	Usuário
Pré-condição	Realizar Caso de Uso 2 ou Caso de Uso 1		
Pós-condição			

Fluxo Básico

1. O sistema recebe a mensagem do usuário.
2. O sistema armazena a mensagem com o identificador do remetente e do destinatário.

Fluxo Alternativo

Fluxo de Exceção

- 2.1. a) Sistema não encontra o usuário na base.

Requisito Nao Funcional

1. Sistema deverá informar ao usuário o feedback das operações realizadas

ID:	1	Caso de Uso:	Enviar Mensagem em SignWriting
Ator Principal	Usuário	Atores Auxiliares	
Pré-condição	Realizar Caso de Uso 10		
Pós-condição	Fluxo segue para o Caso de Uso 8.		

Fluxo Básico

1. Usuário seleciona um contato para o qual quer enviar a mensagem.
2. Sistema exibe a tela da conversa com o respectivo contato.
3. Usuário clica no botão de criar nova mensagem.
4. Sistema exibe a tela de construção de mensagem.
5. Usuário cria a mensagem arrastando os elementos do menu de símbolos.
6. Usuário clica no botão de enviar mensagem.
7. Sistema envia a mensagem.
8. Sistema redireciona para a tela de conversa, já exibindo a nova mensagem no fluxo.

Fluxo Alternativo

Fluxo de Exceção

3.1 - Usuário não está conectado à internet

3.1.1 - Sistema exibe mensagem de erro com o texto "Você não está conectado à internet, conecte-se e tente novamente!"

Requisito Não Funcional

1. Sistema deverá informar ao usuário o feedback das operações realizadas.

ID:	10	Caso de Uso:	Realizar Login
Ator Principal	Usuário	Atores Auxiliares	
Pré-condição			
Pós-condição			

Fluxo Básico

1. Usuário abre a tela inicial.

2. Usuário preenche os campos usuário e senha.

3. Usuário toca em "Entrar".

4. Sistema redireciona para a tela de contatos.

Fluxo Alternativo

Fluxo de Exceção

3.1 - Usuário não está conectado à internet

3.1.1 - Sistema exibe mensagem de erro com o texto "Você não está conectado à internet, conecte-se e tente novamente!"

3.2 - Usuário não está cadastrado

3.2.1 - Sistema exibe mensagem de erro com o texto "Usuário não cadastrado".

3.3 - Senha incorreta

3.3.1 - Sistema exibe mensagem de erro com o texto "Usuário e senha não conferem."

Requisito Não Funcional

1. Sistema deverá informar ao usuário o feedback da ação.

ID:	11	Caso de Uso:	Realizar Cadastro
Ator Principal	Usuário	Atores Auxiliares	Sistema
Pré-condição			
Pós-condição			

Fluxo Básico

1. Usuário abre a tela inicial.

2. Usuário preenche os campos usuário e senha.

3. Usuário toca em "Registrar".

4. Sistema redireciona para a tela de contatos.

Fluxo Alternativo

Fluxo de Exceção

3.1 Usuário não está conectado à internet

3.1.1 - Sistema exibe mensagem de erro com o texto "Você não está conectado à internet, conecte-se e tente novamente!"

3.2 Usuário não preencheu todos os campos.

3.2.1 - Sistema exibe mensagem de erro com o texto "Preencha todos os campos antes de realizar o cadastro."

3.3 Usuário já existente.

3.3.1 - Sistema exibe mensagem de erro com o texto "O nome de usuário já existe. Insira um nome diferente."

Requisito Não Funcional

1. Sistema deverá informar ao usuário o feedback da ação.

ID:	7	Caso de Uso:	Manter Cadastros
Ator Principal	Sistema	Atores Auxiliares	Usuário
Pré-condição	Realizar Caso de Uso 10 ou Caso de Uso 11		
Pós-condição	Realizar Caso de Uso 12		

Fluxo Básico

1. O sistema recebe a um novo cadastro.

2. O sistema armazena o cadastro.

Fluxo Alternativo

1. O sistema recebe um novo login.

1.1 - O sistema retorna o resultado.

1. O sistema recebe uma solicitação da lista de usuários.

1.1 - O sistema a lista com todos os usuários cadastrados.

Fluxo de Exceção

Requisito Nao Funcional

1. Sistema deverá informar ao usuário o feedback das operações realizadas

ID:	12	Caso de Uso:	Manter Sessão
Ator Principal	Sistema	Atores Auxiliares	Usuário
Pré-condição	Realizar Caso de Uso 10 ou Caso de Uso 11		
Pós-condição			

