

# *Revista da Graduação*

---

Vol. 6

No. 1

2013

15

---

**Seção:** Faculdade de Informática

**Título:** SIGNSMS: SISTEMA PARA ESCRITA DE MENSAGENS EM  
LÍNGUA DE SINAIS

Autor: FELIPE JONATHAN SILVA KRAUSE E IVAN SANTOS DA CUNHA

Este trabalho está publicado na Revista da Graduação.

ISSN 1983-1374

<http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/13996>

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FELIPE JONATHAN SILVA KRAUSE  
IVAN SANTOS DA CUNHA

***SignSMS***  
**Sistema para escrita de mensagens em Língua de Sinais**

Trabalho de Conclusão de Curso II  
Orientadora: Profa. Dr. Márcia de Borba Campos

Porto Alegre  
2012

FELIPE JONATHAN SILVA KRAUSE  
IVAN SANTOS DA CUNHA

***SignSMS***  
**Sistema para escrita de mensagens em Língua de Sinais**

Trabalho de Conclusão de Curso apresentada  
como requisito para a obtenção do grau de  
Bacharel em Ciência da Computação pela  
Faculdade de Informática da Pontifícia  
Universidade Católica do Rio Grande do Sul

Orientadora: Profa. Dr. Márcia de Borba Campos

Porto Alegre  
2012

FELIPE JONATHAN SILVA KRAUSE  
IVAN SANTOS DA CUNHA

***SignSMS***  
**Sistema para escrita de mensagens em Língua de Sinais**

Trabalho de Conclusão de Curso apresentada  
como requisito para a obtenção do grau de  
Bacharel em Ciência da Computação pela  
Faculdade de Informática da Pontifícia  
Universidade Católica do Rio Grande do Sul

Aprovada em: \_\_\_\_ de dezembro de 2012.

BANCA EXAMINADORA:

---

Profa. Dr. Beatriz Regina T. Franciosi – PUCRS

---

Profa. Dr. Márcia Cristina Moraes – PUCRS

---

Profa. Dr. Márcia de Borba Campos - PUCRS

Porto Alegre  
2012

## **AGRADECIMENTOS**

Acima de tudo aos nossos familiares, que sempre nos apoiaram independentemente de quão difícil o momento pudesse ser, que souberam entender nossa ausência durante estas épocas especiais e nunca desistiram dessa vitória...

À professora Márcia de Borba Campos pela paciência na orientação, pelo incentivo e por seu apoio e inspiração no amadurecimento dos nossos conhecimentos e conceitos que nos levaram a execução e conclusão deste trabalho.

Ao amigo e colega de trabalho Ricardo Rivaldo pela ajuda nos conceitos da tecnologia *Android* que foram fundamentais para o desenvolvimento da aplicação proposta.

A todos os professores da Faculdade de Informática da PUCRS, em especial, Alfio Martini, Beatriz Franciosi, João Batista, Márcia Cristina Moraes e Ana Benso, que foram tão importantes em nossas vidas acadêmicas.

Aos amigos e colegas pelo incentivo e pelo apoio constantes.

Em especial aos meus pais, irmão, minha noiva Marleide Dutra, que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Felipe Jonathan Silva Krause

Quero agradecer aos meus pais e também aos meus irmãos, pelo apoio para lidar com esses obstáculos do nosso cotidiano. Também gostaria de agradecer a todos os amigos e colegas que se fizeram presente durante esse período.

Ivan Santos da Cunha

## RESUMO

Este trabalho descreve uma solução para o envio e recebimento de mensagens escritas em língua de sinais para tecnologia móvel. Mais especificadamente, é utilizado o sistema *SignWriting* para escrever os sinais e a plataforma de desenvolvimento *Android* para rodar a aplicação.

Para que tal objetivo pudesse ser alcançado, foi realizada uma revisão bibliográfica sobre língua de sinais, sistema de escrita de sinais, interação humano-computador, padrões e princípios de *design* que serviram de suporte às discussões e ao levantamento de requisitos para o sistema aqui apresentado. Também foram estudados aplicativos móveis para identificar e analisar o uso da língua de sinais, em suas diferentes modalidades. Esses estudos também são apresentados nesse trabalho.

A partir do referencial teórico e análise de sistemas correlacionados, foi elaborado o espaço do problema e definido os requisitos para o *SignSMS*, que visa permitir a comunicação assíncrona, por meio de telefones móveis, a partir do sistema de escrita de sinais *SignWriting*.

Por fim, não menos importante, ressalta-se o potencial do aplicativo ora apresentado uma vez que a escrita da língua de sinais tem sido incluída nas escolas brasileiras e nas universidades. Assim, o *SignSMS* se configura como uma ferramenta que, ao permitir a troca de informações utilizando o *SignWriting*, permite que as pessoas possam se comunicar na sua língua materna (no caso das pessoas surdas) e possam (pessoas surdas e ouvintes) melhor aprender esse processo de escrita de língua de sinais.

**Palavras-chave:** Escrita de sinais. *SignWriting*. Mobilidade. *Android*. Acessibilidade. *SignSMS*.

## **ABSTRACT**

This final paper describes a solution for sending and receiving text messages in sign language for mobile technology. More specifically, the system uses the SignWriting system and Android development platform to run the application.

For that this objective could be achieved, we performed a literature review on sign language, sign writing system, human-computer interaction, design principles and patterns that supported the discussions and the requirements elaboration for the system presented here. Mobile applications were also studied to identify and analyze the use of sign language in its different modalities. These studies are also presented in this work.

From the theoretical referential and analysis of co-related systems, was elaborated the problem space and was defined the requirements for SignSMS, which aims to enable asynchronous communication through mobile phones, from the system of sign writing SignWriting.

Finally, but not least, we point out the potential application presented here since the use of sign writing language has been included in Brazilian schools and universities. Thus, the SignSMS is configured as a tool that, by allowing the exchange of information via SignWriting, allows people to communicate in their mother language (in the case of deaf people) and allows (deaf and hearing) better learn this process of writing.

**Keywords:** Sign writing. SignWriting. Mobility. Android. Accessibility. SignSMS.

## LISTA DE ILUSTRAÇÕES

Figura 1 – <i>SignWriting</i> : grupos de símbolos de mãos .....	21
Figura 2 – <i>SignWriting</i> : símbolos de tipos de mão .....	21
Figura 3 - <i>SignWriting</i> : símbolos para orientação da palma da mão .....	21
Figura 4 – <i>SignWriting</i> : símbolos de posições para as mãos no plano vertical e horizontal...21	
Figura 5 – Alfabeto manual com desenhos de mãos e em <i>SignWriting</i> .....	23
Figura 6 - <i>SignWriting</i> : elementos da configuração de mãos.....	24
Figura 7 - <i>SignWriting</i> : elementos da configuração de movimento.....	24
Figura 8 - <i>SignWriting</i> : elementos da configuração de expressão facial.....	24
Figura 9 - <i>SignWriting</i> : elementos da configuração de corpo .....	25
Figura 10 - <i>SignWriting</i> : exemplos de grafia para a palavra “surdo” .....	25
Figura 11 - Interação Humano-Computador: princípios de usabilidade .....	27
Figura 12 – Padrões de <i>design</i> : <i>Action Bar 1</i> .....	40
Figura 13 – Padrões de <i>design</i> : <i>Action Bar 2</i> .....	40
Figura 14 - Padrões de <i>design</i> : <i>Companion Widgets</i> .....	41
Figura 15 - Padrões de <i>design</i> : <i>Dashboard 1</i> .....	41
Figura 16 - Padrões de <i>design</i> : <i>Dashboard 2</i> .....	42
Figura 17 - Padrões de <i>design</i> : <i>Quick Actions 1</i> .....	43
Figura 18 - Padrões de <i>design</i> : <i>Quick Actions 2</i> .....	43
Figura 19 - Padrões de <i>design</i> : <i>Search Bar</i> .....	44
Figura 20 - Plataforma <i>Android</i> .....	46
Figura 21 - <i>Touch screen</i> : interação por meio do toque.....	48
Figura 22 - <i>Touch screen</i> : interação por meio de pressão longa .....	48
Figura 23 - <i>Touch screen</i> : interação por meio de <i>Swipe</i> .....	49
Figura 24 - <i>Touch screen</i> : interação por meio da ação de arrastar .....	49
Figura 25 - <i>Touch screen</i> : interação por meio do toque duplo .....	50
Figura 26 - <i>Touch screen</i> : interação por meio do movimento de pinça para fora .....	50
Figura 27 - <i>Touch screen</i> : interação por meio do movimento de pinça para dentro.....	51
Figura 28 - <i>SignEd</i> : tela principal.....	54
Figura 29 - <i>SignSim</i> : módulo Português escrito .....	54
Figura 30 - <i>SignSim</i> : resolução de conflito: uma palavra - mais de um sinal .....	55
Figura 31 - <i>SignTalk</i> Cliente: interação através da escrita da LIBRAS.....	56
Figura 32 - <i>SignTalk</i> Cliente: interação através do Português escrito .....	56
Figura 33 - <i>SignTalk</i> Servidor: informações de conexão à Sala 1 .....	57
Figura 34 - <i>SignMail</i> : tela principal .....	57
Figura 35 - <i>SignHTML</i> : tela principal .....	58
Figura 36 – Editor 3D: Editor do Pacote <i>Sign</i> (a) e editor do <i>SignHq</i> (b) .....	59
Figura 37 - <i>SignWebEdit</i> : tela principal.....	60
Figura 38 – <i>SW-Edit</i> : tela principal.....	62
Figura 39 – <i>SW-Edit</i> : Abas .....	63
Figura 40 – <i>Sw-Edit</i> : Botão Editar Dicionário .....	65
Figura 41 - <i>Sw-Edit</i> : criação de um dicionário de símbolos (1).....	65
Figura 42 - <i>Sw-Edit</i> : criação de um dicionário de símbolos (2).....	66
Figura 43 – <i>Sw-Edit</i> : criação de um dicionário de símbolos (3).....	66
Figura 44 – <i>Sw-Edit</i> : construção da palavra casa (1) .....	67
Figura 45 - <i>Sw-Edit</i> : construção da palavra casa (2) .....	68



Figura 46 – Sw-Edit: construção da palavra casa (3) .....	68
Figura 47 – Sw-Edit: construção da palavra casa (4) .....	69
Figura 48 – Sw-Edit: construção da palavra casa (5) .....	69
Figura 49 – SW-Edit: salvando um sinal (1) .....	71
Figura 50 – SW-Edit: salvando um sinal (2) .....	71
Figura 51 – SW-Edit: salvando um sinal (3) .....	72
Figura 52 – SW-Edit: salvando um sinal (4) .....	72
Figura 53 – SW-Edit: salvando um sinal (5) .....	73
Figura 54 – SW-Edit: salvando um sinal (6) .....	73
Figura 55 - SW-Edit: selecionando um dicionário.....	75
Figura 56 - SW-Edit: dicionário selecionado .....	75
Figura 57 - SW-Edit: seleção do dicionário na grade de grupos de símbolos.....	76
Figura 58 - SW-Edit: nenhum dicionário existente .....	76
Figura 59 – SW-Edit: Consulta pelo nome do sinal (1) .....	77
Figura 60 - SW-Edit: Consulta pelo nome do sinal (2) .....	78
Figura 61 - Consulta por parte do sinal em SignWriting (1) .....	78
Figura 62 - SW-Edit: Consulta por parte do sinal em SignWriting (2).....	79
Figura 63 - SW-Edit: Dicionário de símbolos selecionado .....	79
Figura 64 - SW-Edit: célula selecionada na área de texto .....	80
Figura 65 - SW-Edit: símbolo inserido na célula selecionada .....	80
Figura 66 - Sign WebMessage: caixa de entrada das mensagens .....	82
Figura 67 – Sign WebMessage: modo de consulta .....	82
Figura 68 – Sign WebMessage: busca de sinal por língua de sinais .....	83
Figura 69 - <i>Sign WebMessage</i> : módulo de criação.....	84
Figura 70 - <i>Sign WebMessage</i> : arquivo SWML.....	84
Figura 71 - <i>Sign 4 Me</i> : ícone da aplicação.....	85
Figura 72 - <i>Sign 4 Me</i> : interface para inserção de texto.....	86
Figura 73 - <i>Sign 4 Me</i> : animação do texto em língua de sinais .....	86
Figura 74 - <i>Sign 4 Me</i> : controle do avatar.....	86
Figura 75 - <i>SkySigns ASL Dictionary</i> : ícone do aplicativo .....	87
Figura 76 - <i>SkySigns ASL Dictionary</i> : escrita por meio do alfabeto manual em <i>SignWriting</i> ..	87
Figura 77 - <i>SkySigns ASL Dictionary</i> : escrita por meio do alfabeto romano.....	88
Figura 78 - <i>SkySigns ASL Dictionary</i> : resultado da busca do sinal .....	88
Figura 79 - <i>SMS to Sign Language for Mobile Phones</i> : texto escrito pelo usuário.....	89
Figura 80 - <i>SMS to Sign Language for Mobile Phones</i> : envio de uma mensagem em formato texto .....	90
Figura 81 - <i>SMS to Sign Language for Mobile Phones</i> : tela contendo a mensagem recebida em texto e em animação .....	90
Figura 82 - <i>SMS to Sign Language for Mobile Phones</i> : tradução em sinais da mensagem de texto .....	90
Figura 83 - <i>Signing Savvy</i> : ícone da aplicação.....	91
Figura 84 - <i>Signing Savvy</i> : vídeo contendo instruções de um sinal em ASL .....	92
Figura 85 - <i>Signing Savvy</i> : tradução da palavra help de texto para ASL.....	92
Figura 86 - <i>Baby Sign Language Dict</i> : tela inicial da aplicação .....	93
Figura 87 - <i>Baby Sign Language Dict</i> : Lista de palavras do ASL Dictionary .....	93
Figura 88 - <i>Baby Sign Language Dict</i> : Video explicando a execução do sinal .....	94
Figura 89 - <i>Baby Sign Language Dict</i> : Mostra as palavras separadas por categorias .....	94

Figura 90 - Baby Sign Language Dict: lista de sinais.....	94
Figura 91 - Baby Sign Language Dict: Bottom Bar.....	95
Figura 92 - ASL American Sign Language: letra "A" em ASL.....	96
Figura 93 - ASL American Sign Language: letra "B" em ASL.....	96
Figura 94 - ASL American Sign Language: configuração .....	96
Figura 95 - PicsArt: Opções de clipart .....	97
Figura 96 - PicsArt: Opções de efeitos de texto.....	98
Figura 97 - PicsArt: Opções de balões de diálogo.....	98
Figura 98 - PicsArt: Menu de Desenho .....	98
Figura 99 - PicsArt: Exemplo de manipulação de objetos.....	99
Figura 100 - PicsArt for kids: Tela inicial .....	100
Figura 101 - PicsArt for kids: Cenário com desenho .....	101
Figura 102 - PicsArt for Kids: Opções de pincel .....	101
Figura 103 - PicsArt for kids: Colorindo cenários.....	101
Figura 104 - PicsArt for Kids: Desenhando a mão livre.....	102
Figura 105 - PicSay: Efeitos .....	102
Figura 106 - PicSay: Opções do editor de imagensOpções do editor de fotos .....	103
Figura 107 - PicSay: Inserção de texto .....	103
Figura 108 - PicSay: ClipArt .....	103
Figura 109 - Signing Savvy: Action Bar.....	104
Figura 110 - signing savvy: Action bar 2.....	104
Figura 111 - Baby Sign Language Dict: Dashboard .....	105
Figura 112 - PicsArt: Quick Action .....	106
Figura 113 - Sign 4 Me: Quick Search.....	106
Figura 114 - SkySigns ASL Dictionary: Quick Search.....	107
Figura 115 - Distribuição dos usuários por versão da plataforma Android .....	111
Figura 116 - MotoDev Studio: perspectiva Package Explorer do projeto .....	115
Figura 117 - Tamanho relativo das imagens de acordo com cada densidade.....	117
Figura 118- Modelo de tarefa "Escrever sinal" .....	123
Figura 119 - Modelo de Tarefa "Enviar Mensagem" .....	124
Figura 120 - Modelo de tarefa "Receber mensagem" .....	125
Figura 121 - Modelo de tarefa "Apagar mensagem" .....	125
Figura 122 - SignSMS - tela inicial: mockup x resultado .....	127
Figura 123 - SignSMS: tela para escrita da mensagem, mockup x resultado .....	128
Figura 124 - SignSMS: menu categoria de símbolos.....	128
Figura 125 - SignSMS: opções do grupo de mãos.....	129
Figura 126 - SignSMS: opções de configurações de mãos do Grupo 1.....	130
Figura 127 - SignSMS: simbolo inserido na área de construção de sinal .....	130
Figura 128 - SignSMS: menu para manipulação de um símbolo selecionado .....	131
Figura 129 - SignSMS: menu de opções da tela de construção da mensagem .....	133
Figura 130 - SignSMS: tela para inserção dos dados de envio da mensagem.....	134
Figura 131 - SignSMS: tela de mensagens recebidas.....	134
Figura 132 - SignSMS: leitura da mensagem recebida .....	135
Figura 133 - SignSMS: menu de opções da tela de leitura da mensagem.....	136

## LISTA DE TABELAS

Tabela 1 – <i>SignWriting</i> : símbolos de movimentos .....	22
Tabela 2 - <i>SignWriting</i> : símbolos de contatos .....	22
Tabela 3 - Comparativo entre as ferramentas do pacote Sign .....	61
Tabela 4- comparativo das funcionalidades de softwares que fazem uso do <i>SignWriting</i> ...	108
Tabela 5- SignSMS: quantidade de símbolos na aplicação .....	116
Tabela 6 - tabela de signos do SignSMS.....	121
Tabela 7 - SignSMS: Menu de ações .....	132

## LISTA DE ABREVIATURAS E SIGLAS

ASL – *American Sign Language*

CC – Ciência da Computação

FACIN - Faculdade de Informática

HTML - HyperText Markup Language

IHC – Interação Humano-Computador

LS – Língua(s) de Sinais

Mobile – Dispositivos móveis

PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul

SDK – Software Development Kit

SMS – *Short Message Service* (Serviço de Mensagem Curta)

TCC – Trabalho de Conclusão de Curso

TCC I – Trabalho de Conclusão de Curso I

TCC II – Trabalho de Conclusão de Curso II

TI – Tecnologia da Informação

W3C - Consórcio *World Wide Web*

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>14</b>
1.1. Objetivos .....	16
1.2. Organização do Volume .....	16
<b>2. REVISÃO BIBLIOGRÁFICA .....</b>	<b>18</b>
2.1. Língua de Sinais.....	18
2.2. SignWriting.....	20
2.3. Interação Humano-Computador .....	26
2.4. Consórcio World Wide Web .....	28
2.5. Padrões e princípios de design .....	35
2.6. Tecnologia móvel .....	45
2.6.1. Recursos de interação.....	47
<b>3. APLICAÇÕES RELACIONADAS .....</b>	<b>52</b>
3.1. Aplicações com suporte à Língua de Sinais .....	52
3.1.1. Pacote Sign.....	53
3.1.2. SW-Edit.....	62
3.1.3. Sign WebMessage .....	81
3.1.4. Sign 4 Me.....	85
3.1.5. SkySigns ASL Dictionary .....	87
3.1.6. SMS to Sign Language for Mobile Phones .....	88
3.1.7. Signing Savvy.....	91
3.1.8. Baby Sign Language Dict. Demo .....	92
3.1.9. ASL American Sign Language .....	95
3.2. Aplicações interativas para dispositivos móveis.....	97
3.2.1. PicsArt .....	97

3.2.2.	PicsArt for Kids .....	99
3.2.3.	PicSay .....	102
3.3.	Padrões .....	104
3.4.	Comparativo das aplicações com suporte a escrita de sinais.....	107
<b>4.</b>	<b>PROJETO .....</b>	<b>110</b>
4.1.	Arquitetura do SignSMS.....	110
4.1.1.	Padrões, princípios de design e boas práticas no SignSMS .....	110
4.1.2.	Estrutura do projeto .....	114
4.1.3.	Descrição dos componentes .....	117
4.2.	Modelagem .....	119
4.2.1.	Perguntas .....	119
4.2.2.	Cenários .....	120
4.2.3.	Tabela de signos.....	121
4.2.4.	Modelos de tarefas .....	122
4.3.	Interface.....	125
4.4.	Funcionalidades x classes Android.....	136
4.5.	Recursos de desenvolvimento .....	137
<b>5.</b>	<b>CONCLUSÃO .....</b>	<b>138</b>
5.1.	Limitações e trabalhos futuros .....	140
	<b>REFERÊNCIAS.....</b>	<b>143</b>

## 1. INTRODUÇÃO

A Computação permeia diferentes áreas do conhecimento, portanto, de forma natural, pode ser vista como um seleiro para aplicações inter/multidisciplinares. O trabalho aqui proposto está relacionado à área da Ciência da Computação, mas com inserção na área de educação especial. Inclui estudos relacionados a tecnologias móveis, tecnologias assistivas, acessibilidade bem como à educação de pessoas surdas, principalmente no que se refere ao uso das línguas de sinais em ambiente computadorizado.

No caso de pessoas surdas, concordamos com Campos & Silveira (2007) de que o avanço das tecnologias e de estudos na área da Educação de Surdos permitiu uma mudança de paradigma no uso e desenvolvimento de tecnologias para que dêem suporte ao uso de línguas de sinais. Nas palavras das autoras (p. 2233):

*No contexto de educação de surdos e uso de TICs, tornam-se ultrapassados trabalhos que tratam de antigas crenças e estereótipos que abordam incapacidades e dificuldades intrínsecas a sujeitos surdos, sem apresentar alternativas de soluções que utilizem língua de sinais. Faz-se necessário, portanto, um posicionamento da equipe de desenvolvimento do projeto do software no que se refere à abordagem de educação de surdos.*

Este trabalho segue o paradigma da educação bilíngue porque, de acordo com Campos (2001), Campos & Silveira (2007), Karnopp (1994), Quadros (2012), Rosa (2009), dentre outros, considera que duas línguas co-existem em um mesmo espaço, sendo que a primeira língua (ou língua materna) das pessoas surdas é a língua de sinais enquanto que a segunda língua (ou língua estrangeira) é a língua oral da comunidade ouvinte a que estão inseridos.

No que se refere à área da Ciência da Computação, a Sociedade Brasileira de Computação (SBC, 2006) destaca grandes desafios da pesquisa em Computação no Brasil para o período de 2006 a 2016. Nesse documento, são propostos cinco desafios, conforme seguem:

1. Gestão da Informação em grandes volumes de dados multimídia distribuídos.
2. Modelagem computacional de sistemas complexos artificiais, naturais e socioculturais e da interação homem-natureza.

3. Impactos para a área da computação da transição do silício para novas tecnologias.
4. Acesso participativo e universal do cidadão brasileiro ao conhecimento.
5. Desenvolvimento tecnológico de qualidade: sistemas disponíveis, corretos, seguros, escaláveis, persistentes e ubíquos.

Nosso trabalho pode ser relacionado ao desafio 4, que se refere as barreiras tecnológicas, educacionais, culturais, sociais e econômicas, que impedem o acesso e a interação. Esse desafio tem por objetivo:

*vencer essas barreiras, por meio da concepção de sistemas, ferramentas, modelos, métodos, procedimentos e teorias capazes de endereçar, de forma competente, a questão do acesso do cidadão brasileiro ao conhecimento. Este acesso deve ser universal e participativo, na medida em que o cidadão não é um usuário passivo, o qual recebe informações, mas também participa da geração do conhecimento. Apenas por meio da oportunidade de participação da construção do conhecimento é que o acesso vai poder levar a uma plena e consciente utilização do conhecimento disponibilizado. (SBC, 2006, pág. 17)*

Essa questão, ainda vem ao encontro do princípio da *Web*, proposto pelo W3C Brasil (CGI.br, 2010), que traz a seguinte afirmação:

*o principal valor da Web é o social. Mais do que tecnológico, este é um ambiente de comunicação humana, de transações comerciais, de oportunidades para compartilhar conhecimentos e, para ser um ambiente universal, deve estar disponível para todas as pessoas, independentemente dos equipamentos e softwares que utilizem, principalmente da cultura em que inserem, da localização geográfica, das habilidades físicas ou mentais, das condições socioeconômicas ou de instrução. (CGI.br, pág. 9)*

Diante desse cenário, surgem as seguintes questões de pesquisa: *Que características deve ter uma aplicação computacional interativa para permitir uma comunicação a distância entre pessoas surdas, no contexto de uma educação bilíngue? E entre pessoas surdas e pessoas ouvintes, considerando o contexto de mobilidade tecnológica?*

Nossa hipótese é de que as inovações tecnológicas apresentadas pelos dispositivos móveis, particularmente os dispositivos que oferecem recursos de *touch screen*, podem auxiliar no desenvolvimento de uma ferramenta que se torne um facilitador na



comunicação entre as pessoas surdas, e dessas com pessoas ouvintes. Mas, para tanto, essa ferramenta deverá suportar a escrita de textos em língua de sinais. Nosso projeto pretende, portanto, minimizar as dificuldades de comunicação, ou maximizar essa comunicação de forma a potencializar o uso da língua de sinais, na sua forma escrita, como meio de comunicação. Como público-alvo da ferramenta, destacamos pessoas surdas ou ouvintes que desejem se comunicar por meio da escrita de línguas de sinais utilizando dispositivo móvel.

### **1.1. Objetivos**

O objetivo geral desta pesquisa é desenvolver um protótipo de ferramenta interativa para comunicação a distância baseada em língua de sinais (LS). Em termos gerais, deverá funcionar de maneira similar a um Serviço de Mensagem Curta (SMS) ou a um correio eletrônico no qual o usuário poderá enviar textos escritos em língua de sinais utilizando um telefone móvel.

Como objetivos específicos, podemos citar:

- Discutir a importância da área de Interação Humano-Computador (IHC) para o desenvolvimento de sistemas computacionais interativos.
- Identificar padrões e estilos de interface para dispositivos móveis, em especial para telefones celulares.
- Construir conhecimento sobre o uso de língua de sinais em dispositivos móveis.
- Sugerir padrões, guias de estilo e estilos de interface, além de boas práticas de interface para aplicações em dispositivos móveis com suporte à comunicação em língua de sinais.
- Construir conhecimento sobre o desenvolvimento de aplicações para dispositivos móveis

### **1.2. Organização do Volume**

Este documento está assim estruturado: o capítulo 2 descreve o estudo inicial sobre aspectos teóricos relacionados ao trabalho. Os seguintes temas são tratados nesse capítulo: língua de sinais, sistema *SignWriting*, Interação Humano-Computador e tecnologia móvel.

O capítulo 3 apresenta alguns exemplos de aplicações. A seção 3.1 apresenta exemplos de *softwares* que fazem uso de língua de sinais. Em especial, as ferramentas que compõem o Pacote *Sign* (Campos, 2001) e o *SW-Edit* porque já utilizam o sistema *SignWriting*. O foco da seção 3.2 foram aplicações com suporte à edição, que foram desenvolvidas para dispositivos móveis. Estas seções também apresentam a descrição das funcionalidades, recursos e problemas encontrados nas principais aplicações analisadas, que são discutidas em maior profundidade para verificar sua adequação ao projeto de interface do *SignSMS*.

O projeto do *SignSMS* está descrito no capítulo 4 no qual é apresentada a relação entre a proposta de interface para o protótipo e o resultado obtido na desenvolvimento das mesmas, a estrutura do projeto do *SignSMS*, bem como uma descrição das funcionalidades previstas.

No capítulo 5 temos as considerações finais sobre o projeto bem como são apresentados os principais desafios, limitações e oportunidades de melhorias como trabalhos futuros.

Este documento se encerra com as referências utilizadas ao longo do texto.

## 2. REVISÃO BIBLIOGRÁFICA

A fundamentação teórica adotada neste trabalho está baseada na educação bilíngue para surdos, conforme já citado, e, portanto, questões relacionadas à língua de sinais e ao Sistema *SignWriting* fazem parte desse embasamento teórico.

Também são focos de estudo, questões relacionadas à área de Interação Humano-Computador, que inclui fatores de qualidade de uso tais como usabilidade, comunicabilidade e acessibilidade, padrões de design, guias de estilos e estilos de interação.

### 2.1. *Língua de Sinais*

As línguas de sinais são línguas de modalidade gestual-visual, sendo utilizadas pela maioria das pessoas surdas. De acordo com Campos (2001), Karnopp (1994), Rosa (2009), e outros autores, no Brasil existem duas línguas de sinais: a língua brasileira de sinais Kaapor (LSKB), que é usada por índios surdos da selva amazônica, e a língua de sinais brasileira usada nos centros urbanos, que já foi referenciada como LSCB e, desde 1993, é denominada de LIBRAS - Língua Brasileira de Sinais.

A LIBRAS é definida na Lei Nº 10.436, de 24 de abril de 2002 (Brasil, 2002), que dispõe sobre a Língua Brasileira de Sinais - Libras e dá outras providências:

*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.*

Na categoria de língua, não são universais. Portanto, a LIBRAS é diferente da Língua de Sinais de Portugal, que é diferente da Língua de Sinais do Uruguai, que é diferente da Língua de Sinais da Argentina, e assim por diante. Também não podem ser consideradas como representações espaciais das línguas faladas.

As línguas de sinais possuem uma gramática com regras próprias. De acordo com Campos (2001), o sistema fonológico estuda as configurações da(s) mão(os), o ponto de articulação e os movimentos, que são tidos como parâmetros primários, e a região de

contato, orientação e disposição da(s) mão(s), que são consideradas como parâmetros secundários. A configuração da(s) mão(s) refere-se às diversas formas que a(s) mão(s) toma(m) na realização do sinal. Ainda, a configuração de uma mão pode permanecer a mesma ou ser alterada durante a articulação de um sinal, como é o caso dos sinais compostos<sup>1</sup>. O Sistema Morfológico, de forma geral, estuda a formação dos sinais e se preocupa, por exemplo, com o gênero, grau, tempo e negação na LIBRAS. Já o Sistema Sintático estuda a organização sintática básica em LIBRAS, que é Objeto – Sujeito – Verbo. Essas, dentre outras questões da LS, são retomadas na seção que trata do sistema de escrita denominado *SignWriting*.

Também, no que se refere à LIBRAS, é importante destacar o Decreto nº 5.626, de 22 de dezembro de 2005 (Brasil, 2005), que regulamenta a Lei no 10.436, de 24 de abril de 2002, que dispõe sobre a Língua Brasileira de Sinais, e o art. 18 da Lei no 10.098, de 19 de dezembro de 2000. O capítulo II desse Decreto trata sobre a inserção da LIBRAS como disciplina curricular, conforme segue.

## CAPÍTULO II

### DA INCLUSÃO DA LIBRAS COMO DISCIPLINA CURRICULAR

*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.*

Os dispositivos legais ora citados servem para demonstrar o potencial da ferramenta aqui proposta como recurso para o processo de ensino e aprendizagem da LIBRAS. Importante também citar que cursos de formação, como o de Licenciatura e o de Bacharelado em Letras Libras oferecido pela Universidade Federal de Santa Catarina, no

---

<sup>1</sup> Sinais compostos são constituídos por mais de um sinal. Por exemplo, o sinal de ESCOLA é formado pelo sinal de CASA mais o de ESTUDAR. O sinal de IGREJA também utiliza o sinal de CASA, só que seguido do sinal de CRUZ.

âmbito da Universidade Aberta do Brasil (UFSC/UAB<sup>2</sup>), por exemplo, possuem em sua matriz curricular três disciplinas de escrita de língua de sinais como obrigatórias. Ainda, em conversa com a professora dessas disciplinas, Dra. Marianne Stumpf, uma ferramenta como o *SignSMS* seria de grande utilidade para apoiar as conversações via telefone que, ainda hoje, ocorrem baseadas na escrita do Português.

## 2.2. *SignWriting*

O Sistema *SignWriting* é um sistema de escrita para escrever línguas de sinais, que foi criado em 1974 por Valerie Sutton (Quadros, 2012; *SignWriting*, 2012).

De maneira similar ao alfabeto romano, que pode ser utilizado para escrever diferentes línguas orais, o *SignWriting* pode ser utilizado para escrever diferentes línguas de sinais. Para isso, possui um conjunto de símbolos para representar expressões faciais, configurações de mãos e dedos, toques, movimentos e pontos de articulação, que é o local onde o sinal é realizado no espaço. De acordo com Stumpf (2005, 2008, 2012), o *SignWriting* comporta aproximadamente 900 símbolos.

No Brasil, esse sistema foi apresentado em 1996 por Antônio Carlos da Rocha Costa, nossa orientadora, Márcia de Borba Campos e Marianne Rossi Stump, que foi a 1ª surda brasileira a escrever em *SignWriting* (Sutton, 2012). Hoje, o ensino da escrita de sinais compõe o currículo de cursos de formação para professores de LIBRAS e de formação de tradutores/intérpretes de Libras/Português, conforme já citado.

Em termos gerais, conforme a literatura (Campos; 2001; Quadros, 2012; Stumpf, 2005, Sutton, 2012), esse sistema é definido por estruturas básicas<sup>3</sup> que são a posição da mão, movimento e contato, sendo que as mãos são organizadas em grupos de acordo com os dedos utilizados (Figura 1). Cada mão pode, ainda, estar fechada, aberta ou “circular” (Figura 2), virada com a palma da mão para o sinalizador, de lado ou com o dorso da mão para esse (Figura 3). Pode, também, estar no plano horizontal ou vertical (Figura 4).

---

<sup>2</sup> A matriz curricular do curso de Licenciatura em Letras Libras pode ser acessada em [http://www.libras.ufsc.br/hiperlab/avalibras/moodle/prelogin/adl/fb/logs/Arquivos/grade\\_curricular/Licenciatura.pdf](http://www.libras.ufsc.br/hiperlab/avalibras/moodle/prelogin/adl/fb/logs/Arquivos/grade_curricular/Licenciatura.pdf). Já a matriz curricular do Bacharelado em Letras Libras está disponível em [http://www.libras.ufsc.br/hiperlab/avalibras/moodle/prelogin/adl/fb/logs/Arquivos/grade\\_curricular/Bacharelado.pdf](http://www.libras.ufsc.br/hiperlab/avalibras/moodle/prelogin/adl/fb/logs/Arquivos/grade_curricular/Bacharelado.pdf).

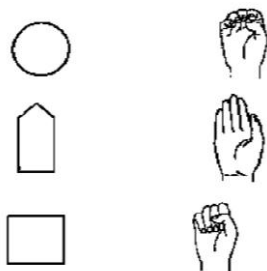
<sup>3</sup> Além destas, há símbolos para expressões faciais, pontos de articulação, sinais de pontuação, dentre outros.

Figura 1 – *SignWriting*: grupos de símbolos de mãos



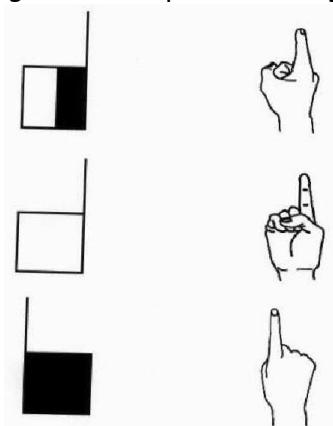
Fonte: Campos (2001)

Figura 2 – *SignWriting*: símbolos de tipos de mão



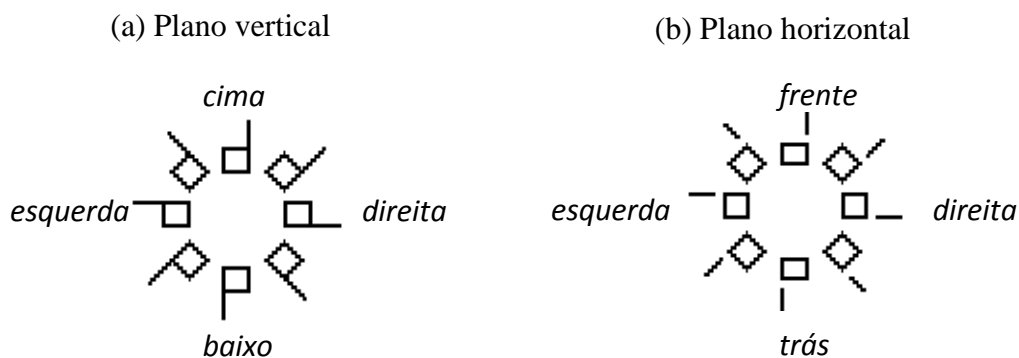
Fonte: Sutton (2012)

Figura 3 - *SignWriting*: símbolos para orientação da palma da mão



Fonte: Sutton (2012)

Figura 4 – *SignWriting*: símbolos de posições para as mãos no plano vertical e horizontal












Fonte: Campos (2001)

No que se refere aos movimentos, pode haver movimento de dedos, mão ou braço.

A Tabela 1 apresenta exemplos de símbolos de movimento.







Tabela 1 – *SignWriting*: símbolos de movimentos

	Fechar a articulação do meio do dedo
	Abrir a articulação do meio do dedo
	Fechar a articulação do nó dos dedos
	Abrir a articulação do nó dos dedos
	Abrir e fechar a articulação do nó dos dedos juntos
	Abrir e fechar a articulação do nó dos dedos alternadamente
	Mão direita para frente (paralela ao chão)
	Mão esquerda para frente (paralela ao chão)
	Mão direita para frente (perpendicular ao chão)
	Mão esquerda para frente (perpendicular ao chão)

Fonte: Campos (2001)

Quanto ao contato, pode ser de mão com mão, mão com corpo ou mão com cabeça, por exemplo, conforme ilustrados na Tabela 2.

Tabela 2 - *SignWriting*: símbolos de contatos

	Tocar em outra parte do corpo
	Pegar em alguma parte do corpo ou roupa
	Tocar entre duas partes do corpo, geralmente entre dois dedos
	Bater em alguma parte do corpo
	Raspar em alguma parte do corpo saindo da superfície
	Esfregar em alguma parte do corpo (tocar e mover sem sair da superfície)

Fonte: Campos (2001)

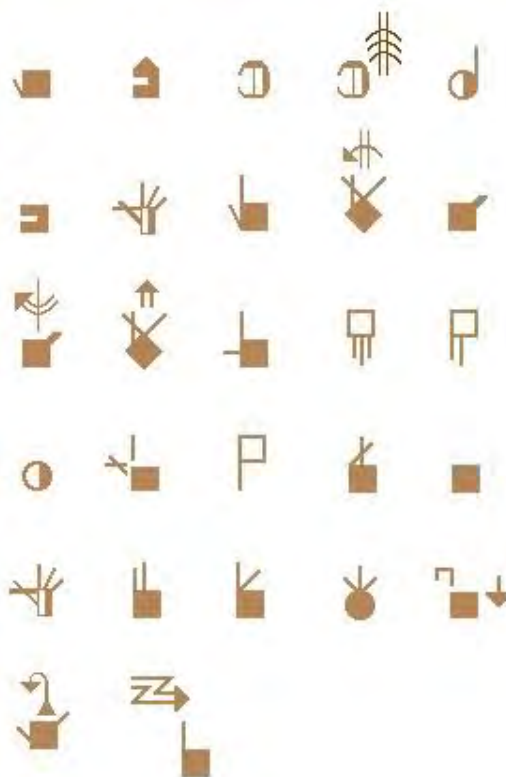
Para fins de ilustração, apresentamos o alfabeto manual representado por desenhos de mãos e pela escrita em *SignWriting* (Figura 5).

Figura 5 – Alfabeto manual com desenhos de mãos e em *SignWriting*

(a) Representado pelo desenho das mãos



(b) Escrito em *SignWriting*



Fonte: <http://escritadesinais.wordpress.com/>

Nesse processo de escrita, Stumpf (2005) explicita os elementos que constituem os sinais. Na Figura 6, estão descritos os elementos que constituem as configurações de mão enquanto a Figura 7 especifica os elementos para os movimentos. A Figura 8 e Figura 9 descrevem, respectivamente, os elementos das configurações de escrita para expressão facial e corpo, quando relevantes ao sinal.



Figura 6 - *SignWriting*: elementos da configuração de mãos

<p>Mão (direita/esquerda):  <b>Configuração de mão:</b>          Grupo:          Sentido:          Palma:          Posição:          Configuração dos dedos:          Local:  <b>Configuração do braço:</b>          Plano do antebraço:          Ângulo do antebraço:          Plano do braço:          Ângulo do braço:</p>
---

Fonte: Stumpf (2005, p. 58)

Figura 7 - *SignWriting*: elementos da configuração de movimento

<p><b>Movimento dos dedos (interno):</b>          Movimento interno:          Frequência:          Dedo (s):  <b>Movimento da mão (externo):</b>  <u>Contato:</u>          Tipo de contato:          Frequência:          Local da mão em contato:          Local em contato com a mão:  <u>Movimento:</u>          Tipo de movimento:          Posição:          Frequência:          Tamanho:  <u>Dinâmica:</u></p>
---

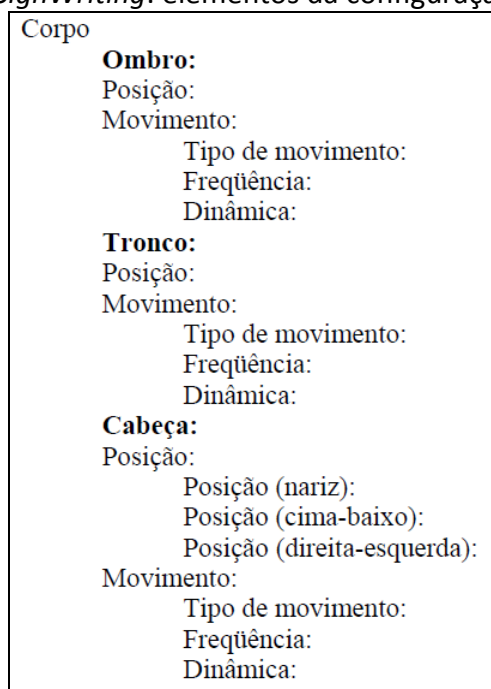
Fonte: Stumpf (2005, p. 58)

Figura 8 - *SignWriting*: elementos da configuração de expressão facial

<p>Expressão Facial:          Testa:          Sobrancelha:          Olhos:          Olhar:          Bochecha:          Nariz:          Boca:          Língua:          Dentes:          Outros:</p>
---

Fonte: Stumpf (2005, p. 59)

Figura 9 - *SignWriting*: elementos da configuração de corpo



Fonte: Stumpf (2005, p. 59)

Ainda, apesar de haver essa especificação na descrição dos sinais, no que se refere aos símbolos, o processo de escrita da Língua Brasileira de Sinais ainda precisa de padronizações. Quadros (2012) destaca que a produção escrita dos sinais pode se diferenciar de pessoa para pessoa, de forma que cada um escreve como acha que deve ser escrito o sinal. Também descreve que este é um processo natural, tal como ocorreu com a escrita das línguas orais quando cada pessoa escrevia o som da forma em que achava mais adequado. Segundo a autora (idem):

*A escrita passou a ser padronizada ao longo do tempo com a invenção da imprensa. A imprensa foi o meio em que a escrita foi difundida rapidamente. A escrita tornou-se pública e naturalmente foi sendo padronizada. (pág. 2)*

Stumpf (2005) exemplifica essa questão com cinco formas de grafar o sinal para a palavra “surdo”, conforme Figura 10, sendo que todas essas grafias estão corretas.

Figura 10 - *SignWriting*: exemplos de grafia para a palavra “surdo”



Fonte: Stumpt (2005)

Sobre essa (falta de) padronização, Stump (idem) explica:

*Na Dinamarca, já há alguns anos, eles decidiram só escrever contato quando absolutamente necessário. E, como o sinal de surdo, é muitas vezes executado sem um real contato com a face e desde que a posição da mão torna claro que o sinal começa em cima e termina tocando mais embaixo, eles escrevem apenas um contato. O sistema é flexível. (pág. 57)*

*Em relação aos símbolos dentro de um sinal escrito, não há ordem específica. Em geral, não é necessário ter uma ordem específica porque os símbolos ficam organizados no papel de modo análogo àquele que os elementos que representam estão organizados no gesto que constitui o sinal, ao redor do ponto de articulação. (...) Ainda assim, existem muitos casos em que é preciso padronizar uma ordem de escrita (...). As comunidades surdas é que precisam definir essa padronização, como parte da ortografia de sinais de sua própria língua de sinais. (pág. 166 - 167)*

Outra questão, também necessária a este trabalho, está relacionada ao sentido da escrita, que pode ser vertical, de cima para baixo, ou horizontal, da esquerda para direita, na mesma ordem em que os sinais são sinalizados.

### **2.3. Interação Humano-Computador**

Cada vez mais a tecnologia vem fazendo parte do dia-a-dia das pessoas, de forma que ela já ocupa um papel importante nas nossas vidas. Entretanto, apesar dessa disponibilidade de recursos e do barateamento no custo dos equipamentos, algumas pessoas não têm conseguido se beneficiar dessa situação. Os usuários de tecnologias possuem características diferenciadas, e essa diversidade de necessidades no uso dos recursos e na forma de como podem fazê-lo, podem ser impeditivas.

A área de Interação Humano-Computador (IHC) define critérios de qualidade de uso. Dentre esses, estão usabilidade, acessibilidade e comunicabilidade. Se uma aplicação é desenvolvida de acordo com esses, ela possui maiores chances de ser utilizada por diferentes pessoas, independente de suas características.

É comum em projetos de desenvolvimento de *software* que o foco esteja direcionado ao desenvolvimento do sistema, ou seja, “às coisas que acontecem dentro do sistema”. A questão da interação com o usuário, por vezes, é deixada em segundo plano, o

que pode resultar em um sistema sem erros funcionais, mas também sem usuários porque descuida desta relação usuário-sistema.

Concordando com Pontes & Orth (1999), a interface pode determinar a aceitação, ou não, de um sistema pelo seu usuário e, quando se trata de *software* que visa o uso por pessoas com alguma deficiência, seja ela temporal ou permanente, o estudo da área de IHC se torna um requisito fundamental para a obtenção de uma interface que seja eficiente, eficaz e fácil de usar. Para garantir que esse resultado seja alcançado no projeto do *SignSMS* será necessário que os requisitos de usabilidade e acessibilidade sejam considerados no processo de desenvolvimento e de design nas diferentes etapas de desenvolvimento do protótipo.

De acordo com Preece et al. (2005), o critério de usabilidade trata de aspectos como o esforço necessário para o usuário realizar uma determinada tarefa utilizando o sistema, da facilidade de aprender a utilizá-lo, se será fácil lembrar como utilizá-lo e também se o sistema é realmente útil para realizar as atividades que o usuário necessite. Na Figura 11 podemos ver os principais princípios de usabilidade e seus objetivos, descritos por Santos (2008).

Figura 11 - Interação Humano-Computador: princípios de usabilidade



Fonte: Santos (2008, p.13)

Já a acessibilidade trata das questões que dificultam ou impossibilitam a interação do usuário com o sistema seja por falta de alguma aptidão física ou alguma característica

especial do usuário. A entidade Acessibilidade Brasil (2012a) discute sobre acessibilidade na *Web* e afirma que:

*Representa para o nosso usuário não só o direito de acessar a rede de informações, mas também o direito de eliminação de barreiras arquitetônicas, de disponibilidade de comunicação, de acesso físico, de equipamentos e programas adequados, de conteúdo e apresentação da informação em formatos alternativos.*

Consideramos que *SignSMS* possa vir a ser uma ferramenta que permita uma maior acessibilidade no sentido de possibilitar que pessoas surdas possam se comunicar na sua própria língua, transpondo o desafio de precisar se valer de uma segunda língua como meio de comunicação escrita por meio de dispositivos móveis.

Com relação ao critério de comunicabilidade, de acordo com Prates et al. (2000), esse pode ser definido como a capacidade do sistema transmitir para usuário de forma eficaz e eficiente as intenções e princípios de interação que guiaram seu design. Desta forma, se o sistema possuir um alto nível de comunicabilidade, o usuário deve ser capaz de entender para que o sistema serve, quais as vantagens em utilizá-lo, como o sistema funciona e quais são os princípios de interação com o sistema.

No escopo de estudo da área de IHC, verificamos que o contexto relacionado a aplicações para dispositivos móveis ainda carece de pesquisas e definições porque, conforme W3C Padrões (2012), a maioria dos padrões, diretrizes, estilos de interface, dentre outras recomendações, ainda estão direcionadas a aplicações do tipo *desktop*, mesmo que para a *Web*. Equipamentos com tamanho reduzido de tela requerem regras próprias, portanto, foi realizado um estudo das questões que incluem usabilidade, comunicabilidade e acessibilidade, padrões, guias de estilo e estilos de interface para aplicações em dispositivos móveis. Tal análise encontra-se distribuída nas seções que tratam da análise das aplicações estudadas e da modelagem do *SignSMS*.

## **2.4. Consórcio World Wide Web**

O Consórcio *World Wide Web* (W3C), que é uma organização internacional, define uma série de padrões para construção de conteúdos para *Web*. Dentre esses, foi criado o *Web Content Accessibility Guidelines 1.0 e 2.0* (WAI) que contém recomendações, conforme, Acessibilidade Brasil (2012b):

*As recomendações contidas nesse documento explicam como tornar o conteúdo Web acessível a pessoas com deficiências. Destinam-se a todos os criadores de conteúdo Web (autores de páginas e projetistas de sites) e aos programadores de ferramentas para criação de conteúdo.*

O W3C define questões relacionadas ao desenvolvimento (implementação) das aplicações, ou seja, questões relacionadas à codificação do *site* em si. Tendo em mente esta preocupação em garantir que as aplicações tenham qualidade e estabilidade, conforme W3C Padrões (2012) e Andrew<sup>4</sup> (2012), é que certas convenções de boas práticas para o desenvolvimento de *sites* vêm sendo criadas ao longo do tempo.

O W3C, por meio da *Mobile Web Initiative* (W3C Mobile, 2012), desenvolveu a cartilha “Boas práticas em *Web Móvel*” para os usuários que pretendem desenvolver aplicações *on-line* para dispositivos móveis. Nesta cartilha são propostas dez regras, das quais descreveremos as que, inicialmente, foram consideradas como úteis ao desenvolvimento do protótipo *SignSMS*.

- Faça seu projeto para uma *Web* única: este conceito leva em conta que o projeto gerado possa ser executado em diversos dispositivos, sem que haja a perda de funcionalidades, o que permitirá o atendimento a um maior número de usuários. Está subdividido em:

- Capacidades: propõe que os recursos oferecidos pelos dispositivos sejam utilizados na aplicação. Desta forma, podem facilitar a interação dos usuários com a aplicação. Fazendo uma referência ao nosso trabalho, pretendemos estudar os recursos comuns oferecidos pelos dispositivos móveis que operam sob o sistema operacional (SO) *Android*, a partir de uma versão base para permitir uma navegação mais rica aos usuários.
- Problemas na implementação: está relacionado aos problemas oriundos da implementação. Relacionando ao item anterior, por exemplo, há recursos oferecidos nos dispositivos que podem reduzir o tempo de implementação de uma aplicação, como, por exemplo, o recurso de *zoom* ou de rotação automática da tela. Para tanto, cabe ao desenvolvedor

---

<sup>4</sup> Membro do *Web Standards Project* (WaSP), que é uma iniciativa para definição de normas que assegurem o acesso simples e acessível às tecnologias *Web* para todos. Mais informações em <http://www.Webstandards.org>.

gerenciar a maneira de como a aplicação utilizará essas “bibliotecas”, ao invés do desenvolvedor ter que criar um *framework* para a aplicação implementar o método de *zoom* ou o de rotação.