Fluxo Básico

1. O sistema recebe um novo cadastro ou login.

2. O sistema armazena a sessão, com o identificador da instalação da aplicação e o identificador do usuário.

Fluxo Alternativo

Fluxo de Exceção

Requisito Nao Funcional

APÊNDICE B – CASOS DE TESTE

ID Caso de Uso : 01			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	Usuário cria um sinal na caixa de desenho e clica em "enviar"	O sistema deve exibir o status da mensagem e notificar se foi enviada. O sistema deve exibir a mensagem na tela de conversa.	Passou
2	Usuário clica para no botão de "nova mensagem", a partir da tela de conversa, e após cancela a ação, clicando em voltar.	Sistema deve exibir a tela de conversa.	Passou
3	Usuário cria uma mensagem na caixa de desenho e clica em "enviar", porém sem estar conectado à internet.	O sistema deve exibir o alerta "Mensagem não enviada, você não está conectado à internet."	Passou
ID Caso de Uso : 02			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	Usuário deixa a caixa de mensagem em branco e clica em "enviar"	O sistema deve exibir o alerta "Digite algo para enviar!" e exibir a mensagem na tela de conversa	Passou
2	Usuário escreve uma mensagem na caixa de mensagem e clica em "enviar"	O sistema deve exibir a notificação de "Mensagem enviada!" e a mensagem deve aparecer no histórico da conversa em seguida.	Passou
3	Usuário escreve uma mensagem na caixa de mensagem e clica em "enviar", porém sem estar conectado à internet.	O sistema deve exibir o alerta "Mensagem não enviada, você não está conectado à internet."	Passou
ID Caso de Uso : 07			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	Usuário efetua o login corretamente.	O sistema deve exibir a tela de contatos com todos os usuários cadastrados.	Passou
ID Caso de Uso : 10			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	Usuário preenche campo usuário com um cadastro inexistente e clica em "entrar"	O sistema deve exibir o alerta de "Usuário não cadastrado"	Passou
2	Usuário preenche campo usuário com valor existente, porém o campo senha com um valor incorreto, e clica em "entrar".	O sistema deve exibir o alerta de "Usuário e senha não conferem"	Passou
3	Usuário preenche campos usuário e senha com valores existentes, e clica em "entrar".	O sistema deve exibir a tela de lista de usuários.	Passou
4	Usuário preenche campos e clica em "entrar", porém sem estar conectado à internet.	O sistema deve exibir o alerta "Você não está conectado à internet. Conecte-se e tente novamente!"	Passou
5	Usuário clica em "entrar" sem preencher os campos.	O sistema deve exibir o alerta "Preencha os campos com valores válidos"	Passou
ID Caso de Uso : 11			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	Usuário preenche campo usuário com um cadastro existente e clica em "registrar"	O sistema deve exibir o alerta de "Usuário já cadastrado"	Passou
2	Usuário preenche campos usuário e senha com valores válidos, e clica em "registrar".	O sistema deve exibir a tela de lista de usuários.	Passou
3	Usuário preenche campos e clica em "registrar", porém sem estar conectado à internet.	O sistema deve exibir o alerta "Você não está conectado à internet. Conecte-se e tente novamente!"	Passou
4	Usuário clica em "registrar" sem preencher os campos.	O sistema deve exibir o alerta "Preencha os campos com valores válidos"	Passou
ID Caso de Uso : 12			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	1 Usuário efetua o login corretamente. 2 Usuário fecha a aplicação. 3 Usuário reabre a aplicação.	O sistema deve exibir a tela de contatos com todos os usuários cadastrados.	Passou
ID Caso de Uso : 13			
<i>CT</i>	<i>Valores de entrada:</i>	<i>Valores de Saída esperados;</i>	<i>Resultado:</i>
1	Usuário envia duas mensagens em sequência a um contato.	O sistema deve exibir as mensagens ordenadas por data, sendo a mais recente mais abaixo.	Passou
2	Usuário seleciona um contato na tela de usuários.	O sistema deve exibir a tela de conversa com todas as mensagens trocadas por eles nos últimos 30 dias.	Passou

APÊNDICE C – PROJETO

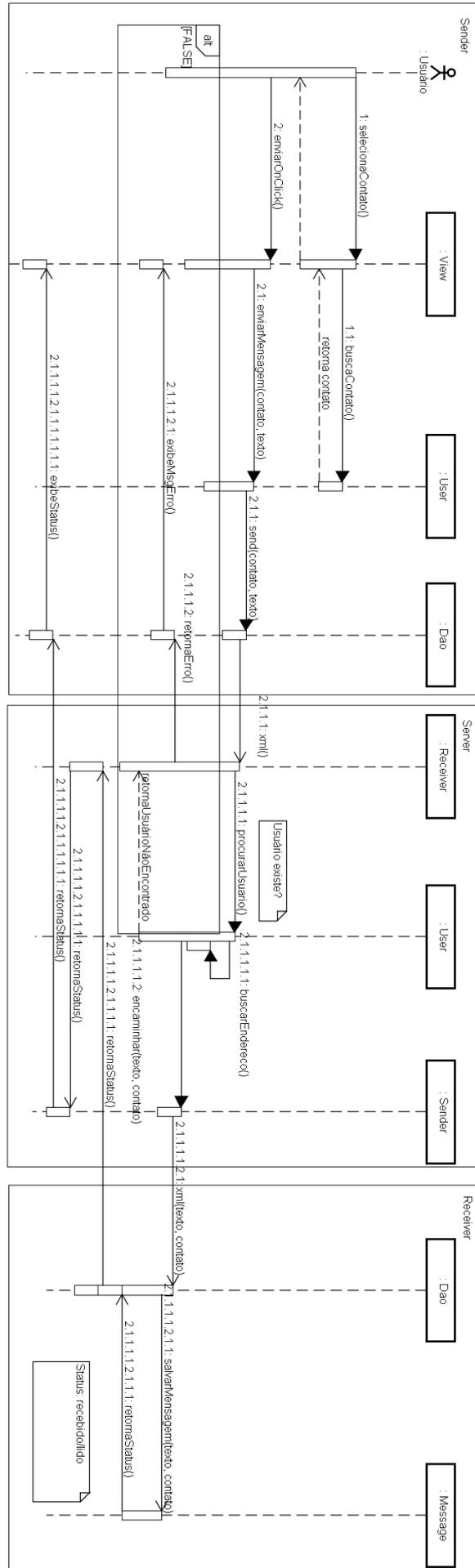


Figura 26 – Diagrama de seqüência