- Testes: sugere que o desenvolvedor faça testes em diferentes dispositivos e emuladores, a fim de capturar algum erro no código que poderia passar despercebido, quando aplica testes em apenas um cenário.
- Confie nos padrões *Web*: este conceito eleva a importância da utilização de padrões para o desenvolvimento das aplicações, uma vez que o mercado oferece diferentes tipos de dispositivos que, por sua vez, possuem configurações distintas de navegação na *Web*. Dentre as características deste conceito citamos, dentre outras:
  - Compatibilidade de formato: está relacionado à compatibilidade do dispositivo com relação ao conteúdo que ele suporta. Nossa aplicação terá que enviar as mensagens escritas em *SignWriting* em um formato que seja suportado pelos dispositivos móveis que operam sob *SO Android*.
  - Formato preferido de conteúdo: relacionado às preferências do usuário, propõe que sempre que possível o usuário possa escolher o formato de envio do conteúdo desejado.
  - Mensagens de erro: define que as mensagens de erro geradas pela aplicação devem ser de fácil entendimento ao usuário e que o usuário possa regressar ao conteúdo/estado anterior da aplicação, antes da ocorrência do erro. Também pode estar relacionada ao tratamento e prevenção de erros.
- Evite os riscos conhecidos: este conceito aborda estratégias para minimizar os problemas causados por uma prototipação inadequada do projeto, reduzindo os problemas de usabilidade e acessibilidade que podem ser causados pelas restrições de tamanho de telas, botões e outras características dos dispositivos móveis. Destacamos os itens abaixo:

- Janelas *pop-ups*: devem ser evitadas janelas *pop-up* bem como mudanças de telas sem informar o usuário.
  - Tabelas aninhadas: o uso dessas tabelas não é recomendado, pois não funcionam bem em telas de tamanho limitado.
  - Leiaute com tabelas: assim como o uso de tabelas aninhadas não é recomendado, deve-se evitar o uso de tabelas para definir leiautes de telas.
  - Imagens de espaçamento: não é recomendado o uso de imagens de espaçamento, pois podem não funcionar em alguns dispositivos.
  - *Frames*: seu uso não é recomendado pela incompatibilidade com diversos dispositivos.
  - Imagens mapeadas: devem ser evitadas, a menos que se tenha certeza de que o dispositivo destinado à aplicação as interpreta bem.
- Seja prudente com a limitação dos dispositivos: esta recomendação tem por objetivo salientar aos desenvolvedores a importância de levar em conta, na hora da implementação, a limitação dos dispositivos. O W3C apresenta exemplos de itens relacionados à *Web*, como *cookies*, suporte a folhas de estilos (CSS), etc. Abaixo destacamos alguns itens:
    - Suporte à tabela: devem ser utilizadas somente se o dispositivo suportar esse recurso. E, sendo utilizadas, devem-se utilizar alternativas à apresentação tabular.
    - Tipografia: relacionado ao tipo de letra utilizado na aplicação. Segundo a cartilha do W3C, não se deve confiar na disponibilidade de um tipo de fonte específico em todos os dispositivos.
    - Uso de cores: é recomendado que as informações visualizadas em cores possam ser vistas sem cor.
  - Aperfeiçoe a navegação: trata da simplicidade na navegação e no uso de teclado, por exemplo. É importante ressaltar que um grande desafio deste projeto foi propor uma maneira de organizar todos os recursos necessários à escrita de



sinais, o que inclui questões de navegação entre as diferentes opções de símbolos e telas, por exemplo. Como subitens, citamos:

- Barra de navegação: o documento sugere que a navegação deve estar disposta na parte superior da tela, reduzindo-a ao mínimo possível.
  - Navegação: devem ser utilizados critérios de navegação consistentes.
  - Tecla de acesso rápido: seu uso facilita o acesso aos itens de menus de navegação e funções mais utilizadas.
- Teste antes os gráficos e cores: este item traz a preocupação com as questões gráficas relacionadas ao contraste das cores de fundo com os elementos de primeiro plano, também ressalta a questão do tamanho das imagens e gráficos.

Como subitens estão incluídos:

- Redimensione as imagens: sempre que possível, redimensionar as imagens de tamanho intrínseco no servidor de dados (ex. *bitmap*).
- Gráficos grandes: imagens com tamanho maior que o tamanho da tela ou com alta resolução devem ser evitadas, a menos que haja perda de informação relevante do conteúdo apresentado.
- Tamanho de imagem específico: é recomendado informar o tamanho da imagem na marcação se ela possuir um tamanho específico.
- Alternativas aos elementos não-textuais: forneça um equivalente em forma de texto para cada elemento não textual.
- Contraste de cores: certifique-se que haja contraste suficiente entre cor de fundo e do primeiro plano.
- Legibilidade da imagem de fundo: ao utilizar imagens como plano de fundo, deve-se garantir que o conteúdo continua sendo legível no dispositivo correspondente.
- Medidas: os atributos das linguagens de marcação e os valores das propriedades das folhas de estilo não devem ser medidos em *pixels* nem em unidades absolutas.

- Faça um tamanho reduzido: o tamanho do conteúdo deve ser levado em conta ao se desenvolver uma aplicação devido à limitação do tamanho da tela nos dispositivos móveis em geral. A cartilha do W3C propõe que o conteúdo seja criado de forma reduzida, sempre que possível, a fim de facilitar a navegação do usuário na aplicação. Quando não for possível utilizar conteúdos de forma reduzida, o recurso de rolagem deve ser utilizado em apenas um sentido da tela, evitando sempre que possível uma rolagem secundária.
  - Minimizar: o documento recomenda a utilização de marcações concisas e eficazes.
  - Limite no tamanho da página: certificar-se de que o tamanho da página é adequado para as limitações de memória do dispositivo.
  - Rolagem de tela: conforme já citado, a rolagem secundária não é recomendada. Sempre que possível adequar o conteúdo ao tamanho da tela do dispositivo, e quando não possível, limitar a rolagem em apenas um sentido.
- Economize o uso da rede: traz dicas com relação a itens específicos de navegação, como atualização automática da página, utilização de protocolos que diminuam o tempo de espera do usuário, cachê, etc.
- Facilite a entrada de dados: define que o uso de entrada de dados na aplicação seja minimizado quando esta se destina a recursos *Web*. Nosso projeto tem como funcionalidade principal o envio de mensagens escritas no sistema de língua de escrita de sinais *SignWriting* entre dispositivos móveis. São subitens:
  - Uso do teclado: é recomendado que seja minimizado o uso do teclado para entrada de dados das aplicações *Web*.
  - Valores pré-definidos: valores pré-definidos facilitam a navegação e sempre que possível devem ser apresentados aos usuários.
  - Modo de entrada de dados pré-definidos: se o dispositivo for compatível, prefira usar modo de entrada de texto, linguagem e/ou formato de entrada pré-definidos (*defaults*).

- Ordem de tabulação: criar uma ordem lógica no uso de objetos.
- Pense nos usuários da *Web* móvel: considera que os usuários de aplicações móveis dispõem de pouco tempo e por vezes fazem várias atividades ao mesmo tempo. Por isso, a cartilha sugere que sejam considerados os seguintes subitens:
  - Clareza: utilizar linguagem simples e clara.
  - Conteúdo relevante: o conteúdo relevante da página deve aparecer na frente do conteúdo secundário.
  - Conteúdo limitado: fornecer somente o conteúdo que for solicitado pelo usuário.
  - Adequação: certificar que o conteúdo seja adequado para ser usado em dispositivos móveis.
  - Tamanho de página utilizável: dividir as páginas em seções utilizáveis de tamanho limitado.

Os itens da cartilha “Boas práticas em Web Móvel” (W3C Mobile, 2012) serviram de guia para o desenvolvimento da aplicação *SignSMS*, bem como outros documentos que trataram desse tema. Assim, cabe salientar que a cartilha traz recomendações para o desenvolvimento de aplicações com serviços integrados com a internet (aplicações *web*) nas quais a navegação do usuário pelo conteúdo *web* apresentado se faz necessária. Para o desenvolvimento do *SignSMS* adotamos as práticas pertinentes ao escopo do projeto e os detalhes referentes aos itens utilizados estão descritos na seção 4.1.1, assim, neste momento, citamos apenas os critérios que foram considerados na modelagem do projeto *SignSMS*:

- Faça seu projeto para uma *Web* única
  - Capacidades
  - Problemas na implementação
  - Testes
- Confie nos padrões *Web*

- Compatibilidade de formato
- Seja prudente com a limitação dos dispositivos
  - Tipografia
  - Uso de cores
- Aperfeiçoe a navegação
  - Barra de navegação
  - Navegação
- Teste antes os gráficos e cores
  - Contraste de cores
- Faça um tamanho reduzido
  - Rolagem de tela
- Pense nos usuários da *Web* móvel
  - Clareza

## **2.5. Padrões e princípios de design**

Barbosa & da Silva (2010) destacam a importância de haver padrões de *design* (*design patterns*), que são descrições de melhores práticas num determinado domínio de *design*, por isso, capturam a sabedoria coletiva de *designers* experientes em IHC para a proposição de boas soluções a problemas específicos. Padrões de *design* não são descrições passo a passo sobre como projetar uma interface inteira, mas descrições para problemas pontuais. Ainda, conforme sintetiza Silveira & Campos (2012):

- Cada padrão pode ser descrito em maior ou menor nível de detalhes e pode ser adequado a apenas um contexto.
- Os padrões não são isolados.
- Os padrões se relacionam com outros padrões de diferentes maneiras.
- Devem ser analisados cuidadosamente e adaptados a cada projeto.

- Descrição dos padrões pode ser diferenciada. Mas, geralmente, possui um nome, uma descrição da solução que indica uma breve descrição do problema que atende, com exemplos de imagens.

Há diferentes padrões de *design*, cada qual com suas características e um contexto recomendado para uso. A seguir, enfatizamos padrões que se destacam na implementação de aplicações para dispositivos móveis, mais especificamente sobre o desenvolvimento de aplicativos para a plataforma *Android*.

Ainda, ao se tratar da plataforma *Android*, não há padrões impostos, mas, sim, princípios de *design* a serem considerados na implementação. De acordo com *Android Developers* (2012a), existem três metas principais que podem ser consideradas no desenvolvimento de aplicações para dispositivos móveis, que podem ser organizadas em outras recomendações, conforme segue.

- Encante-me (*Enchant Me*): traz dicas sobre os efeitos visuais da aplicação. Inclui:
  - Delicie-me de maneiras surpreendentes (*Delight me in surprising ways*): o desenvolvimento de uma interface bonita, com elementos animados, cuidadosamente posicionados na tela, e/ou efeitos sonoros, podem trazer sensação de bem estar contribuindo para tornar a experiência de uso do sistema mais agradável.
  - Os objetos reais são mais divertidos do que botões e menus (*Real objects are more fun than buttons and menus*): permitir que os usuários manipulem os objetos em suas aplicações reduz o esforço cognitivo necessário para realizar uma tarefa ao mesmo tempo em que torna o uso emocionalmente mais satisfatório do que trabalhar somente com textos.
  - Deixe-me fazer isso, é meu (*Let me make it mine*): permitir que o usuário personalize seus toques pessoais, padrões de tela e outras personalizações opcionais que não prejudiquem as tarefas primárias da aplicação.
  - Conheça-me (*Get to know me*): saber reconhecer as preferências dos usuários ao longo do tempo, por exemplo, tornando fácil o acesso às opções mais utilizadas pelo usuário.

- Simplifique minha vida (*Simplify My Life*): refere-se à praticidade das ações do usuário no ambiente e *feedbacks* do sistema.
  - Mantenha *feedback* curto (*Keep it brief*): usar frases curtas com palavras simples, pois as pessoas tendem a ignorar frases muito longas.
  - Fotos são mais rápidas do que palavras (*Pictures are faster than words*): considerar o uso de imagens para expressar idéias. Por se tratar de representações gráficas, estas ganham destaque na interface e atenção das pessoas podendo ser muito mais eficientes do que palavras. Vem ao encontro das justificativas já apresentadas no tópico “Os objetos reais são mais divertidos do que botões e menus”.
  - Decida por mim, mas deixe-me ter a palavra final (*Decide for me but let me have the final say*): a aplicação pode tentar antecipar os desejos do usuário e agir por conta própria ao invés de solicitar permissão do usuário em cada ação, pois muitas escolhas e decisões podem deixar as pessoas descontentes. No caso de o sistema errar, permita a ação de “desfazer”. Vem ao encontro das justificativas já apresentadas no tópico “Deixe-me fazer isso, é meu”.
  - Apenas mostrar o que eu preciso quando eu precisar (*Only show what I need when I need it*): muita informação pode poluir a interface e sobrecarregar os usuários de dados indesejados. Uma opção é quebrar as tarefas e informações em pequenos pedaços “digeríveis”, assim como ocultar as opções que não são essenciais no momento e/ou sugerir ao usuário qual rumo tomar.
  - Eu deveria sempre saber onde estou (*I should always know where I am*): dar aos usuários recursos para que eles possam saber o caminho de volta. Para isto, desenvolver opções nos aplicativos com aparências distintas e usar transições para mostrar as relações entre as telas. Fornecer *feedback* sobre as tarefas em andamento também ajuda o usuário e ter controle da navegação.

- Nunca perder as minhas coisas (*Never lose my stuff*): salvar os documentos que os usuários gastaram tempo para criar e permitir que eles acessem de qualquer lugar (celular, *internet*, *tablet*, etc). Armazenar as configurações, toques pessoais, e documentos criados permite que o processo de atualização se torne mais fácil devido ao armazenamento prévio das configurações e arquivos dos usuários. Vem ao encontro das justificativas já apresentadas no tópico “Conheça-me”.
- Se parece o mesmo, deve agir da mesma forma (*If it looks the same, it should act the same*): tornar as diferenças funcionais visualmente distintas. Evite opções que parecem semelhantes, mas que agem de forma diferente sobre a mesma entrada.
- Apenas me interrompa se for importante (*Only interrupt me if it's important*): a aplicação deve evitar interromper a navegação dos usuários se o *feedback* ao usuário é irrelevante. As pessoas querem manter o foco, se a interrupção não for resultante de atividade crítica e/ou sensível ao tempo, pode ser cansativa e frustrante.
- Faça-me sentir surpreendente (*Make Me Amazing*): refere-se ao modo como o usuário aprende a utilizar um aplicativo, desde o aprendizado intuitivo às mensagens de *feedback* para resolução de erros retornados pelo sistema.
  - Disponibilize dicas/funcionalidades que funcionam em todos os lugares (*Give me tricks that work everywhere*): tornar o aplicativo mais fácil de aprender aproveitando os padrões visuais e de memória muscular (referente às ações físicas do usuário para determinada ação/gesto) de outros aplicativos *Android*.
  - Não é minha culpa (*It's not my fault*): se algo der errado, dar ao usuário instruções claras e cordiais de recuperação, mas poupá-los dos detalhes técnicos. É preferível que a aplicação possa corrigir o erro em segundo plano. Vem ao encontro das justificativas já apresentadas nos tópicos “Mantenha *feedback* curto” e “Apenas mostrar o que eu preciso quando eu precisar”.

- Espalhe incentivos (*Sprinkle encouragement*): permitir a quebra de tarefas complexas em pequenos passos que podem ser facilmente realizados. Dar *feedback* sobre as ações, mesmo que seja apenas um brilho sutil. Vem ao encontro das justificativas já apresentadas no tópico “Apenas mostrar o que eu preciso quando eu precisar”.
- Faça o trabalho pesado para mim (*Do the heavy lifting for me*): fazer com que usuários menos experientes possam usufruir de recursos mais avançados de maneira prática, permitindo-lhes fazer coisas que nunca pensaram que pudessem fazer.
- Faça as coisas importantes com rapidez (*Make important things fast*): nem todas as ações são iguais. O desenvolvedor deve decidir o que é mais importante no aplicativo e tornar as funções fáceis de encontrar e rápidas de usar, como o botão de disparo de uma câmera, ou o botão de pausa em um leitor de música. Vem ao encontro das justificativas já apresentadas no tópico “Faça o trabalho pesado para mim”.

Também destacamos cinco padrões de projeto para a interface das aplicações para *Android*, estes diferentemente dos padrões de projeto, referem-se a um conjunto de boas práticas já conhecidas e amplamente aplicadas que visam aprimorar a experiência do usuário com a interface gráfica (C.E.S.A.R., 2011). São eles:

- *Action Bar*: é uma barra com um conjunto de funcionalidades básicas e de auxílio à navegação persistente na interface das aplicações *Android*. De forma geral, esta barra reúne as principais funcionalidades em um local de acesso rápido e previsível. A Figura 12 e Figura 13 ilustram exemplos de uso de *Action Bar*.



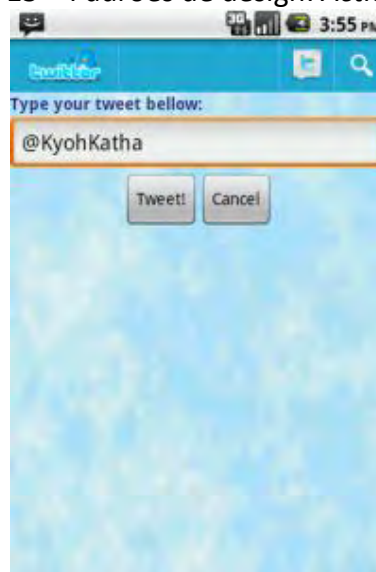
Figura 12 – Padrões de design: *Action Bar 1*



1. *Action Bar* (barra de ação principal)
2. *Top Bar* (barra superior)
3. *Bottom Bar* (barra inferior)

Fonte: Android Developer (2012)

Figura 13 – Padrões de design: *Action Bar 2*

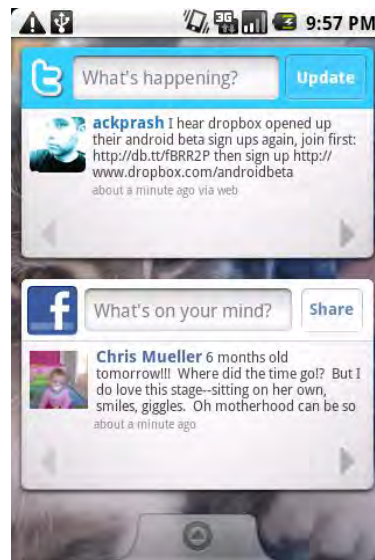


Fonte: C.E.S.A.R (2011)

- *Companion Widgets*: é um componente que reúne as principais funcionalidades de uma aplicação, geralmente centralizados na tela inicial, possibilitando ao usuário acessá-las sem precisar lançar toda a aplicação (C.E.S.A.R. 2011), ou seja, uma versão sintetizada da original, que pode se comunicar com a versão completa, quando preciso. Por concentrar as funcionalidades, este componente pode acelerar a interação e facilitar a usabilidade do usuário com os recursos

disponíveis. A Figura 14 ilustra o uso de *Companion Widgets* nas barras de busca dos aplicativos *Twitter* e *Facebook*.

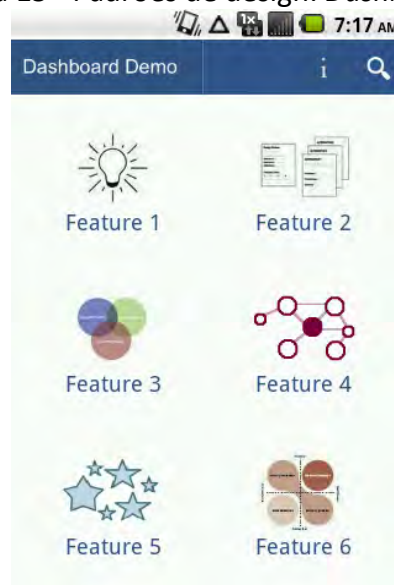
Figura 14 - Padrões de design: *Companion Widgets*



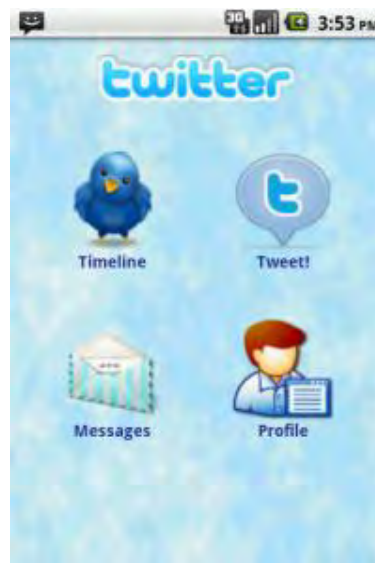
Fonte: C.E.S.A.R. (2011)

- *Dashboard*: é utilizado para a criação da tela inicial das aplicações *Android*, nesta tela pode haver conteúdo dinâmico ou estático e também as principais opções da aplicação, facilitando a navegação do usuário. Na Figura 15 e Figura 16 pode-se observar as diversas funcionalidades de um exemplo de aplicação do *Dashboard*.

Figura 15 - Padrões de design: *Dashboard 1*

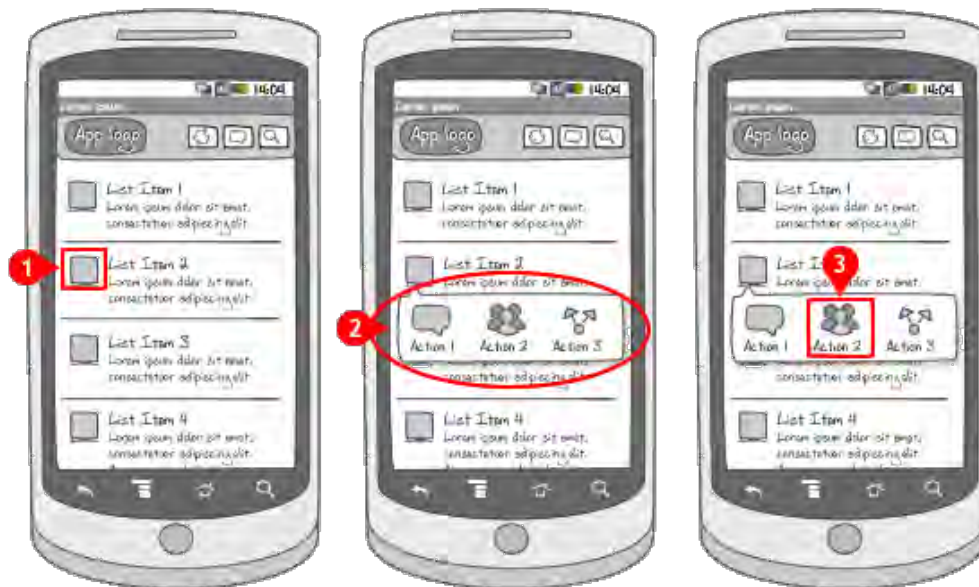


Fonte: Lahti's (2011)

Figura 16 - Padrões de design: *Dashboard 2*

Fonte: C.E.S.A.R. (2011)

- *Quick Actions*: são utilizadas para ações rápidas de um elemento na tela. Para usar uma ação rápida o usuário deve interagir com um elemento com esta opção habilitada e escolher a ação desejada. C.E.S.A.R. (2011) cita como exemplo uma lista de contatos, que ao ser selecionado um determinado contato podem ser exibidas opções relacionadas ao elemento selecionado, tais como enviar uma mensagem, remover contato, editar suas informações, entre outras. A Figura 17 e Figura 18 ilustram a interação do usuário para acessar a janela de opções rápidas de um elemento.

Figura 17 - Padrões de design: *Quick Actions* 1

Fonte: Android Patterns (2011)

Figura 18 - Padrões de design: *Quick Actions* 2

Fonte: C.E.S.A.R. (2011)

- **Search Bar:** é um padrão aplicado para buscas rápidas por toda aplicação. Por ser uma ação de uso frequente, este mecanismo é alocado na *Action Bar*, conforme ilustrado na Figura 19. Ainda, C.E.S.A.R. (2011) explica que um recurso interessante deste padrão é o de dicionário de sugestões, que, a cada letra digitada, verifica se existe alguma palavra chave relacionada.

Figura 19 - Padrões de design: Search Bar



Fonte: C.E.S.A.R. (2011)

Por fim, este estudo permitiu identificar padrões e recomendações pertinentes para a implementação do *SignSMS*. Os padrões listados a seguir estão descritos em conjunto com as práticas e princípios de *design* apresentadas na seção 4.1.1. São eles:

- Encante-me:
  - Delicie-me de maneiras surpreendentes.
  - Os objetos reais são mais divertidos do que botões e menus.
- Simplifique minha vida:
  - Fotos são mais rápidas do que palavras.
  - Eu deveria sempre saber onde estou.

Com relação aos padrões de projeto para interface, dentre os cinco citados nesta mesma seção, apenas os padrões *Dashboard* e *Search Bar* foram utilizados, estes serão abordados na seção da interface. Tecnicamente a escolha da API base do projeto não permite o uso dos padrões *Action Bar*, *Companion Widgets* e *Quick*, pois os mesmos não são suportados na versão da API escolhida para o *SignSMS*.

## 2.6. Tecnologia móvel

As tecnologias móveis nos últimos anos vêm ganhando cada vez mais espaço na preferência dos usuários. De acordo com Moura (2008), tal mudança é decorrente dos avanços na área de tecnologia e de telecomunicações, aliada ao fato de que hoje um aparelho de telefone móvel não serve apenas para fazer ligações telefônicas, ou para manter uma agenda de contatos. Esses aparelhos vêm agregando novas funcionalidades e com um alto poder de processamento.

Lecheta (2010) também destaca que o mercado de celulares está crescendo cada vez mais. A variedade de modelos de dispositivos móveis também é crescente, de forma similar ao número de Sistemas Operacionais que neles são executados, como, por exemplo, o *iPhone OS*, *Windows Mobile*, *BlackBerry*, *Symbian*, *Android*, entre outros. Tal variedade dificulta a escolha do programador pela plataforma em que vai ser desenvolvida sua aplicação.

Para o *SignSMS*, optamos por desenvolver a aplicação para *Android* pelo fato da *Google*, empresa responsável pelo desenvolvimento dessa plataforma, manter o código fonte aberto (*open source*) e, ainda, por disponibilizar um *kit* de desenvolvimento<sup>5</sup> de aplicações em *Java*<sup>6</sup> e um *plugin* para o *Eclipse*. Ainda, a *Google* disponibiliza redes e comunidades para que os desenvolvedores possam trocar ideia entre si sobre a tecnologia do *Android*, *SDK*, desenvolvimento de Aplicativos, dúvidas e tutoriais (Lecheta, 2010). Aliado a esses facilitadores, o *Android* possui uma área dedicada para arquivos locais no próprio dispositivo, o que poderá ser útil ao funcionamento do nosso protótipo.

O *Android* é um ambiente multitarefa, no qual os aplicativos podem ser executados em paralelo. Lançado em 2008 pela *Google* e pela *Open Handset Alliance*, conforme TecMundo (2012), já domina grande parte do mercado no segmento de tecnologias móveis.

O *Android* é um sistema operacional baseado no *Linux*, de forma que seus serviços essenciais como os relacionados à segurança, ao gerenciamento de memória, ao

---

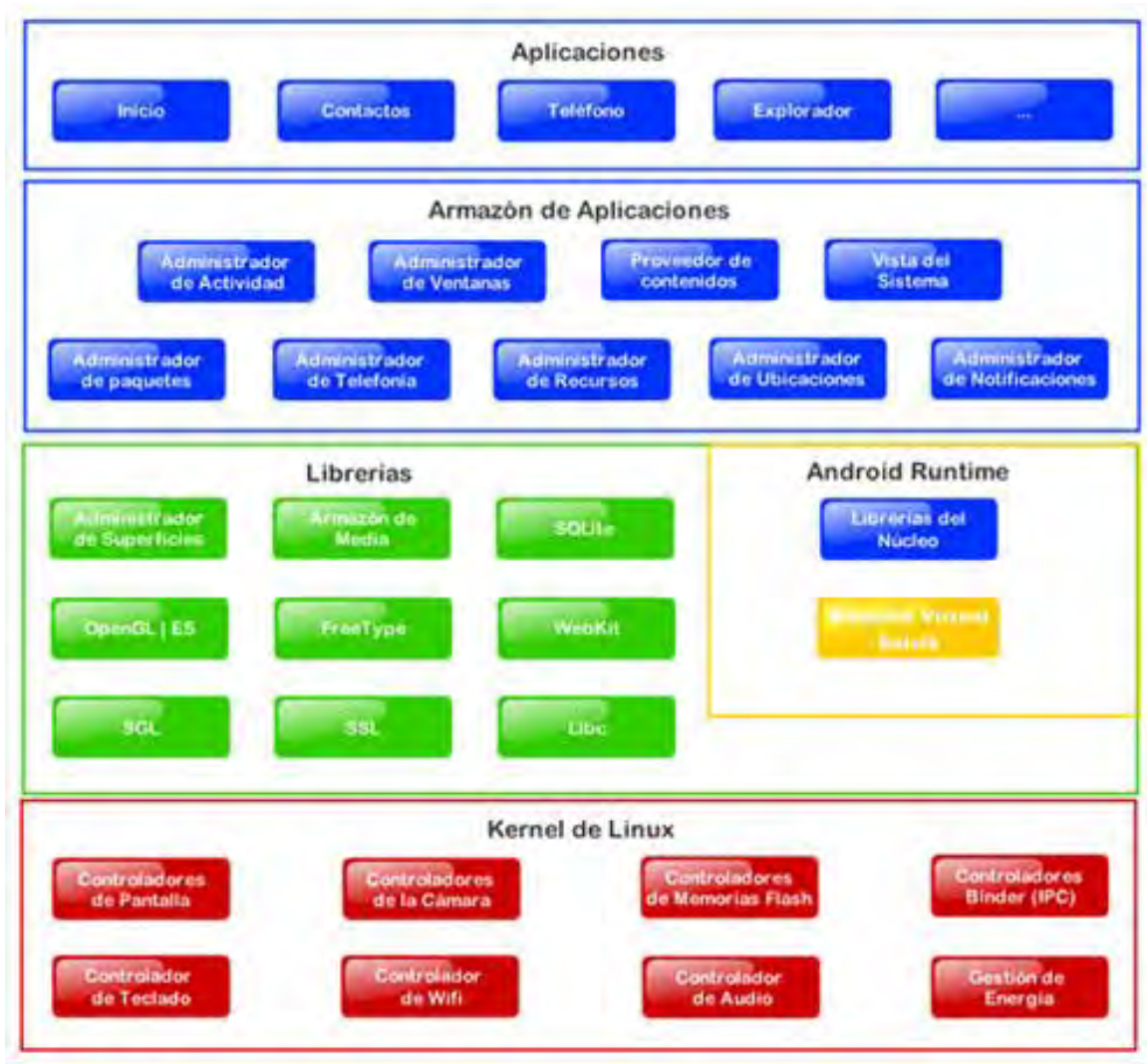
<sup>5</sup> O Kit de desenvolvimento é denominado de Android SDK, e, segundo Alessandro de Oliveira Faria, “proporciona ferramentas e chamadas via API na linguagem JAVA para desenvolvimento de programas. Resumidamente, são programas para celulares com um sistema operacional, middleware e interface.”. Disponível em: [http://www.dicas-l.com.br/arquivo/como\\_instalar\\_o\\_android\\_sdk.php](http://www.dicas-l.com.br/arquivo/como_instalar_o_android_sdk.php). Acesso em: 11 abr. 2012.

<sup>6</sup> A linguagem Java possui vasta documentação e tutoriais disponibilizados em livros e/ou na Internet.



gerenciamento de processos, à rede e aos *drivers* são do *Linux*. O *kernel* do *Linux* também funciona como uma camada de abstração entre o *hardware* do dispositivo e o restante do conjunto de *softwares* que são desenvolvidos em paralelo. A Figura 20 ilustra um esquema da arquitetura do Sistema *Android*.

Figura 20 - Plataforma *Android*



Fonte: Vilchez (2012)

O *Android* já possui aplicativos centrais como um cliente de *e-mail*, programa para SMS, calendário, mapas, navegador e gerenciador de contatos, dentre outras aplicações instaladas que, segundo Lecheta (2010), contribuem para ser a opção de desenvolvedores no que se refere a aplicações para dispositivos móveis.

O sistema possui uma série de bibliotecas em C/C++ usadas por diversos componentes do *Android*, que permitem trabalhar com diferentes formatos de mídia como

*MPEG4*<sup>7</sup>, *MP3*<sup>8</sup>, *AAC*<sup>9</sup>, *JPG*<sup>10</sup>, *PNG*<sup>11</sup> entre outros. Também é possível a exibição de conteúdo tanto em duas dimensões quanto em três dimensões cuja implementação é baseada na *OpenGL*<sup>12</sup>.

Os aplicativos do *Android* executam em uma máquina virtual chamada *Dalvik*, que possui a capacidade de executar múltiplas instâncias, assim cada processo executa em uma instância diferente do *Dalvik*. Os arquivos são criados por um compilador Java, que converte o resultado no formato *.dex*, que é formato dos executáveis do *Dalvik*.

### 2.6.1. Recursos de interação

Quanto aos recursos de interação usuário-sistema, focaremos em ações que se relacionam ao recurso de *touch screen*, na qual a interação com o dispositivo é feita através de uma ação gestual. As variações dessas ações caracterizam as formas com que o *Android* interpreta a interação do usuário com o dispositivo. A seguir, apresentamos uma breve descrição das principais formas de interação gestual do usuário com os dispositivos que apresentam tais recursos no *Android*, bem como as funcionalidades mais comuns associadas a cada ação (Android Developer, 2012b).

- Toque: essas ações têm por finalidade ativar a funcionalidade principal para um determinado item. A ação gestual associada é pressionar e soltar o item, conforme Figura 21.

---

<sup>7</sup> *MPEG4* significa *Motion Picture Experts Group*.

<sup>8</sup> *MP3* significa *MPEG Audio Layer-3*.

<sup>9</sup> *AAC* significa *Advanced Audio Coding*.

<sup>10</sup> *JPG* ou *JPEG* significa *Joint Pictures Expert Group*.

<sup>11</sup> *PNG* significa *Portable Network Graphics*.

<sup>12</sup> *OPENGL* significa *Open Graphics Library*.



Figura 21 - *Touch screen*: interação por meio do toque



Fonte: Android Developers (2012b)

- Pressão longa: ativa o modo de seleção de dados, permitindo ao usuário selecionar um ou mais itens em uma exibição, e agir de acordo com os dados usando uma barra de ação contextual. A ação gestual inclui pressionar, esperar alguns instantes e soltar o item, conforme ilustrado na Figura 22.

Figura 22 - *Touch screen*: interação por meio de pressão longa



Fonte: Android Developers (2012b)

- Swipe: recurso que permite a rolagem do conteúdo horizontalmente ou navegação entre as telas da aplicação. A ação gestual do usuário inclui

pressionar, mover para a esquerda ou direita e soltar a tela. A Figura 23 representa essa ação.

Figura 23 - *Touch screen*: interação por meio de *Swipe*



Fonte: Android Developers (2012b)

- Arrastar: reorganiza os dados em uma tela ou move dados de um contêiner para outro (ex: mover pastas na Tela Principal). A ação gestual inclui pressionar longamente o item, mover e solta-lo. A Figura 24 ilustra essa ação.

Figura 24 - *Touch screen*: interação por meio da ação de arrastar



Fonte: Android Developers (2012b)

- Toque duplo: este recurso permite ampliar o conteúdo de uma janela e também pode ser usado como ação secundária de seleção de texto. A ação gestual associada é dois toques rápidos na tela/texto. A Figura 25 ilustra esta opção.

Figura 25 - *Touch screen*: interação por meio do toque duplo



Fonte: Android Developers (2012b)

- Movimento de pinça para fora (*Pinch open*): este recurso faz zoom para aproximar o conteúdo da tela. A ação gestual inclui pressionar com dois dedos a tela, fazer o movimento dos dedos para fora e soltar a tela, conforme ilustra a Figura 26.

Figura 26 - *Touch screen*: interação por meio do movimento de pinça para fora



Fonte: Android Developers (2012b)

- Movimento de pinça para dentro (*Pinch close*): faz *zoom* para distanciar o conteúdo da tela. A ação gestual do usuário inclui pressionar com dois dedos a tela, fazer o movimento dos dedos para dentro e soltar a tela, como é mostrado na Figura 27.

Figura 27 - *Touch screen*: interação por meio do movimento de pinça para dentro



Fonte: Android Developers (2012b)

Dos recursos de interação mencionados neste capítulo, o *SignSMS* utiliza os recursos básicos “Toque” e “Arrastar”, que são descritos na seção 4.3, que descreve as funcionalidades da aplicação por meio da interface do *SignSMS*.

### 3. APLICAÇÕES RELACIONADAS

Neste capítulo apresentamos um estudo sobre aplicações computacionais que fazem uso de língua de sinais e do sistema de escrita *SignWriting*, que é apresentado na seção 3.1.

Também são apresentados exemplos de aplicações interativas para dispositivos móveis, que façam ou não uso de língua de sinais, mas que tratem de edição, que foram analisados para compor as características do *SignSMS*. Esses exemplos estão descritos na seção 3.2.

#### 3.1. Aplicações com suporte à Língua de Sinais

Existem variadas aplicações computacionais com suporte à língua de sinais. Campos (2001), Bohrz (2011), por exemplo, descrevem aplicações que representam a LS por meio do alfabeto manual, desenho dos sinais em duas ou três dimensões, vídeo, avatares e, o que mais interessa ao escopo deste trabalho, por meios da escrita de sinais. Para a realização deste TCC, analisamos aplicativos para desktop e *web* porque utilizam a escrita de sinais, e aplicativos baseados em tecnologia móvel, mesmo que representem a LS por desenho das mãos, vídeo, avatar, conforme segue.

- Aplicações para *desktop*:
  - Pacote *Sign*
  - *Sw-Edit*
- Aplicação para *Web*:
  - *Sign WebMessage*
- Aplicações para *mobile*:
  - *Sign 4 Me*
  - *SkySigns ASL Dictionary*
  - *SMS to Sign Language for Mobile*
  - *Signing Savvy*
  - *Baby Sign Language*

- *ASL American Sign Language*

### 3.1.1. Pacote *Sign*

O Pacote *Sign* foi desenvolvido no âmbito da Faculdade de Informática da PUCRS sob orientação da Profa. Márcia de Borba Campos. Todas as aplicações do Pacote *Sign* começam com o termo “*Sign*” para fazer referência ao sistema de escrita de sinais utilizado, que foi o *SignWriting*. Por meio desse sistema, é possível escrever todos os componentes de um sinal, quer sejam de expressão fácil, mão, toque, movimento e ponto de articulação, conforme já descrito anteriormente.

Fazem parte do Pacote *Sign* os seguintes softwares, cuja explicação tem como fonte Campos (2001, 2004, 2006, 2007):


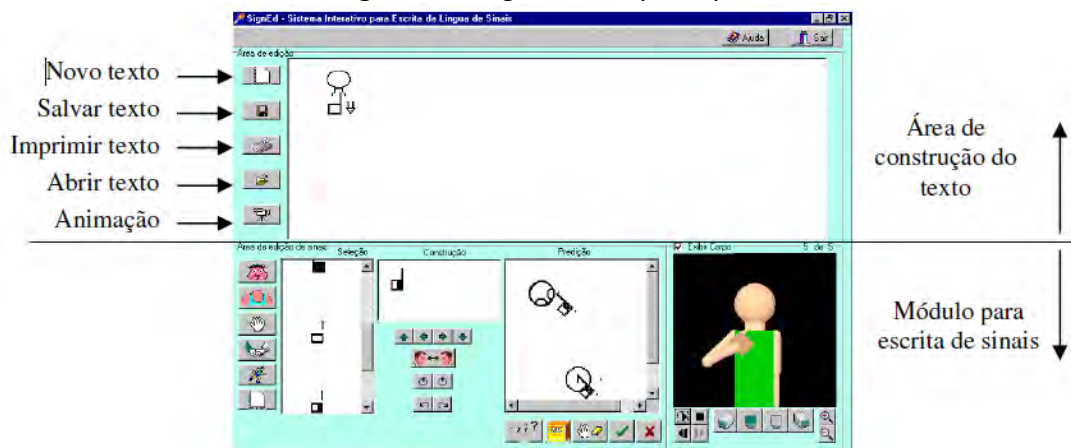
- *SignEd*: é o editor de escrita de língua de sinais do pacote, que permite representar os elementos que compõem o sinal por meio do *SignWriting*. Os sinais construídos no *SignEd* são armazenados em uma base que possui a escrita do sinal e as configurações para a representação gráfica do mesmo por meio de um avatar. Para escrever um sinal, o editor organiza os símbolos conforme os grupos do *Signwriting*, organizando o menu em expressão facial, ombro, mãos, toques e movimento. Possui, ainda, um mecanismo de predição de sinal de forma que, ao ser escrito um elemento, o editor mostra todos os sinais que o contém. Desta forma, minimiza o processo de escrita por parte do usuário e auxilia na padronização da escrita do mesmo. Também é possível escrever utilizando o alfabeto manual. O *SignEd* não possibilita ao usuário a escrita de sinais diretamente na área de construção do texto, pois, para inserir um sinal no texto, o usuário deve clicar no botão . A interface da ferramenta pode ser vista na Figura 28. A escrita do texto em sinais ocorre na horizontal, da esquerda para a direita.

Figura 28 - *SignEd*: tela principal

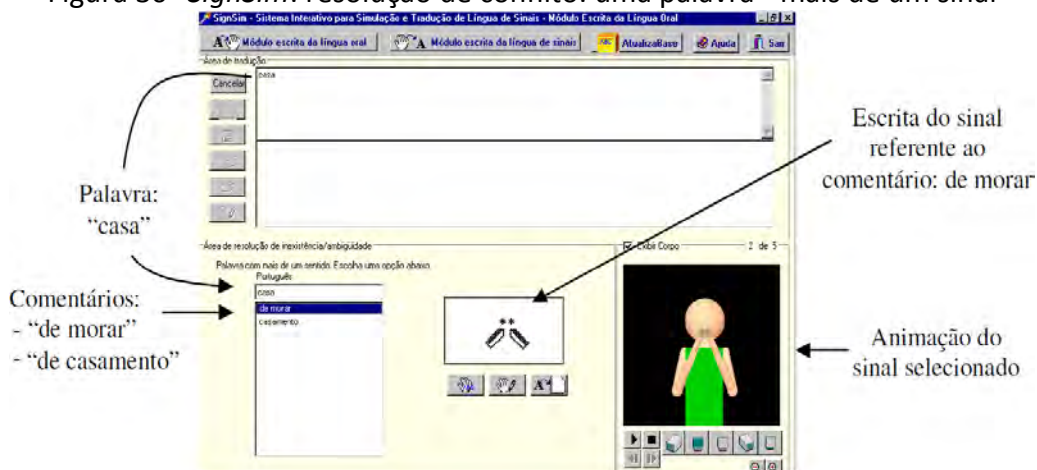
Fonte: Campos (2001, p. 77)

- *SignSim*: basicamente, *SignSim* é um tradutor semi-automático de língua de sinais para escrita de língua oral, e vice-versa. Campos (2001) explica que se faz necessária a intervenção do usuário quando, durante o processo de tradução, for encontrado mais de um sinal para uma mesma palavra, ou, mais de uma palavra para um mesmo sinal, ou ainda, nenhum sinal para uma palavra, ou, nenhuma palavra para um determinado sinal. A escrita do sinal é feita em um módulo similar ao do *SignEd*, sendo que para todo sinal também se faz necessário especificar sua representação espacial. A Figura 29 mostra a interface do *SignSim* utilizando o Módulo Português escrito, já a Figura 30 mostra um exemplo de conflito de uma palavra com mais de um sinal.

Figura 29 - *SignSim*: módulo Português escrito

Fonte: Campos (2001, p. 79)

Figura 30 - *SignSim*: resolução de conflito: uma palavra - mais de um sinal



Fonte: Campos (2001, p. 79)

- SignTalk*: é uma ferramenta de chat que permite a comunicação síncrona entre seus usuários, seja através da LIBRAS ou do Português escrito. Para tanto, é composto pela aplicação *SIGNTALK* Servidor, que é responsável por disponibilizar as salas *on-line* para o bate-papo, e pela aplicação *SIGNTALK* Cliente, que é por meio da qual os usuários participam do chat, selecionam a sala *on-line* que desejam participar, escolhem se a mensagem é para ser enviada a todos os participantes, ou para alguns em particular, escolhem se desejam receber as mensagens em LIBRAS, em Português escrito ou em ambas as línguas, bem como qual a língua que irão utilizar para escrever suas mensagens. Caso deseje escrever através da LIBRAS, é ativado um módulo para escrita de sinais e para a construção de novos sinais similar ao do *SignED*. A Figura 31 e Figura 32 mostram, respectivamente, a interface do modo *SignTalk* Cliente enquanto a Figura 33 ilustra a interface do modo Servidor.



Figura 31 - SignTalk Cliente: interação através da escrita da LIBRAS

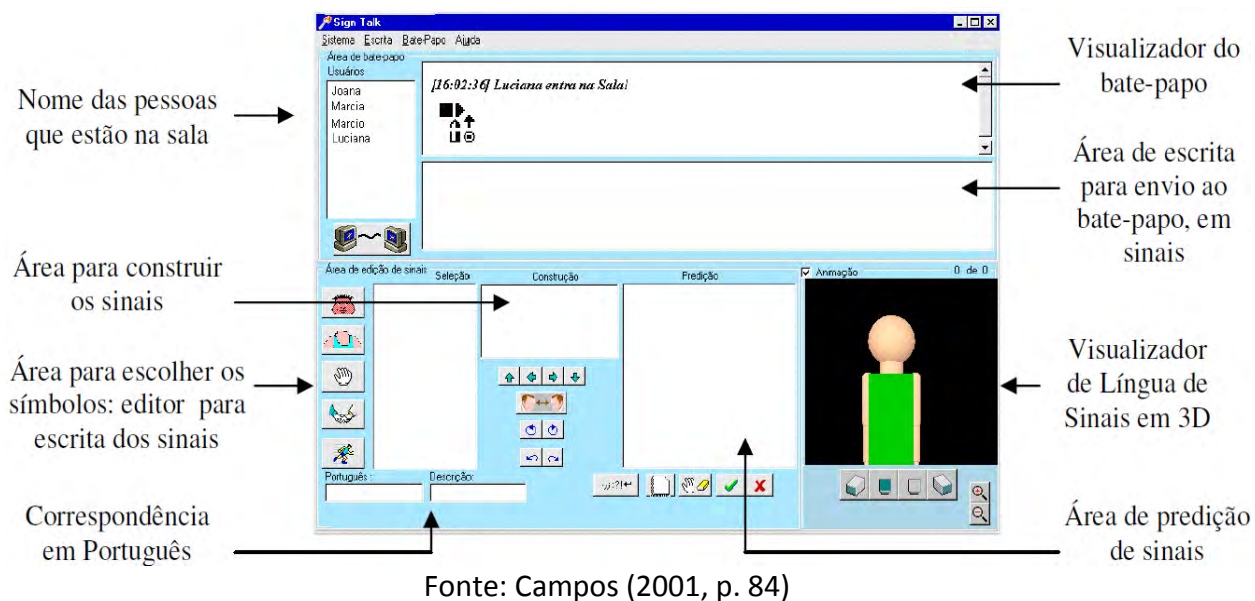


Figura 32 - SignTalk Cliente: interação através do Português escrito

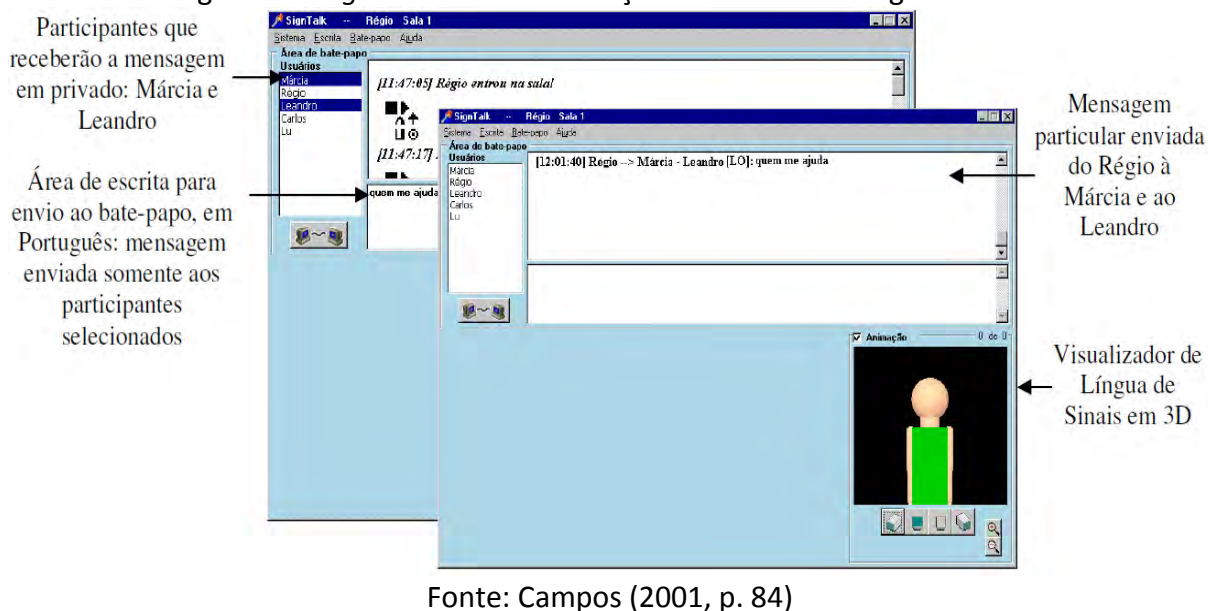
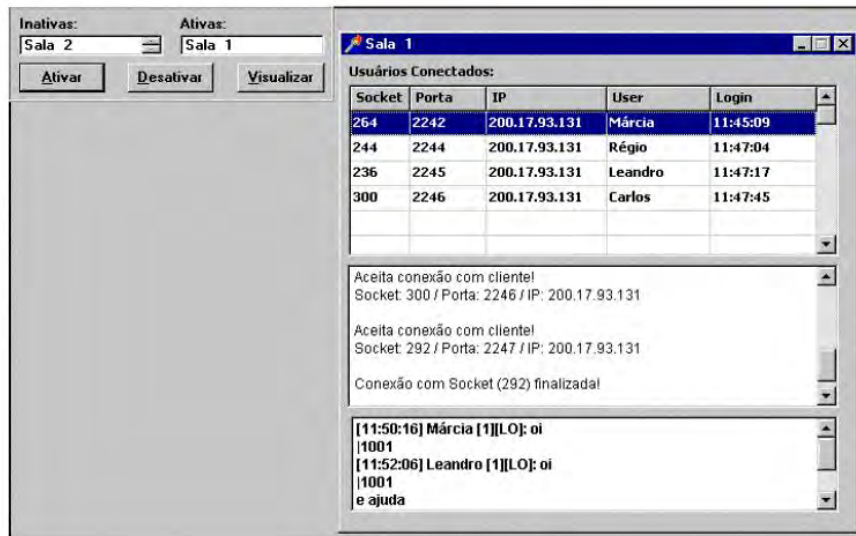


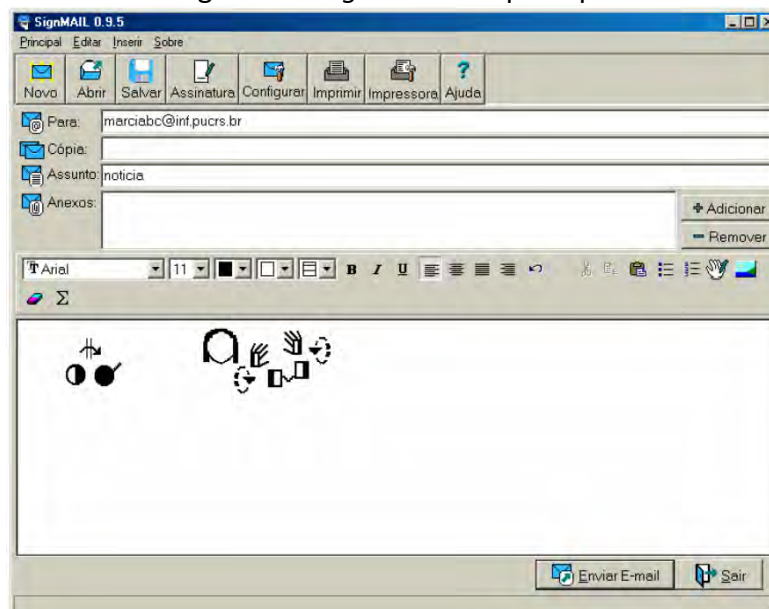
Figura 33 - *SignTalk* Servidor: informações de conexão à Sala 1



Fonte: Campos (2001, p. 86)

- *SignMail*: é uma ferramenta para envio de e-mail escrito em língua de sinais. Para tanto, utiliza um módulo de escrita de sinais, similar ao apresentado para o *SignEd*. O usuário recebe a mensagem em sua ferramenta habitual, sendo que a mensagem em escrita de sinais é encaminhada como um arquivo em anexo. A Figura 34 mostra a tela de escrita do *SignMail*.

Figura 34 - *SignMail*: tela principal

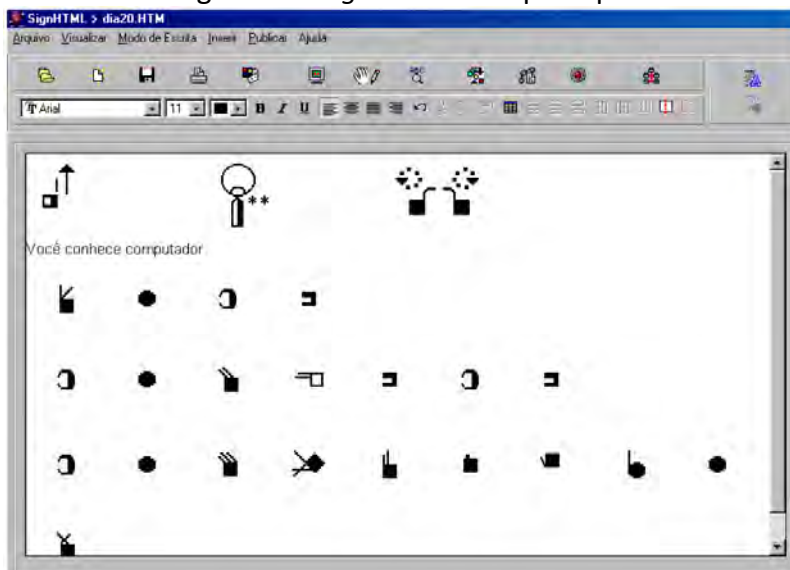


Fonte: Campos (2007)

- *SignHTML*: é um editor HTML para construção de *sites* com suporte à escrita da Língua de Sinais. Assim como as outras ferramentas que compõem o PacoteSign,

também utiliza o módulo de escrita de sinais do *SignEd*. A Figura 35 ilustra a tela do *SignHTML*.

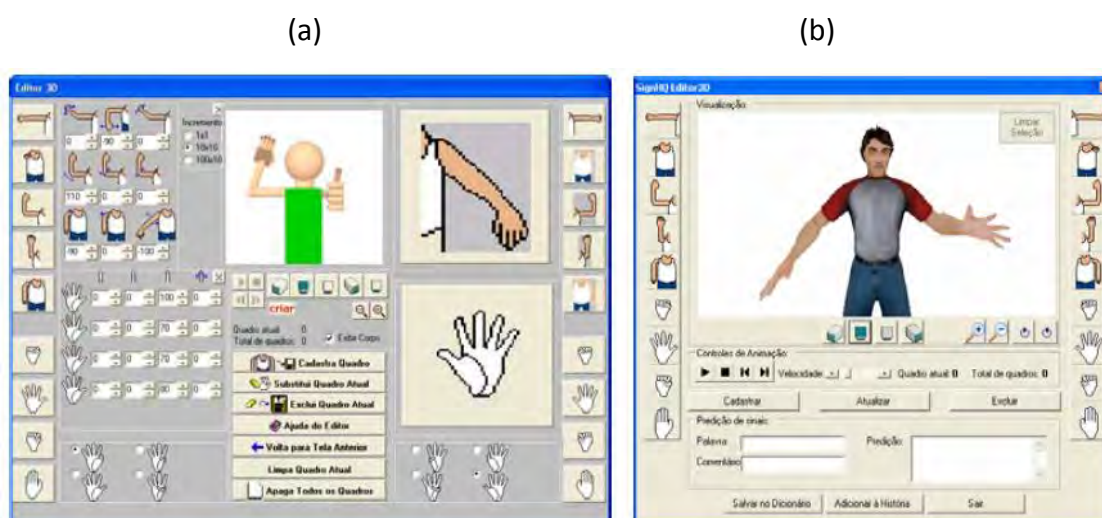
Figura 35 - *SignHTML*: tela principal



Fonte: Campos (2007)

- *SignHQ*: de acordo com Campos et al (2004) é um sistema para a criação e a leitura de histórias em quadrinhos (HQ) com suporte à utilização de Língua de Sinais. Como um editor de HQ computadorizado, permite a inserção de cenários, personagens, objetos, textos e balões de diálogo, bem como a definição de animações para os personagens dos quadros. Neste aplicativo, a criação dos sinais já foi construído de forma mais otimizada do que as aplicações anteriores do Pacote *Sign* no que se à construção da representação espacial do sinal, que é feita por meio de avatar. Assim, o usuário indica o movimento dos dedos, mãos, braços interagindo diretamente com o personagem. A Figura 36 compara a tela do editor 3D do Pacote *Sign* e da ferramenta *SignHQ*.

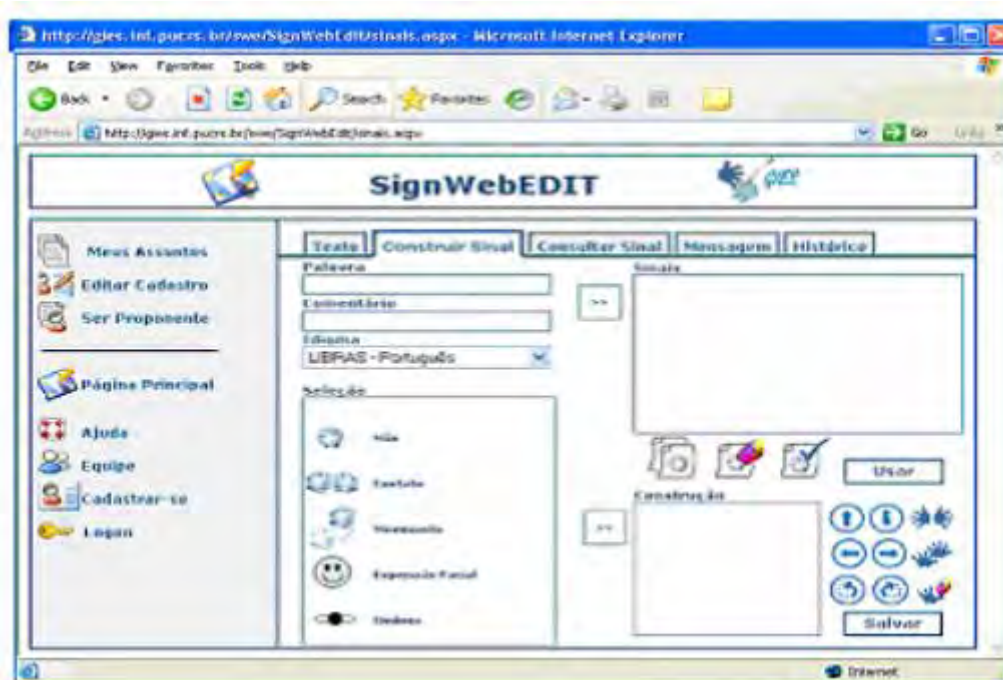
Figura 36 – Editor 3D: Editor do Pacote *Sign* (a) e editor do *SignHq* (b)



Fonte: Campos et al. (2004)

- SignWebEdit: de acordo com Campos et al (2006), é uma ferramenta *on-line* para construção coletiva de textos, que suporta a escrita de sinais. Neste editor foram definidos quatro tipos de usuário: Administrador, Proponente de assuntos, Colaborador e Internauta. O Administrador possui, principalmente, a responsabilidade de controle das configurações do sistema, a validação de sinais, bem como a definição de permissões aos demais tipos de usuários. O Proponente é quem cria uma entrada para um assunto e define quais usuários Colaboradores poderão participar do texto. O Colaborador é aquele que participa do texto escrevendo, excluindo ou alterando trechos de texto. Por fim, existe o Internauta que pode acessar a ajuda online, conhecer a equipe de desenvolvimento do sistema e cadastrar-se para acessar o sistema, não podendo participar da elaboração de textos, enquanto não receber um convite para tal. A Figura 37 mostra a tela principal da aplicação.

Figura 37 - SignWebEdit: tela principal



Fonte: Campos (2006)

Em suma, o processo de escrita de sinais nos aplicativos do Pacote *Sign* ocorre por meio de um editor, que possui opções para expressão facial, ombro, mãos, toque e movimento, cujos símbolos são organizados conforme os grupos do *SignWriting*. A composição dos sinais é realizada em uma área, na qual o usuário pode manipular os símbolos deslocando-os para cima, para baixo, esquerda e direita, pode rotacioná-los de 45° em 45°, ou espelhá-los no eixo horizontal, de acordo com a necessidade de escrita do sinal, por meio de botões. Desta forma, as opções de configurações são minimizadas porque as variações são especificadas pelo usuário no processo de escrita. Ainda, para facilitar esse processo, as ferramentas utilizam o recurso de predição de sinais, que é disponibilizar sinais que utilizam os símbolos informados pelo usuário durante a construção de um sinal, auxiliando na padronização da escrita dos sinais. Após a construção da escrita do sinal, o usuário disponibiliza-o na área de escrita do texto, de forma que a escrita não ocorre diretamente na área do texto. No que se refere ao sentido da escrita, essa ocorre na horizontal, da esquerda para a direita. As ferramentas do Pacote *Sign* foram testadas por Campos (2001) com grupos de usuários ouvintes e surdos. Um comparativo entre as ferramentas pode ser visto na Tabela 3.



Tabela 3 - Comparativo entre as ferramentas do pacote Sign

		<i>SOFTWARE DESENVOLVIDOS</i>		
		<i>SignEd</i>	<i>SignSim</i>	<i>SignTalk</i>
C R I T É R I O S	Categoria	- Editor	- Tradutor	- Chat
	Apresenta os sinais através do alfabeto manual ou na LS	- Alfabeto manual - LS: na LIBRAS	- Alfabeto manual - LS: na LIBRAS	- Alfabeto manual - LS: na LIBRAS
	Forma de apresentação dos sinais	- Escrita ( <i>SignWriting</i> ) - Personagem (“boneco” em 3 dimensões)	- Escrita ( <i>SignWriting</i> ) - Personagem (“boneco” em 3 dimensões)	- Escrita ( <i>SignWriting</i> ) - Personagem (“boneco” em 3 dimensões)
	Possui animação	- Sim	- Sim	- Sim
	Expressa movimento em tempo real, ou através de setas ou através da escrita de sinais	- Em tempo (animação do “boneco”) - Escrita ( <i>SignWriting</i> )	- Em tempo (animação do “boneco”) - Escrita ( <i>SignWriting</i> )	- Em tempo (animação do “boneco”) - Escrita ( <i>SignWriting</i> )
	Utiliza expressão facial	- Sim, através da Escrita ( <i>SignWriting</i> )	- Sim, através da Escrita ( <i>SignWriting</i> )	- Sim, através da Escrita ( <i>SignWriting</i> )
	Permite a construção de novos sinais	- Sim	- Sim	- Sim
	Permite editar/alterar sinais	- Sim	- Sim	- Sim
	Traduz palavras ou textos	- Não	- Sim	- Sim
	Possui consulta aos sinais	- Sim, através da escrita dos sinais (área de predição)	- Sim, através da escrita dos sinais (área de predição) e através do Português	- Sim, através da escrita dos sinais (área de predição) e através do Português
	Permite imprimir as informações	- Sim	- Sim	- Sim
	Salva as ações do usuário	- Sim	- Sim	- Sim
	Forma de interação homem-máquina	- Teclado e <i>mouse</i> (botões e menus)	- Teclado e <i>mouse</i> (botões e menus)	- Teclado e <i>mouse</i> (botões e menus)
Permite interação interpessoal	- Não	- Não	- Sim	

Fonte: Campos (2001)

Dentre os resultados, destacamos as funcionalidades a serem aplicadas e/ou discutidas para a implementação do *SignSMS*:

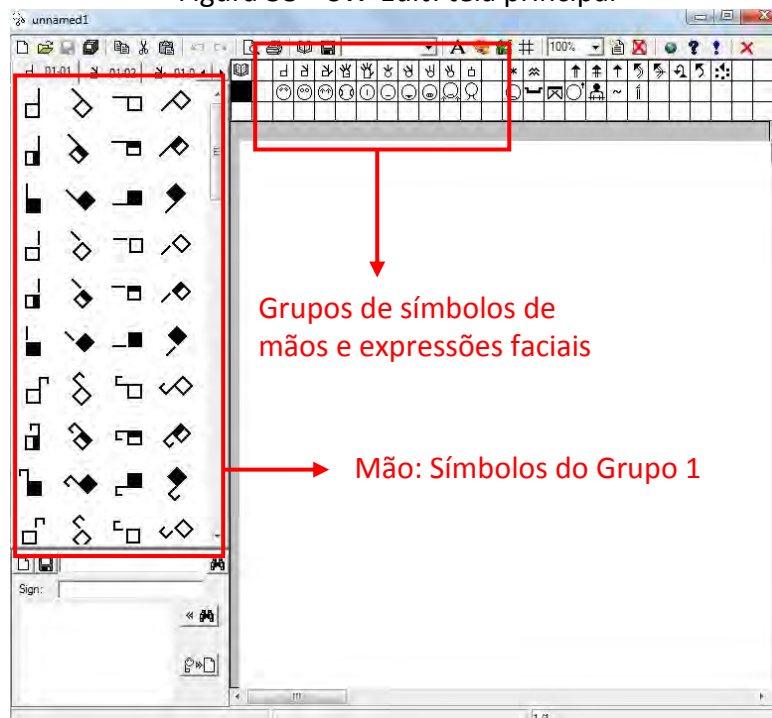
- Permite a escrita por meio do alfabeto romano, alfabeto manual e escrita de sinais.
- Utiliza-se o sistema *SignWriting* de escrita de sinais.
- O menu de escrita de sinais é organizado conforme os grupos de símbolos do *Signwriting*, permitindo que o usuário informe expressão facial, ombro, mãos, toques e movimento.

- O texto escrita em sinais ocorre na horizontal, da esquerda para a direita.
- Permitir a construção/edição/exclusão/consulta de sinais são opções que serão estudadas para serem adicionadas as funcionalidades previstas e serão mais bem discutidas no capítulo correspondente ao projeto do *SignSMS*.

### 3.1.2. *SW-Edit*

O *SW-Edit* é um editor para escrita de línguas de sinais por meio do Sistema *SignWriting* (Costa & Torchelsen, 2012). Nesta aplicação, diferente das ferramentas do Pacote *Sign*, a escrita do sinal ocorre diretamente em uma área principal. Para isso, o menu de símbolos de mãos já disponibiliza modelos com rotação de 45º em 45º, o menu de toques já oferece símbolos repetidos e os espelhamentos nos eixos X e Y são disponibilizados por meio de um menu e aplicados no símbolo previamente selecionado na área principal. A Figura 38 ilustra a tela inicial da aplicação.

Figura 38 – *SW-Edit*: tela principal

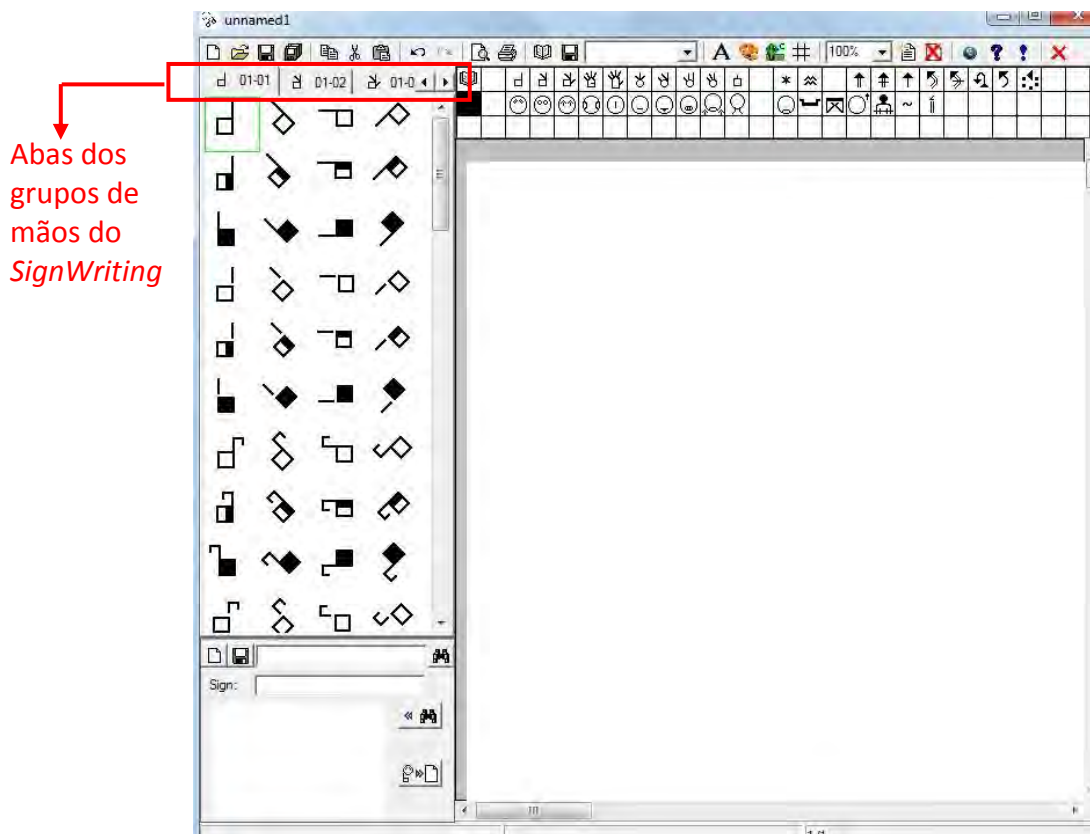


Fonte: *SW-Edit* (2012)

Em termos de interface, o *SW-Edit* destaca-se pela ausência de menus escritos, já que são representados por ícones ou por símbolos do *SignWriting*. As opções dos menus possuem legenda que são disponibilizadas com o passar do *mouse*. Com relação aos símbolos, de forma similar ao Pacote *Sign*, esses são disponibilizados por categorias, que são

organizadas em abas, conforme ilustrado Figura 39. O programa tem seu funcionamento baseado, principalmente, em ações do *mouse*. As ações referentes ao teclado baseiam-se na movimentação dos símbolos inseridos na área de produção do texto (setas direcionais, que podem ser utilizadas para melhor posicionar os símbolos) e para escrever palavras em alfabeto romano, quando a opção “Inserir Caixa de Texto” é selecionada pelo usuário e adicionada ao texto corrente. Para construir uma frase, podem-se incluir os símbolos, um a um, e já organizá-los, ou inserir todos os símbolos na área de texto, e depois organizar um a um, arrastando-os. Também é possível imprimir o texto produzido e adicionar novos sinais ao banco de dados, podendo reutilizá-las sempre que necessário. Para isso, o usuário deve associar um dicionário de símbolos e selecioná-lo sempre que desejar adicionar um novo sinal ao dicionário já existente.

Figura 39 – *SW-Edit*: Abas



Fonte: *SW-Edit* (2012)

O uso do *SW-Edit* é razoavelmente simples, mas para usuários novatos pode apresentar algumas dificuldades. Diferente da avaliação do Pacote *Sign*, optamos por avaliar a interface do aplicativo por meio de inspeção baseada nas heurísticas de Nielsen, que são (NIELSEN, 1994):



1. Visibilidade do estado do sistema (*feedback*)
2. Correspondência entre o sistema e o mundo real
3. Controle e liberdade do usuário
4. Consistência e padronização
5. Prevenção de erros
6. Reconhecimento em vez de memorização
7. Flexibilidade e eficiência de uso
8. Projeto estético e minimalista
9. Auxilia os usuários a reconhecerem, diagnosticarem e se recuperarem de erros
10. Ajuda e documentação

Para a avaliação, foram escolhidas as seguintes funcionalidades:

1. Criar novo dicionário de símbolos
2. Construir um sinal
3. Salvar um sinal
4. Selecionar um dicionário de símbolos existente
5. Consultar um sinal
6. Usar um sinal

A seguir, um breve relato, para o qual é importante salientar que o foco foi verificar problemas de interface que devem ser evitados na construção do *SignSMS*.

No que se refere à criação de um novo dicionário de símbolos destacamos:

- Heurística(s) violada(s): Visibilidade do estado do sistema, Reconhecimento em vez de memorização, Correspondência entre o sistema e o mundo real.
- Problema: botão Editar Dicionário serve para criar novos dicionários.
- Local: barra de menus ao lado direito do botão imprimir (Figura 40).

- Severidade: 2 (problema pequeno). Ao clicar no botão o usuário descobrirá que este botão é o botão para criar um novo dicionário de símbolos e não para editar um dicionário existente.
- Recomendação: alterar o nome do botão para Novo Dicionário.

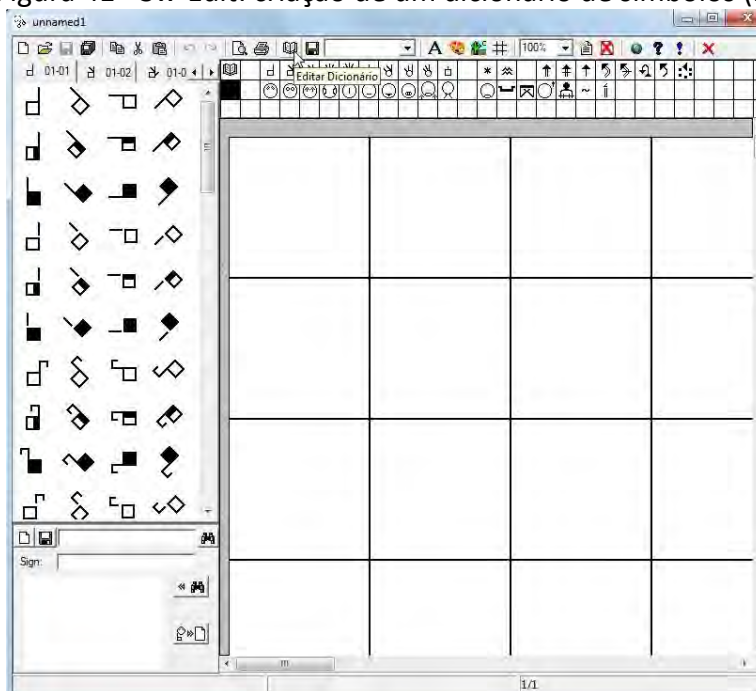
Comentário: para criar um novo dicionário de símbolos é preciso clicar no botão Editar Dicionário, clicar no botão OK (os campos *SignWriting*, País, Autor e Descrição na tela Novo Dicionário são de preenchimento facultativo), inserir o nome do novo dicionário de símbolos e salvar o dicionário clicando no botão Salvar (Figura 41, Figura 42, Figura 43).

Figura 40 – Sw-Edit: Botão Editar Dicionário



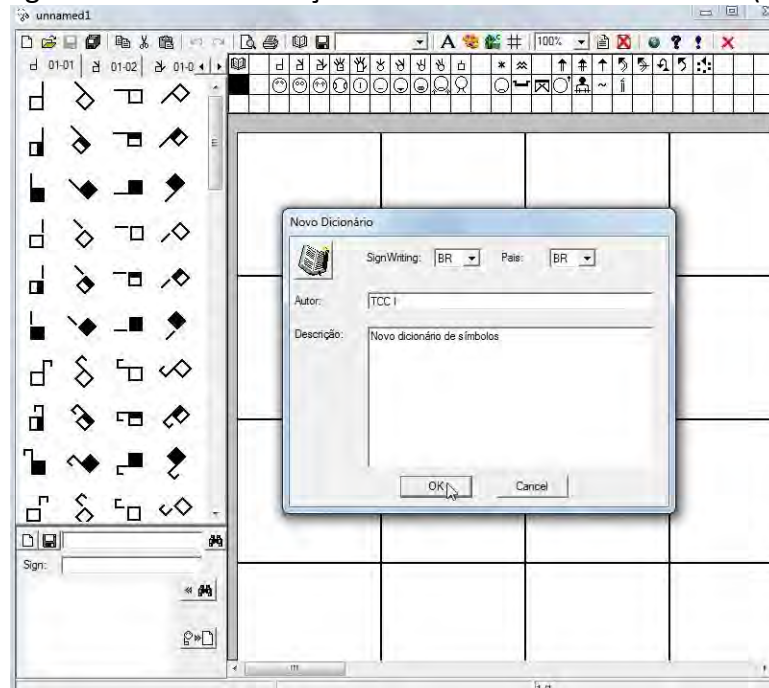
Fonte: SW-Edit (2012)

Figura 41 - Sw-Edit: criação de um dicionário de símbolos (1)



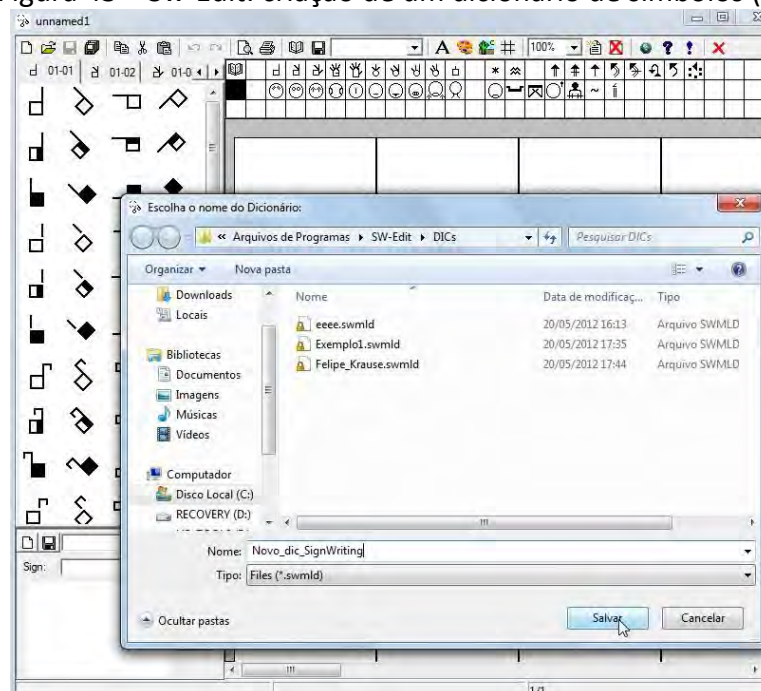
Fonte: SW-Edit (2012)

Figura 42 - Sw-Edit: criação de um dicionário de símbolos (2)



Fonte: SW-Edit (2012)

Figura 43 – Sw-Edit: criação de um dicionário de símbolos (3)



Fonte: SW-Edit (2012)

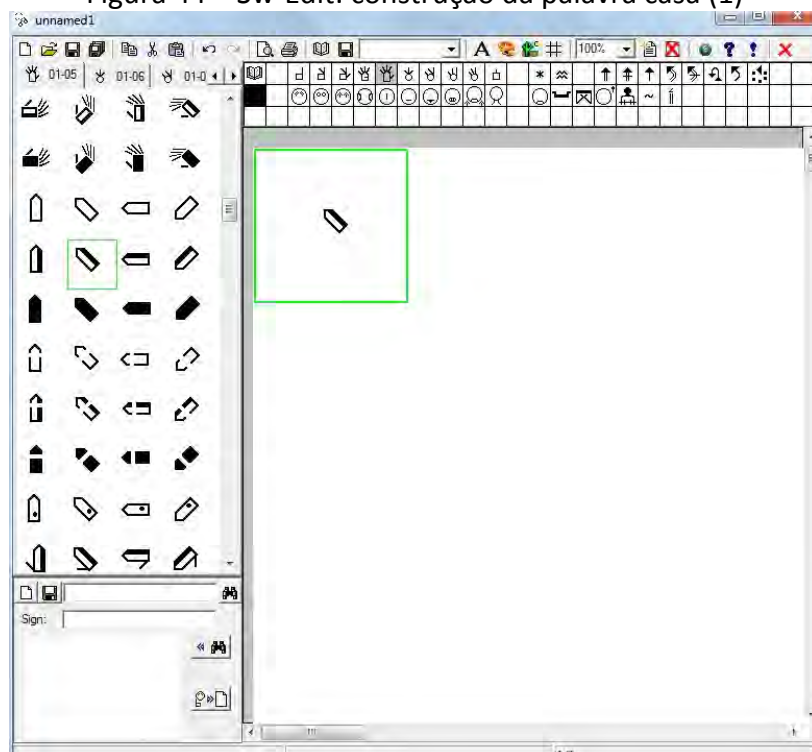
No que se refere à construção de um sinal, destacamos:

- Heurística(s) violada(s): Reconhecimento em vez de memorização.

- Problema: as opções de configuração de um sinal são acessíveis apenas utilizando o botão direito do *mouse* (Figura 46).
- Local: dentro da área de texto, selecionando uma figura e clicando com o botão direito do *mouse*.
- Severidade: 2 (problema pequeno). Geralmente as ferramentas apresentam opções de configuração através do botão direito do *mouse*, não cabe ao usuário saber desta relação, mas após uma análise das opções de configuração das figuras na interface da aplicação, restará a opção das funcionalidades oferecidas pelo *mouse*.
- Recomendação: a opção Ajuda deveria conter uma descrição das funcionalidades da aplicação.

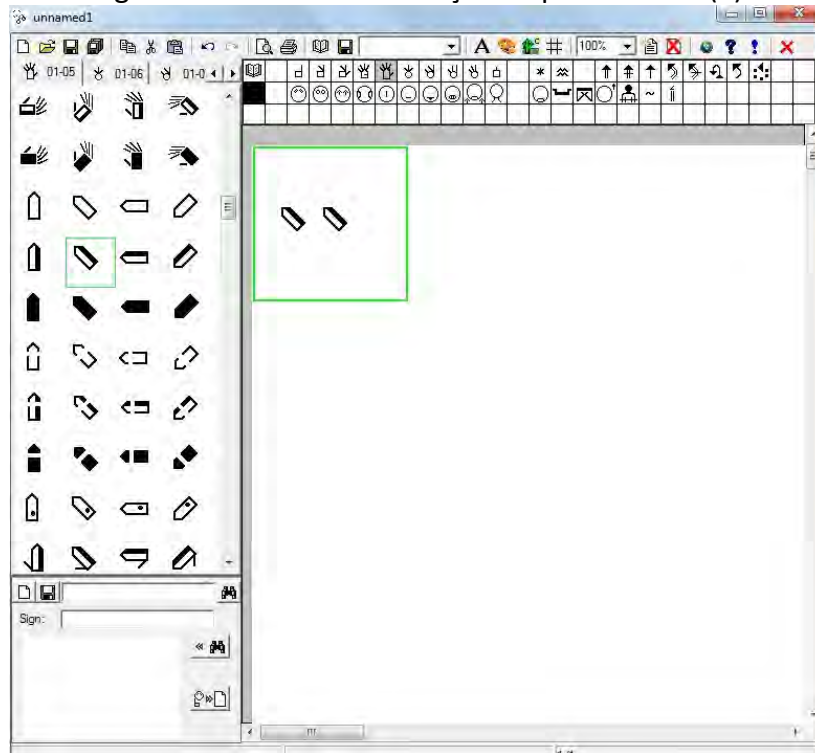
Comentário: para construir um sinal é necessário adicionar um ou mais símbolos das abas de símbolos para a área de texto ou à área de adição de novos sinais e organizá-los e configurá-los, de modo que estes formem o sinal desejado. A Figura 44, Figura 45, Figura 46, Figura 47 e Figura 48 demonstram as etapas para a construção da palavra casa.

Figura 44 – Sw-Edit: construção da palavra casa (1)



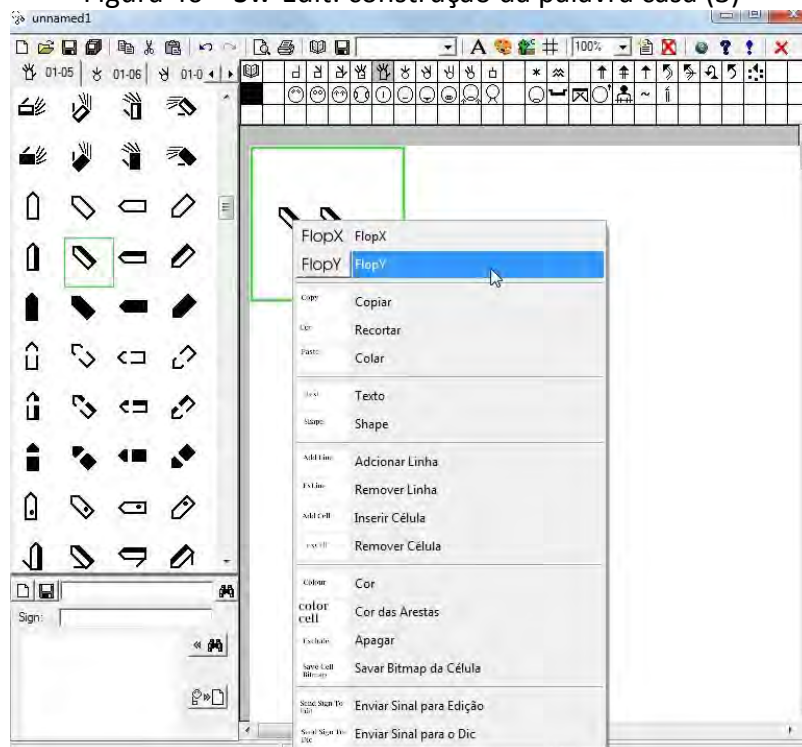
Fonte: SW-Edit (2012)

Figura 45 - Sw-Edit: construção da palavra casa (2)



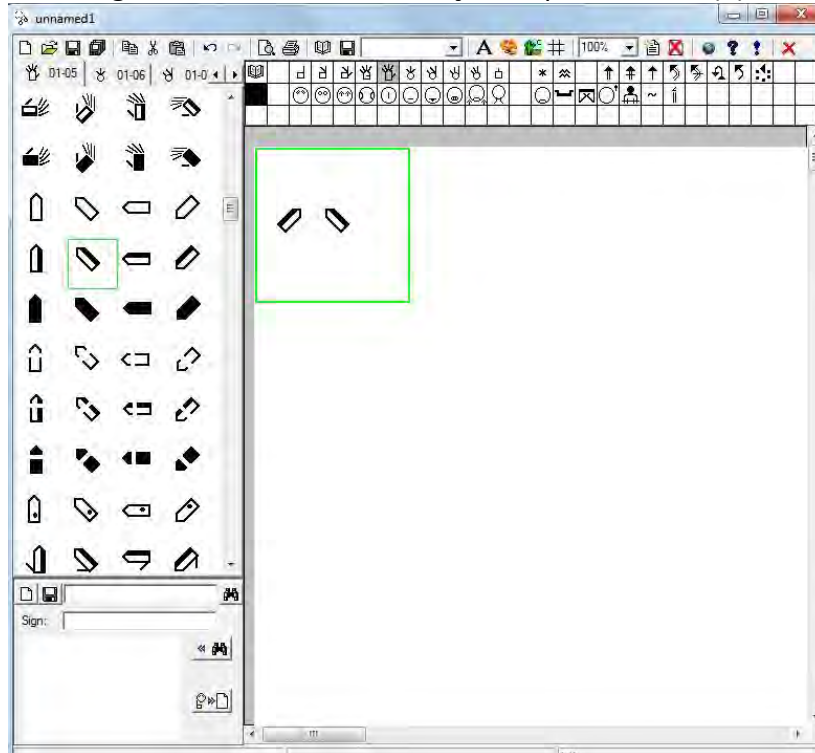
Fonte: SW-Edit (2012)

Figura 46 – Sw-Edit: construção da palavra casa (3)



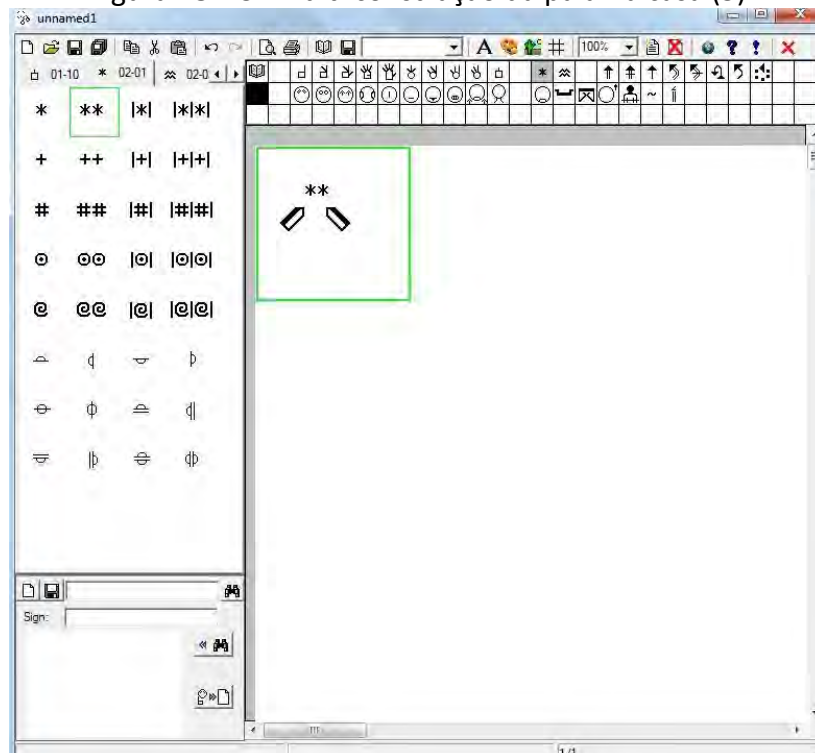
Fonte: SW-Edit (2012)

Figura 47 – Sw-Edit: construção da palavra casa (4)



Fonte: SW-Edit (2012)

Figura 48 – Sw-Edit: construção da palavra casa (5)



Fonte: SW-Edit (2012)

No que se refere ao processo de salvar um sinal, destacamos:

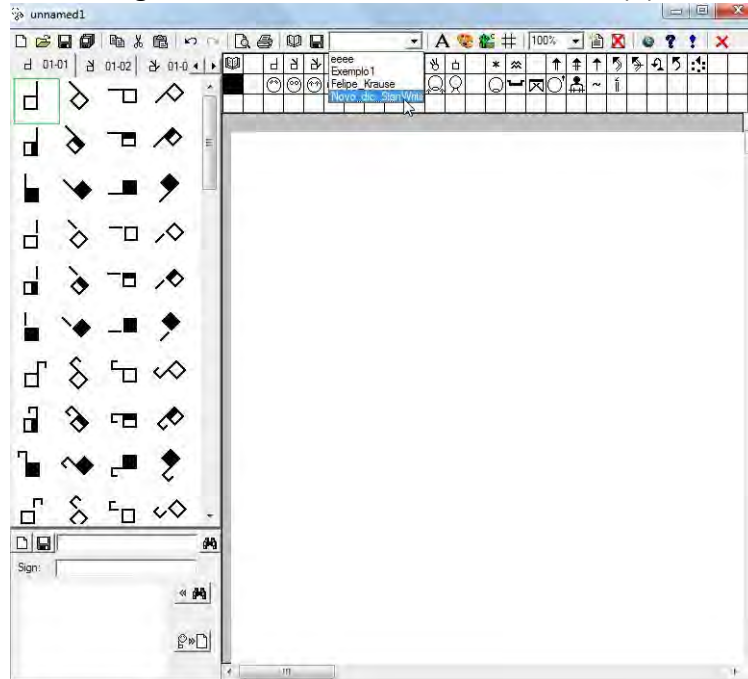
- Heurística(s) violada(s): Visibilidade do estado do sistema, Reconhecimento em vez de memorização.
- Problema: a interface não é intuitiva, pois não há referência para que o usuário saiba onde deve criar o novo símbolo. Isso se deve ao fato da área em que o símbolo é ajustado para ser adicionado no dicionário de símbolos não ter destaque na interface. Caso o usuário construa um símbolo na grade de texto e clicar no botão Salvar novo símbolo nada acontece.
- Local: área para editar/adicionar um símbolo, localizada no canto inferior esquerdo da aplicação.
- Severidade: 3 (problema grande). Tratando-se de um problema de interface, as ações corretivas demandariam esforço de implementação e como o produto já está em uso no mercado uma opção seria desenvolver um patch para aplicar as mudanças necessárias.
- Recomendação: remover a área de adição de um novo sinal, pois esta é pequena e sem destaque na interface e permitir que um sinal montado na área de texto possa ser adicionado no dicionário corrente. Ao selecionar a opção de adicionar o sinal na grade selecionada, uma janela *pop-up* é aberta para que o usuário possa nomear o sinal que será adicionado no dicionário corrente.

Comentário: quanto à ação de salvar um novo sinal, esta não apresenta dificuldades na execução, mas requer que o usuário siga alguns passos simples, demonstradas pela Figura 50, Figura 51, Figura 52, Figura 53 e Figura 54:

1. Selecionar um dicionário para incluir o novo símbolo;
2. Arrastar para a área de edição os símbolos necessários para formar o sinal;
3. Posicionar os símbolos na área de edição, de forma que o sinal fique legível;
4. Inserir o nome do novo símbolo;
5. Clicar no botão Salvar novo símbolo.

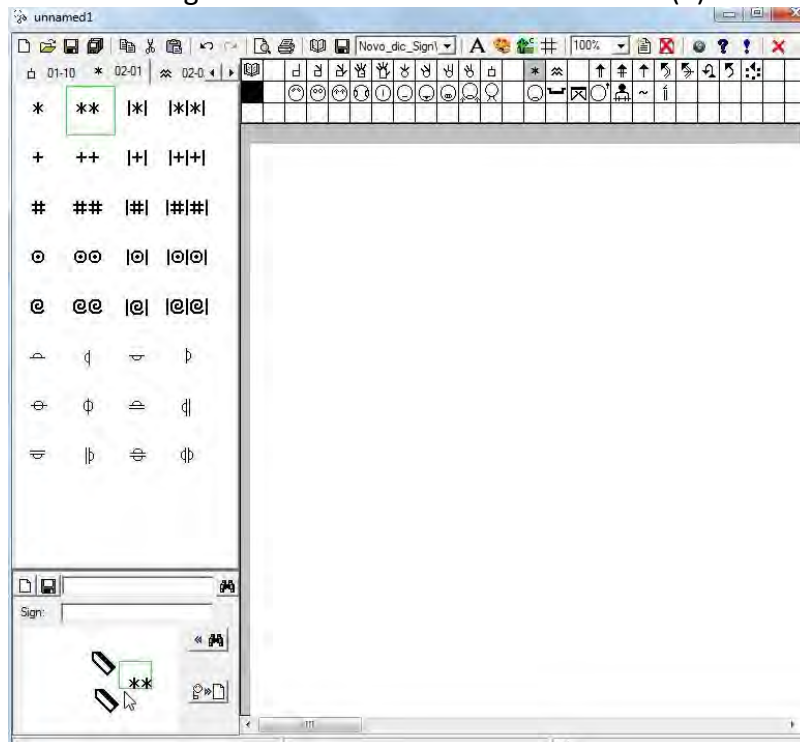


Figura 49 – SW-Edit: salvando um sinal (1)



Fonte: SW-Edit (2012)

Figura 50 – SW-Edit: salvando um sinal (2)



Fonte: SW-Edit (2012)



Figura 51 – SW-Edit: salvando um sinal (3)

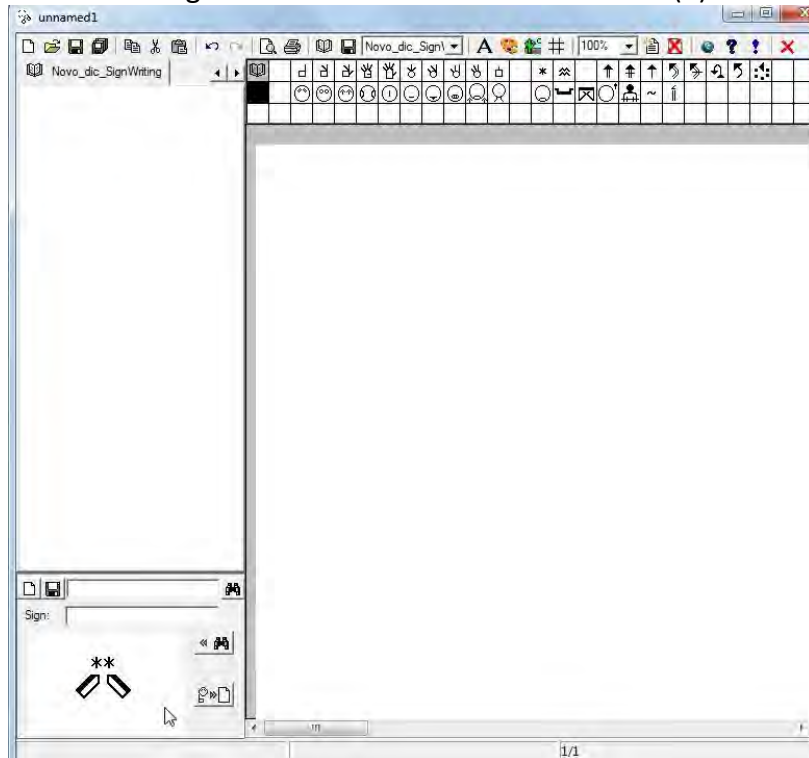
Fonte: *SW-Edit* (2012)

Figura 52 – SW-Edit: salvando um sinal (4)

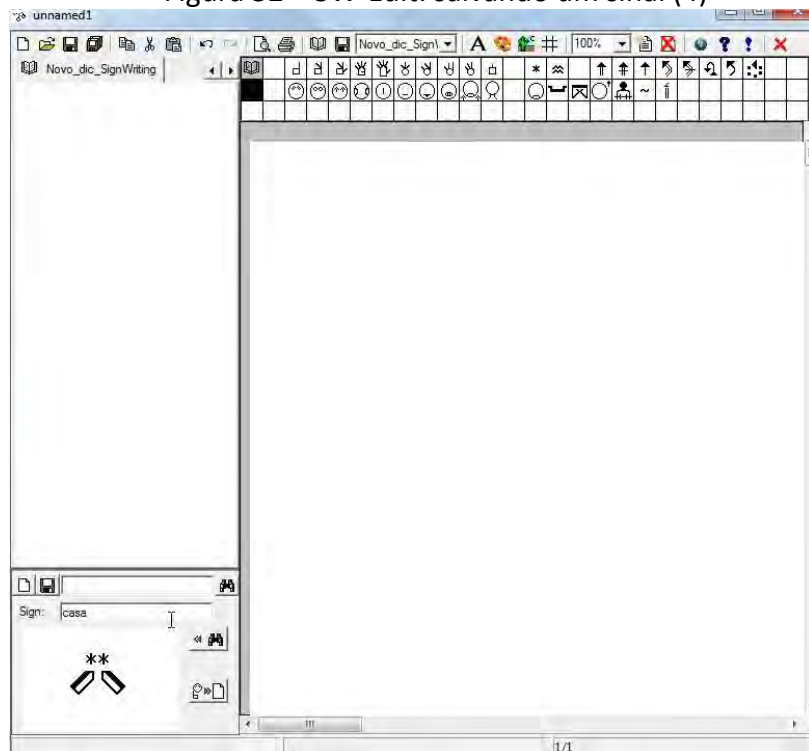
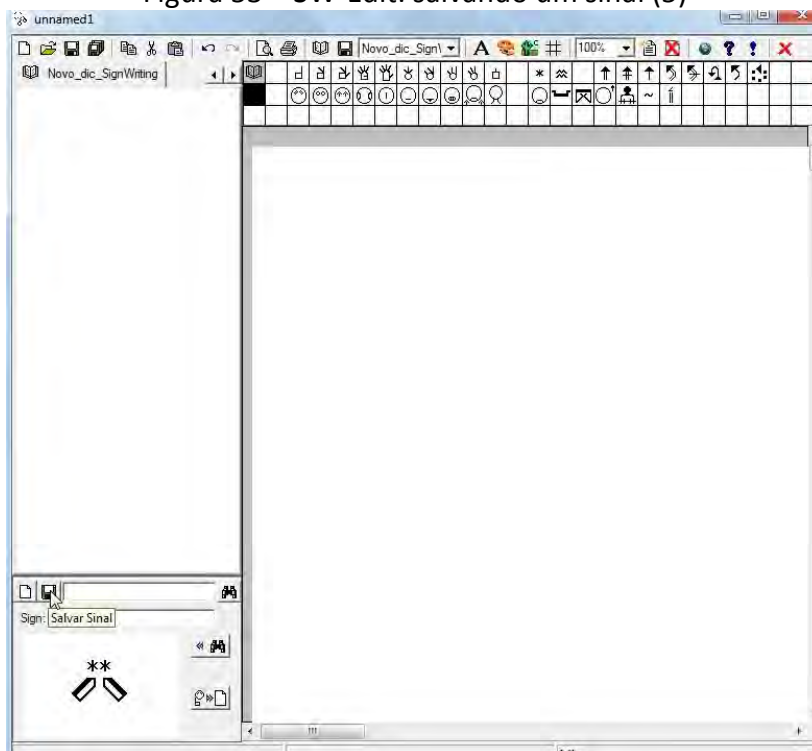
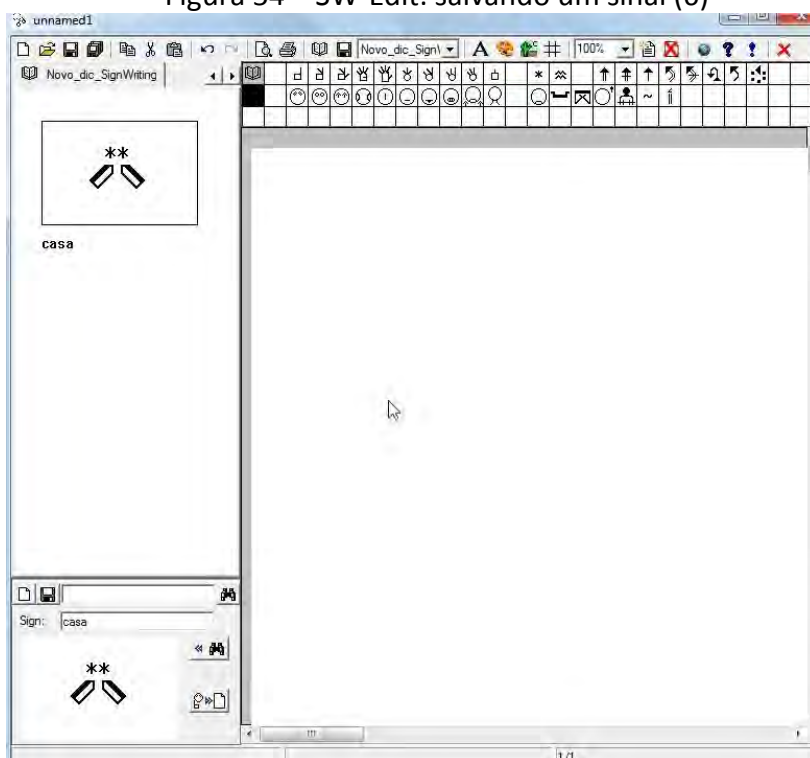
Fonte: *SW-Edit* (2012)

Figura 53 – SW-Edit: salvando um sinal (5)



Fonte: *SW-Edit* (2012)

Figura 54 – SW-Edit: salvando um sinal (6)



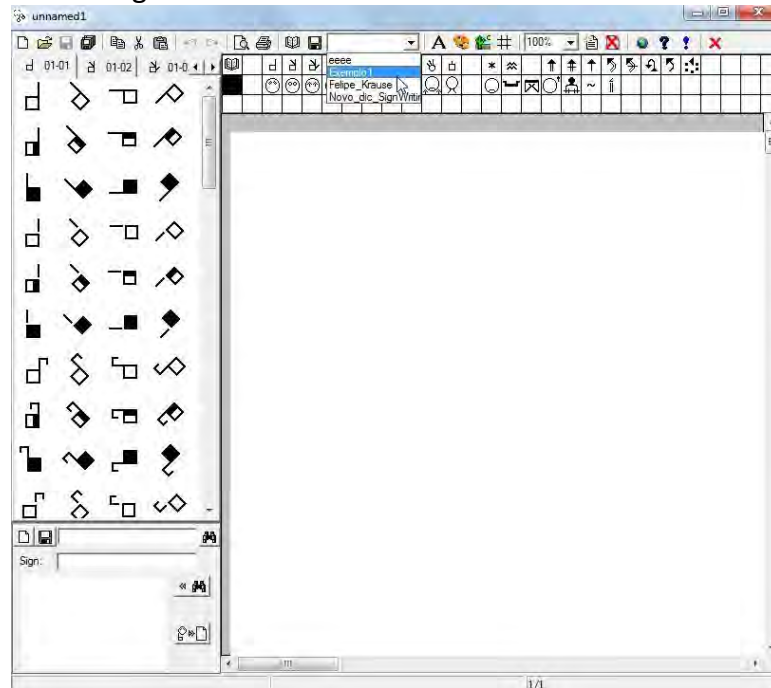
Fonte: *SW-Edit* (2012)

No que se refere ao processo de selecionar um dicionário de símbolos existente, destacamos:

- Heurística(s) violada(s): Visibilidade do estado do sistema.
- Problema: a seleção de um dicionário de símbolos não mostra os símbolos salvos quando este dicionário é selecionado. Para visualizar os símbolos o usuário deve selecionar um dicionário na caixa de seleção e depois clicar no botão dicionário na grade de grupo de símbolos.
- Local: caixa de seleção localizada entre o botão Salvar dicionário corrente e o botão Inserir caixa de texto.
- Severidade: 3 (problema grande). Ao selecionar um dicionário e verificar que nenhum símbolo apareceu na tela de símbolos, que fica do lado esquerdo da tela, o usuário pode achar que o dicionário está corrompido, tentar selecioná-lo mais de uma vez, e até ficar perdido na interface sem saber o que fazer.
- Recomendação: exibir os símbolos do dicionário assim que este for selecionado pelo usuário.

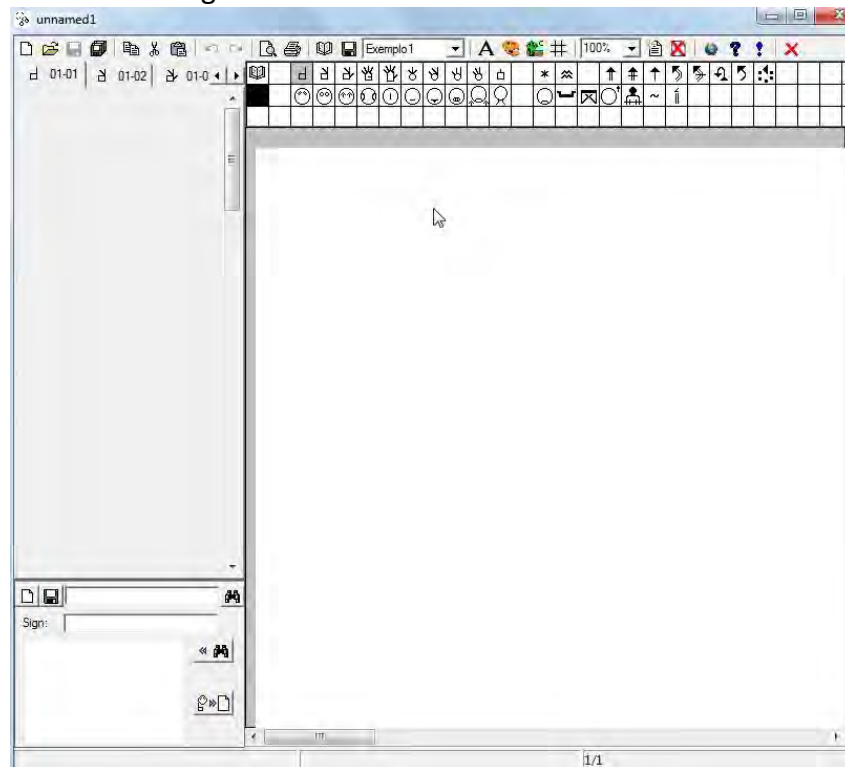
Comentário: para selecionar um dicionário de símbolos o usuário deve clicar na caixa de seleção de dicionários e selecionar um dicionário existente (Figura 55), depois de selecionado, o dicionário não mostra os símbolos automaticamente (Figura 56), o usuário deve clicar na figura referente ao dicionário na grade de símbolos para que os símbolos do dicionário apareçam (Figura 57). Caso não haja dicionários disponíveis, a caixa de seleção retornará vazia (Figura 58).

Figura 55 - SW-Edit: selecionando um dicionário



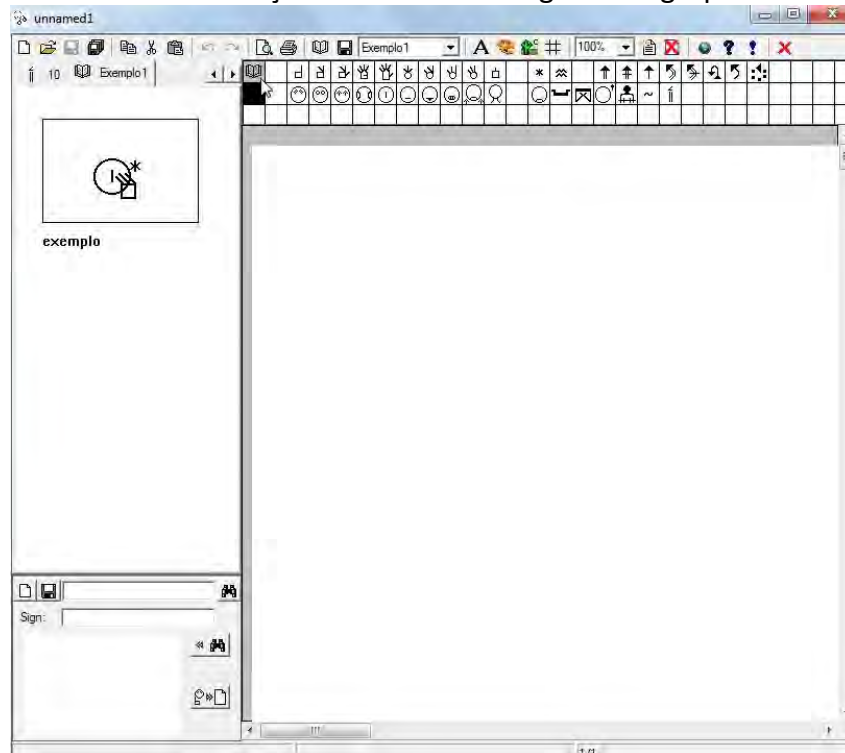
Fonte: SW-Edit (2012)

Figura 56 - SW-Edit: dicionário selecionado



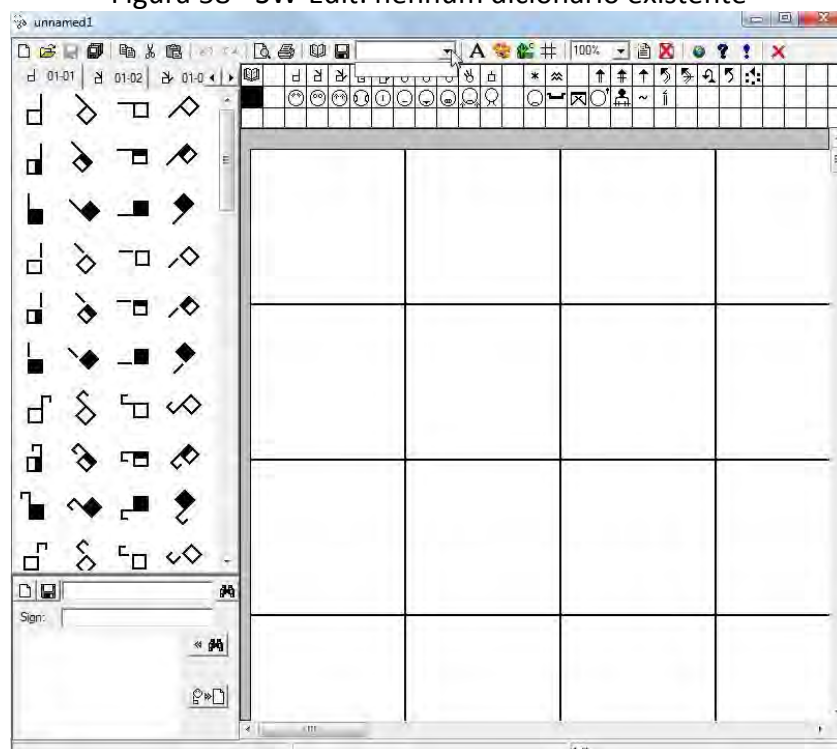
Fonte: SW-Edit (2012)

Figura 57 - SW-Edit: seleção do dicionário na grade de grupos de símbolos



Fonte: *SW-Edit* (2012)

Figura 58 - SW-Edit: nenhum dicionário existente



Fonte: *SW-Edit* (2012)

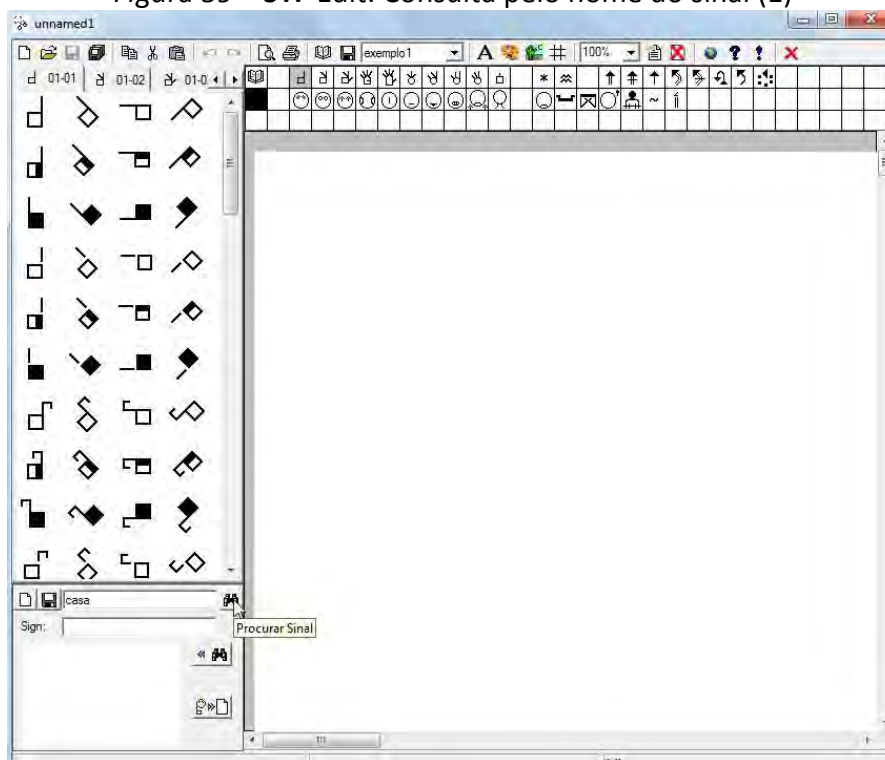
No que se refere ao processo de consultar um sinal, destacamos:

- Heurística(s) violada(s): Visibilidade do estado do sistema.

- Problema: a consulta pelo nome do sinal deve ser feita somente no primeiro campo de texto na região de consulta de sinais da interface.
- Local: caixa de texto abaixo do lado direito do botão Salvar Sinal.
- Severidade: 2 (problema pequeno). Como existem dois campos de texto, o usuário terá de fazer a operação duas vezes, caso preencha o campo de texto não correspondente na primeira tentativa.
- Recomendação: inserir um texto explicativo na caixa de texto referente à pesquisa de sinais pelo nome como, por exemplo, consultar sinal pelo nome.

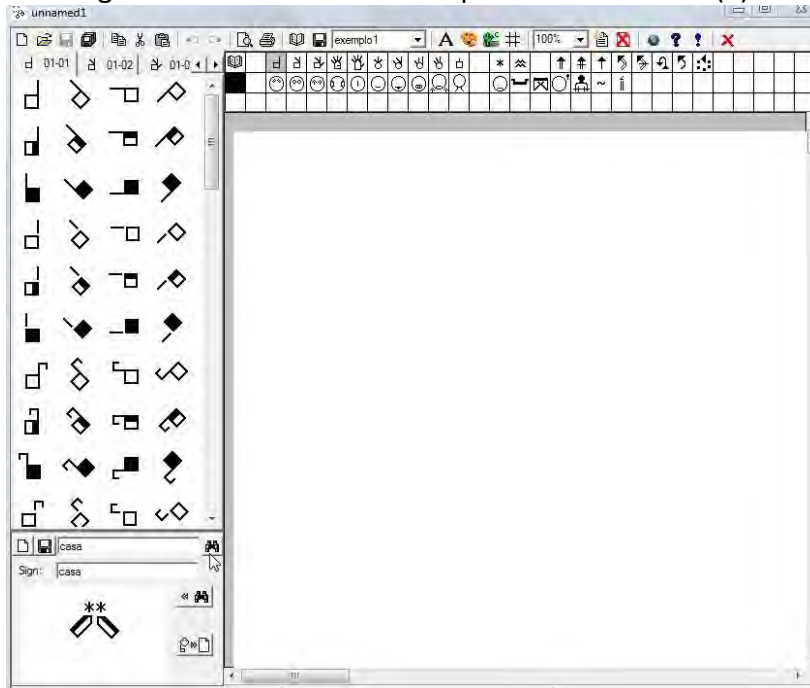
Comentário: a ação de consulta de um sinal pode ser feita de duas maneiras: após selecionar um dicionário de símbolos (Figura 58) pode-se tanto pesquisar o sinal pelo nome dado ao sinal em alfabeto romano (Figura 59, Figura 60), quanto por uma figura que faz parte do sinal em *SignWriting* (Figura 61, Figura 62).

Figura 59 – SW-Edit: Consulta pelo nome do sinal (1)



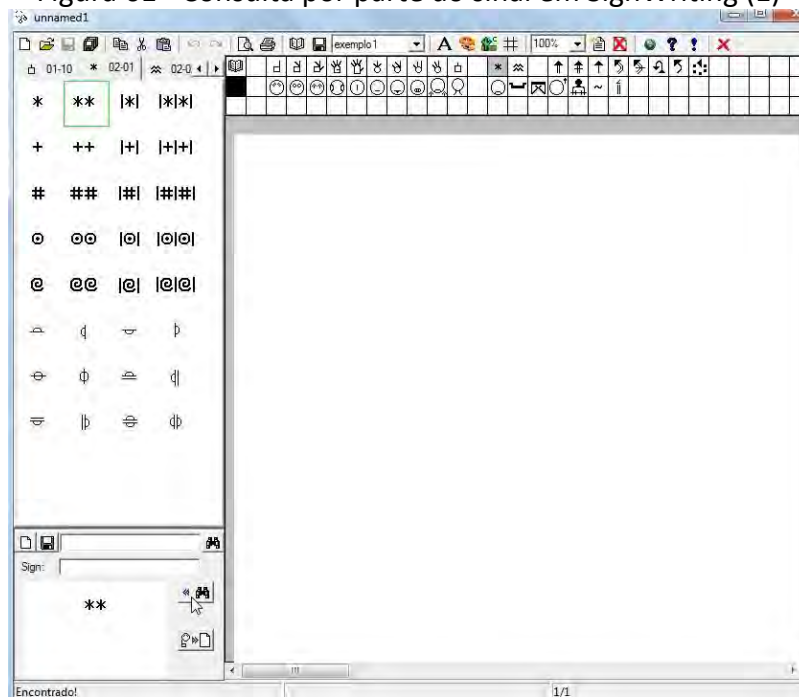
Fonte: SW-Edit (2012)

Figura 60 - SW-Edit: Consulta pelo nome do sinal (2)



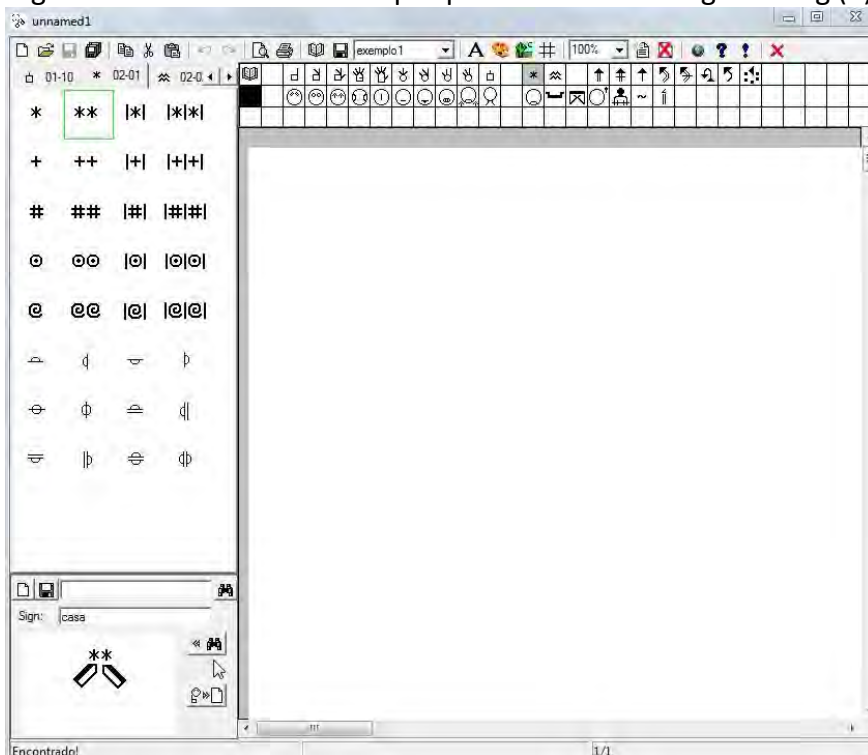
Fonte: SW-Edit (2012)

Figura 61 - Consulta por parte do sinal em SignWriting (1)



Fonte: SW-Edit (2012)

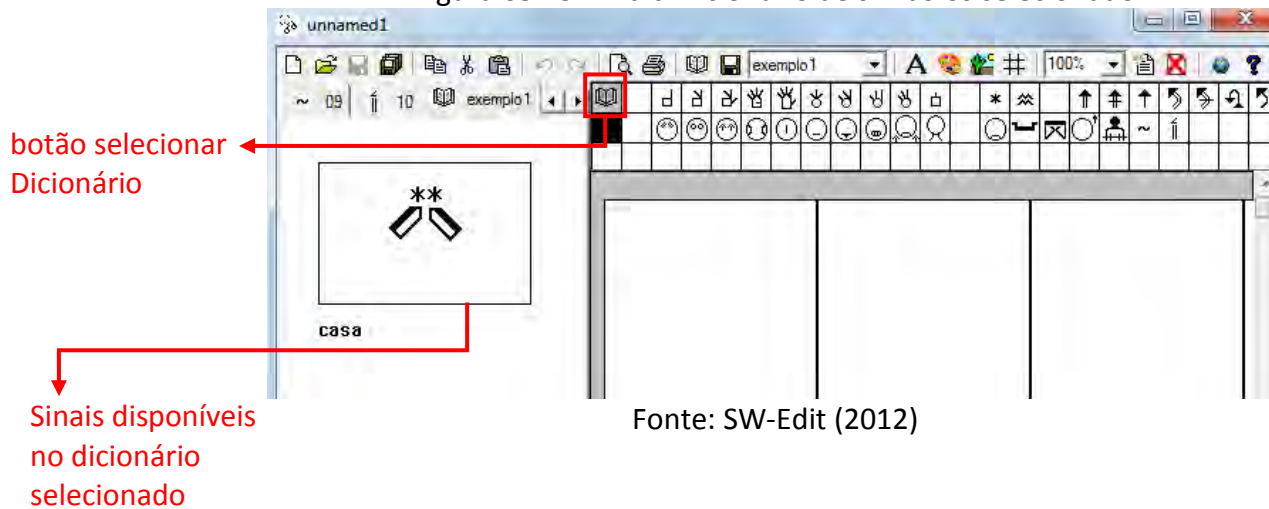
Figura 62 - SW-Edit: Consulta por parte do sinal em SignWriting (2)



Fonte: SW-Edit (2012)

Não foram encontradas violações nas heurísticas de Nielsen referentes à ação de usar um sinal dada à facilidade da mesma. Após selecionar um dicionário de símbolos, basta clicar no botão dicionário na grade de símbolos para visualizar os sinais disponíveis no dicionário (Figura 63). Depois de apresentados os sinais disponíveis, para utilizá-los no texto pode-se tanto clicar e arrastar o sinal para uma área do texto qualquer ou selecionar uma grade no texto e dar um duplo clique no sinal desejado para inseri-lo no texto.

Figura 63 - SW-Edit: Dicionário de símbolos selecionado

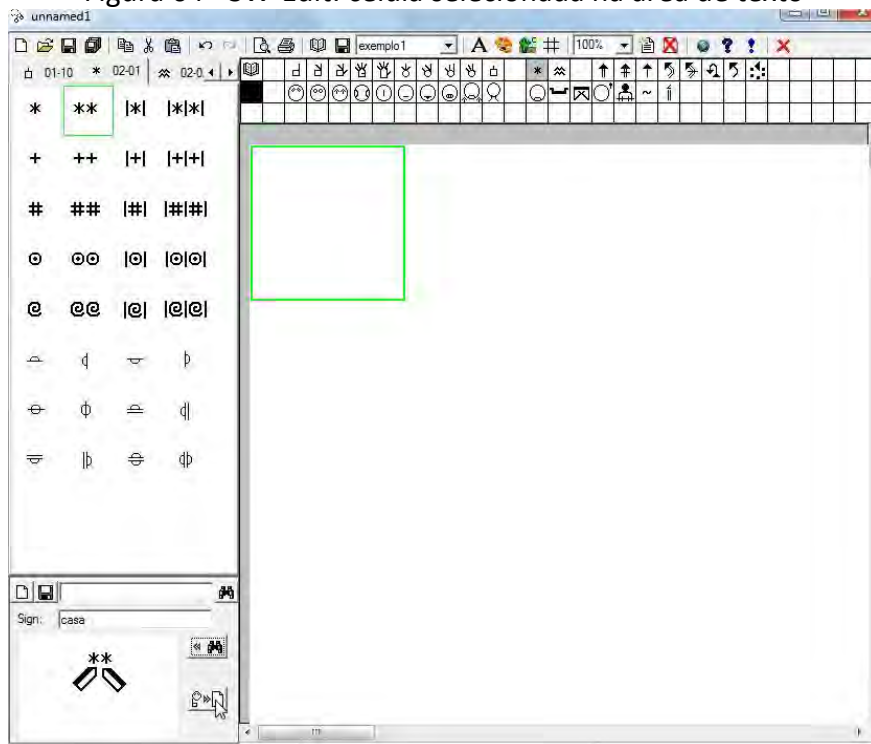


Fonte: SW-Edit (2012)



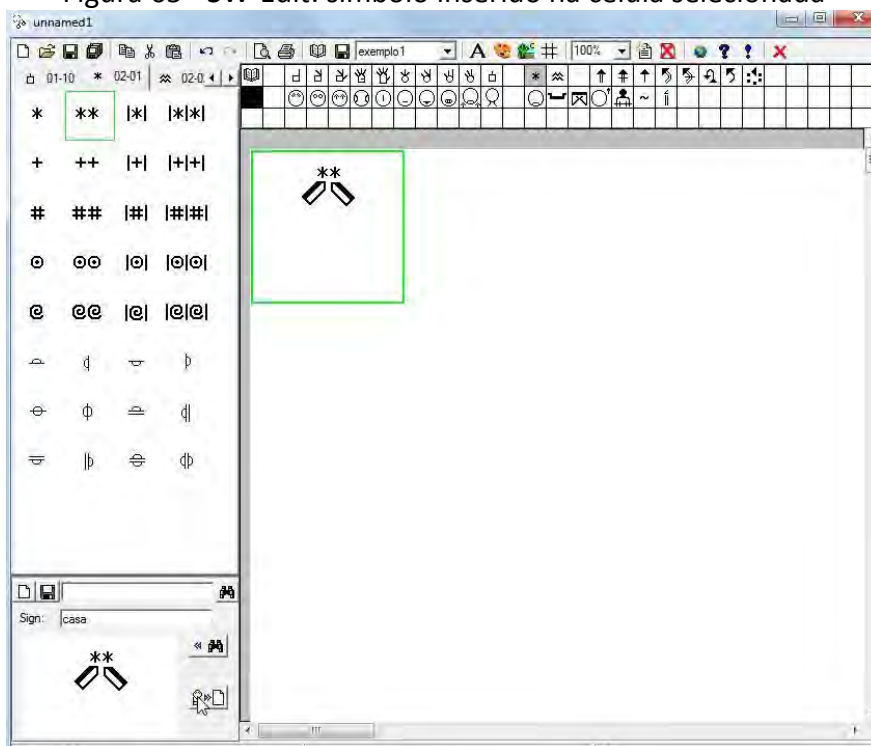
Caso o sinal desejado tenha sido resultado de uma pesquisa o usuário pode selecionar uma área na região do texto (Figura 64) e clicar no botão adicionar símbolo para inserir o sinal na região selecionada (Figura 65).

Figura 64 - SW-Edit: célula selecionada na área de texto



Fonte: SW-Edit (2012)

Figura 65 - SW-Edit: símbolo inserido na célula selecionada



Fonte: SW-Edit(2012)

### 3.1.3. *Sign WebMessage*

No que se refere a aplicações para Web, destacamos o *Sign WebMessage*, que é uma ferramenta para comunicação assíncrona na qual o usuário pode interagir tanto por meio da Língua Portuguesa quanto por meio do *SignWriting*. De acordo com Souza & Pinto (2003), tem como objetivo principal minimizar as dificuldades de comunicação escrita entre os surdos e entre os surdos e ouvintes.

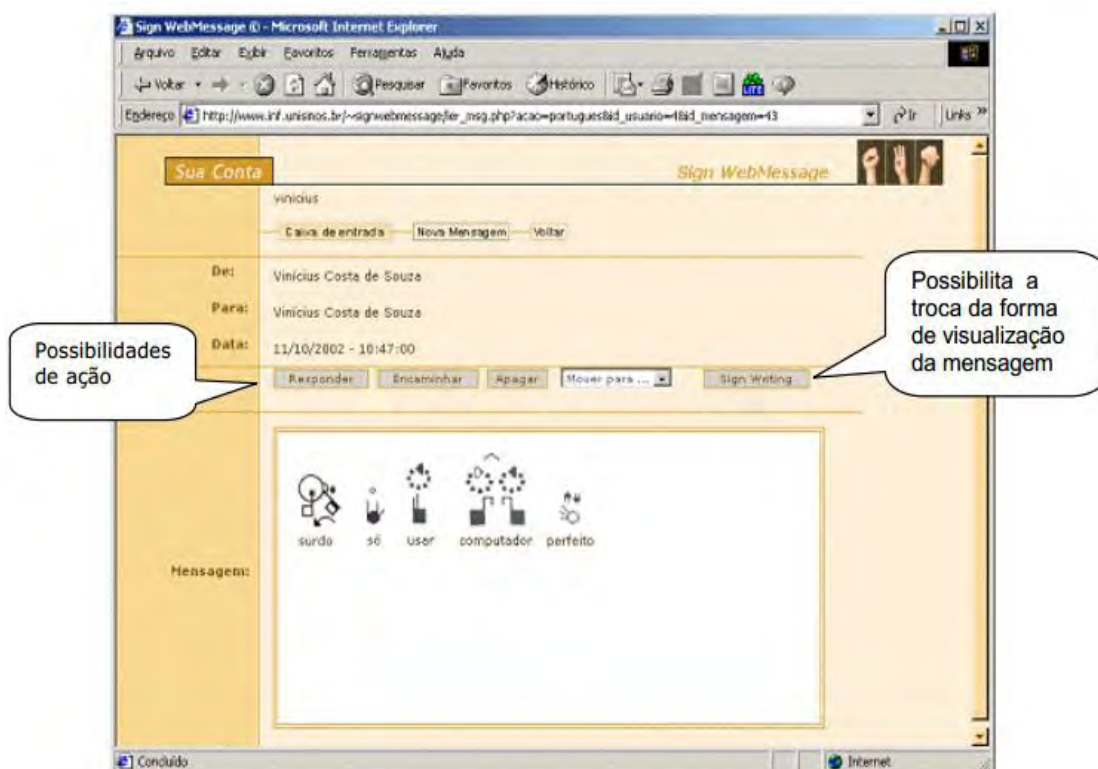
A troca de mensagens entre os usuários é semelhante ao funcionamento de um correio eletrônico. O usuário deve estar cadastrado no ambiente para poder usufruir das funcionalidades. O *Sign WebMessage* permite, então, a troca de mensagens assíncrona entre os usuários cadastrados no sistema, permite que as mensagens sejam gerenciadas pelo usuário de modo que o usuário tenha controle sobre as ações básicas de exclusão, encaminhamento, resposta e leitura das mensagens recebidas e enviadas.

Dentre as funcionalidades semelhantes às de um correio eletrônico, podemos citar:

- Envio de mensagens pela web
- Gerenciamento das mensagens recebidas
- Leitura de mensagens
- Lista de contatos

Somadas a estas funcionalidades temos customizações referentes ao uso do *SignWriting* como forma de escrita de mensagens e opção de criação de novos sinais no ambiente. A Figura 66 ilustra uma mensagem selecionada para leitura na caixa de entrada das mensagens.

Figura 66 - Sign WebMessage: caixa de entrada das mensagens



Fonte: Souza (2002)

Na criação de uma mensagem, o usuário pode fazer consultas ao dicionário de sinais da aplicação para a escrita em *SignWriting*. Para isto, o usuário deve digitar uma palavra na opção de busca ou procurar na lista de palavras do dicionário, conforme Figura 67.

Figura 67 – Sign WebMessage: modo de consulta



Fonte: Souza (2002)

O usuário ainda pode consultar um sinal utilizando a busca a partir da língua de sinais de acordo com as características gestuais do sinal. A Figura 68 ilustra a busca por língua de sinais. Esse processo se assemelha ao recurso de predição de sinais do Pacote *Sign*.

Figura 68 – Sign WebMessage: busca de sinal por língua de sinais



Fonte: Souza (2002)

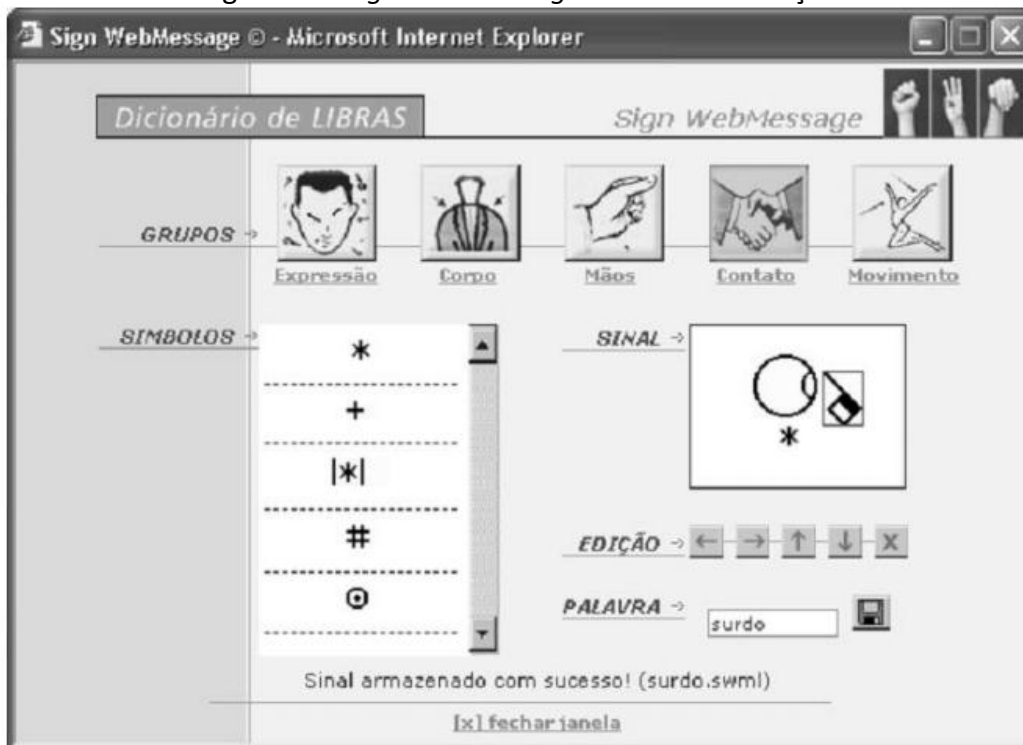
Depois de selecionar o sinal, basta inseri-lo na mensagem e repetir o processo para outros sinais de forma a escrever uma sentença.

Outra funcionalidade do *Sign WebMessage* é a opção de criar novos sinais para o dicionário do ambiente. Para isto o usuário deve acessar o módulo para construção de sinais, selecionar os símbolos que farão parte do novo sinal, posicioná-los, forma similar ao editor do Pacote *Sign*, e finalmente criar e adicionar o novo sinal na base de sinais do ambiente (Souza, 2002). O módulo de criação de novos sinais também permite que o usuário apague os sinais em caso de erro. A Figura 69 ilustra módulo de criação de sinais e a Figura 70 representa o arquivo SWML<sup>13</sup> correspondente. O SWML é uma iniciativa promovida pela comunidade do SignWriting.org e consiste de um arquivo XML que

<sup>13</sup> SWML significa SignWriting Markup Language e é um formato baseado em XML para o armazenamento e processamento de documentos escritos em língua de sinais do sistema SignWriting, e para a inserção de textos de linguagem gestual em documentos HTML. Mais informações em <<http://www.signwriting.org/archive/docs1/sw0031-SWML-Programming.pdf>>. Acesso em 10 jun. 2012

representa um conjunto de configurações de sinais de um documento de linguagem gestual ou um banco de dados escritos em SignWriting.

Figura 69 - *Sign WebMessage*: módulo de criação



Fonte: Souza & Pinto (2004)

Figura 70 - *Sign WebMessage*: arquivo SWML

```
<swml version="1.0-d2" symbolset="SSS-1999">
<generator>
  <name>Sign WebMessage</name>
  <version>1.0</version>
</generator>
<sw_dic>
  <sign_box>
    <symbol x="35" y="15">
      <shape number="231" fill="1" variation="1">
        <transform flop="0" rotation="3">

```

Fonte: Souza & Pinto (2004)

Este capítulo já descreveu exemplos de aplicações para *desktop* e para *Web*. Os aplicativos a seguir são direcionados a dispositivos móveis. Ressalta-se que esse estudo teve

como objetivo verificar a forma de representação das LS, bem como as formas de interação com o usuário no que se refere, também, a aplicação de padrões de *design*.

### 3.1.4. *Sign 4 Me*

É um aplicativo para *iPhone* desenvolvido pela empresa *Vcom3D*<sup>14</sup>, que não requer acesso à Internet para seu funcionamento. Seu objetivo é fazer a tradução entre a escrita de uma língua oral (ASL) para a de sinais. Para tal, utiliza um avatar.

Segundo o desenvolvedor<sup>15</sup> o *Sign 4 Me* conta com um banco de frases que pode ser incrementado com frases introduzidas pelo usuário, além de contar com um banco léxico para que as palavras possam ser mapeadas e representadas pelo avatar. Sua biblioteca inclui mais de 11.500 palavras e a entrada de dados permite que o usuário digite sentenças, frases, palavras ou mesmo o alfabeto. Todas as entradas são salvas no histórico e podem ser localizadas por data ou por ordem alfabética, sendo que o usuário pode apagar o que não mais desejar manter no histórico.

Quanto aos controles do avatar, é possível acelerar ou diminuir a velocidade da tradução dos sinais, ligar ou desligar o recurso de zoom, e controlar a posição do avatar na tela). Quanto ao modo de navegação, os menus não apresentam características diferenciadas para a implementação do *SignSMS*, o que este aplicativo nos remete é a questão de que um design elegante também é um diferencial em uma aplicação. A Figura 71 ilustra o ícone da aplicação enquanto a Figura 72, Figura 73, Figura 74 demonstram telas da funcionalidade da aplicação.

Figura 71 - *Sign 4 Me*: ícone da aplicação



Fonte: *Sign 4 Me* (2012)

---

<sup>14</sup> VCOM3D – Mais informações em <<http://www.vcom3d.com/index.php?id=sign4me>>.

<sup>15</sup> A descrição do aplicativo *Sign 4 Me* foi feita baseada na descrição das funcionalidades no site do desenvolvedor, de forma que não foi possível testar diretamente a aplicação.



Figura 72 - *Sign 4 Me*: interface para inserção de texto



Fonte: *Sign 4 Me* (2012)

Figura 73 - *Sign 4 Me*: animação do texto em língua de sinais



Fonte: *Sign 4 Me* (2012)

Figura 74 - *Sign 4 Me*: controle do avatar



Fonte: *Sign 4 Me* (2012)

### 3.1.5. *SkySigns ASL Dictionary*

Este aplicativo está disponível no site AppBrain<sup>16</sup>, e foi desenvolvido por Schuyler Duveen. Esta aplicação é direcionada para pessoas que conheçam o *SignWriting*, ou que gostariam de ter contato com a escrita de sinais. Segundo o desenvolvedor<sup>17</sup>, esta ferramenta consiste em um dicionário que relaciona uma palavra no idioma inglês com sua escrita em *SignWriting* (ASL).

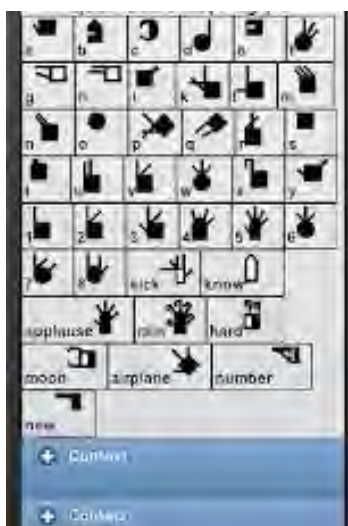
Uma característica interessante é que não há necessidade de conexão com a Internet, pois este aplicativo armazena todos os símbolos utilizados no dispositivo. A Figura 75 mostra o ícone da aplicação enquanto a Figura 76 e a Figura 77 ilustram a entrada de dados por meio do alfabeto manual em *SignWriting* e por meio da escrita da língua oral, fazendo uso do mecanismo de predição de palavra. Já a Figura 78 ilustra o sinal, que foi pesquisado.

Figura 75 - *SkySigns ASL Dictionary*: ícone do aplicativo



Fonte: SkySigns ASL Dictionary (2012)

Figura 76 - *SkySigns ASL Dictionary*: escrita por meio do alfabeto manual em *SignWriting*



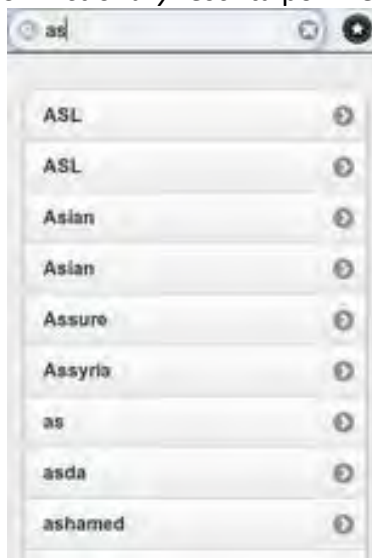
Fonte: SkySigns ASL Dictionary (2012)

<sup>16</sup> AppBrain -- Mais informações em < <http://pt.appbrain.com/app/skysigns-asl-dictionary/us.skyb.skysigns> >.

<sup>17</sup> A descrição do aplicativo *SkySigns ASL Dictionary* foi feita baseada na descrição das funcionalidades no site de distribuição do aplicativo, de forma que não foi possível testar diretamente a aplicação.



Figura 77 - *SkySigns ASL Dictionary*: escrita por meio do alfabeto romano



Fonte: SkySigns ASL Dictionary (2012)

Figura 78 - *SkySigns ASL Dictionary*: resultado da busca do sinal



Fonte: SkySigns ASL Dictionary (2012)

### 3.1.6. *SMS to Sign Language for Mobile Phones*

Segundo Jemni (2010), este aplicativo foi desenvolvido pela iniciativa G3ict<sup>18</sup> e tem como objetivo traduzir automaticamente uma mensagem de texto (SMS) para uma mensagem animada (MMS) em ASL. Pela descrição no site, não foi possível verificar se o serviço requer uso da Internet ou da operadora de telecomunicação para a troca das mensagens.

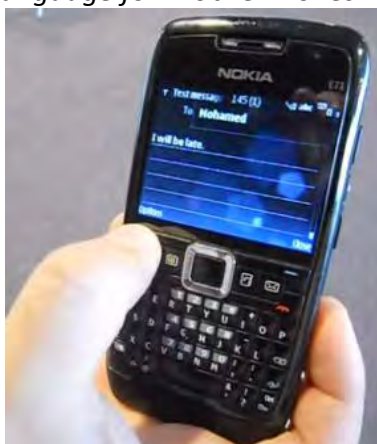
De acordo com Jemni (2010), nesta mensagem, um avatar animado é responsável por reproduzir a mensagem traduzida do texto para os sinais, sendo que o processo de

<sup>18</sup> *SMS to Sign Language for Mobile Phones* – Para mais informações, acesse <<http://g3ict.com/about>>.

tradução acontece a partir do momento em que a mensagem de texto é enviada pelo remetente. Desta forma, esta mensagem chega a um servidor operador de telecomunicações, que interpreta o texto e repassa a mensagem animada para o destinatário. Assim como o aplicativo *Sign 4 Me*, uma vez que o processo de tradução do texto escrito para animação não é foco do nosso estudo, não será detalhado neste trabalho.

A interface no aplicativo *SMS for Sign Language for Mobile Phones*<sup>19</sup> é igual à interface do serviço de envio de SMS do dispositivo, tendo como funcionalidade diferenciada que as mensagens de texto são encaminhadas a um servidor externo que faz a tradução da mensagem para a animação. A Figura 79, Figura 80, Figura 81 e Figura 82 demonstram o processo de envio e recebimento de uma mensagem: escrita do texto, envio da mensagem, recebimento da mensagem com anexo, visualização da mensagem por meio do avatar.

Figura 79 - *SMS to Sign Language for Mobile Phones*: texto escrito pelo usuário

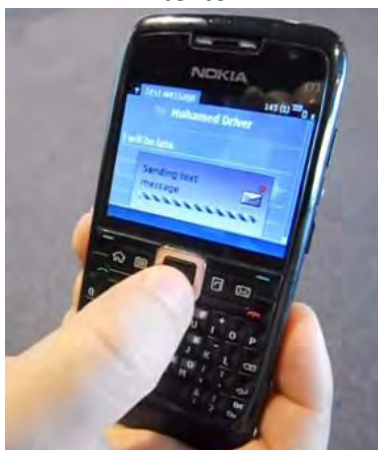


Fonte: Jemni (2010)

---

<sup>19</sup> A descrição do *SMS to Sign Language for Mobile Phones* foi feita baseada na descrição das funcionalidades no site do desenvolvedor, de forma que não foi possível testar diretamente a aplicação.

Figura 80 - *SMS to Sign Language for Mobile Phones*: envio de uma mensagem em formato texto



Fonte: Jemni (2010)

Figura 81 - *SMS to Sign Language for Mobile Phones*: tela contendo a mensagem recebida em texto e em animação



Fonte: Jemni (2010)

Figura 82 - *SMS to Sign Language for Mobile Phones*: tradução em sinais da mensagem de texto



Fonte: Jemni (2010)

### 3.1.7. *Signing Savvy*

De acordo com Signing Savvy (2012), essa aplicação é um dicionário que contém um banco de vídeos em alta resolução de sinais da Língua de Sinais Americana (ASL), que requer acesso à Internet para seu funcionamento. Ainda, segundo Signing Savvy<sup>20</sup>, essa aplicação possui recursos de pesquisa de vídeos de língua de sinais, construção de listas de vocabulários personalizadas que podem ser compartilhadas entre os membros do *Signing Savvy*. No caso da pesquisa, caso não se tenha um vídeo para a sentença pesquisada, o aplicativo retorna mais de um vídeo, conforme as partes da sentença.

Os usuários desse aplicativo podem se cadastrar ou não. No caso de usuários não cadastrados, ele apenas pode verificar a dica do dia relacionado a um sinal. Já os usuários cadastrados, além de visualizar as dicas, também podem realizar pesquisa de texto para buscar vídeos correspondentes.

A Figura 83 ilustra o ícone da aplicação enquanto a Figura 84 e Figura 85 mostram algumas telas do funcionamento da aplicação.

Figura 83 - *Signing Savvy*: ícone da aplicação



Fonte: Signing Savvy (2012)

---

<sup>20</sup> A descrição do *Signing Savvy* foi feita baseada na descrição das funcionalidades no *site* do desenvolvedor, de forma que não foi possível testar diretamente a aplicação.

Figura 84 - *Signing Savvy*: vídeo contendo instruções de um sinal em ASL



Fonte: Signing Savvy (2012)

Figura 85 - *Signing Savvy*: tradução da palavra help de texto para ASL



Fonte: Signing Savvy (2012)

### 3.1.8. *Baby Sign Language Dict. Demo*

Esta é a versão de demonstração (*demo*) do aplicativo *Baby Sign Language Dict*, desenvolvido pela *My Smart Hands*, que não requer acesso à Internet para seu funcionamento (Baby Sign Language Dict, 2012).

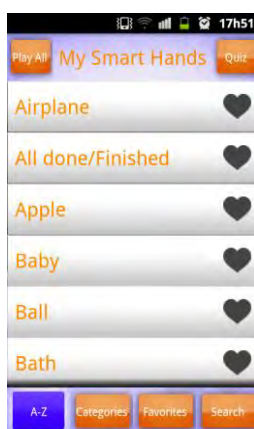
Este aplicativo consiste em um dicionário com vídeos explicativos sobre os sinais da Língua de Sinais Americana (ASL). A versão demo contém aproximadamente 72 sinais enquanto a versão completa possui mais de 300 sinais (BABY SIGN LANGUAGE DICT, 2012). A Figura 86 ilustra o menu inicial da aplicação, que possui as opções de *ASL Dictionary*, que é uma forma de dicionário nas quais as palavras aparecem em ordem alfabética (Figura 87) e, quando é selecionada uma palavra é mostrado um vídeo com a realização do sinal enquanto o sinalizador explica como realizá-lo (Figura 88). A outra opção é a *Baby Signing Lessons* (Figura 89), que, na prática, executa de forma similar a anterior, com a diferença de que os sinais não estão organizados alfabeticamente, mas sim por categorias, conforme ilustrado na Figura 90.

Figura 86 - *Baby Sign Language Dict*: tela inicial da aplicação



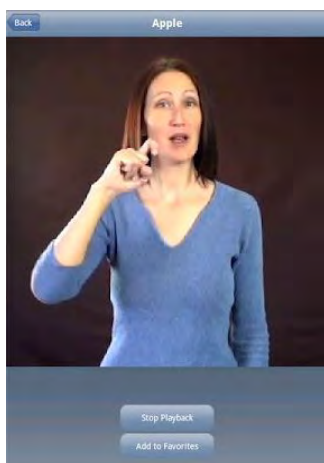
Fonte: Baby Sign Language Dict (2012)

Figura 87 - *Baby Sign Language Dict*: Lista de palavras do ASL Dictionary



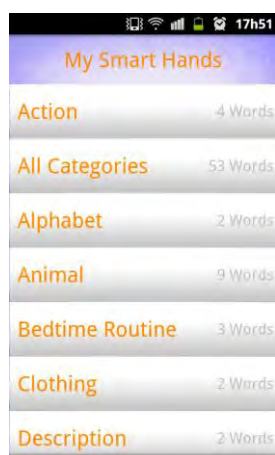
Fonte: Baby Sign Language Dict (2012)

Figura 88 - Baby Sign Language Dict: Video explicando a execução do sinal



Fonte: Baby Sign Language Dict (2012)

Figura 89 - Baby Sign Language Dict: Mostra as palavras separadas por categorias



Fonte: Baby Sign Language Dict (2012)

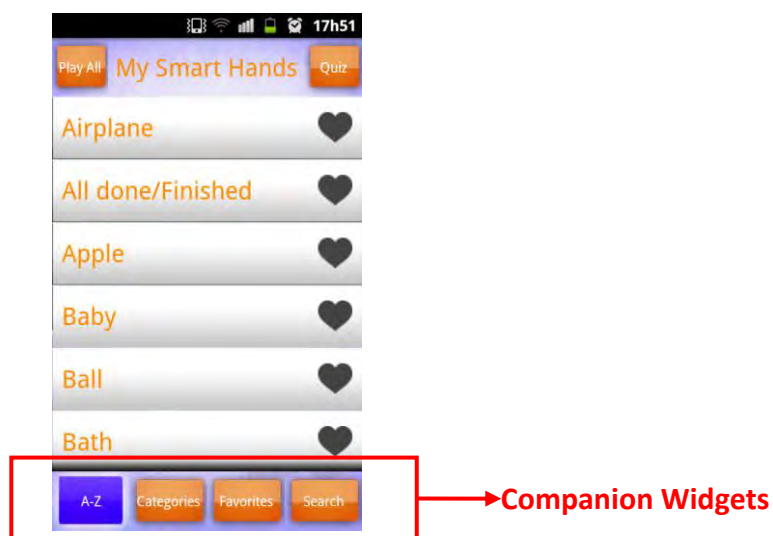
Figura 90 - Baby Sign Language Dict: lista de sinais



Fonte: Baby Sign Language Dict (2012)

No *Baby Sign Language Dict*, os principais recursos do aplicativo estão agrupados no *Bottom Bar* desta tela da aplicação como ilustra a Figura 91.

Figura 91 - Baby Sign Language Dict: Bottom Bar



Fonte: Baby Sign Language Dict (2012)

### 3.1.9. ASL American Sign Language

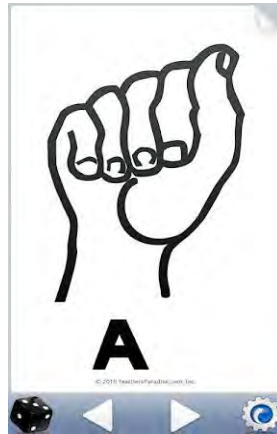
Esta ferramenta<sup>21</sup>, que não requer acesso à Internet para seu funcionamento, mostra um desenho das letras do alfabeto em língua de sinais, no caso em ASL, e o som da letra (ASL American Sign Language, 2012). A opção *default* da aplicação é mostrar as letras sequencialmente. Para isso, o usuário navega por meio das setas para direita e esquerda, que estão no menu inferior. Por exemplo, a Figura 92 e Figura 93 ilustram em ASL as letras “A” e “B”, respectivamente. Outras opções de uso são por meio do “dado”, que realiza um sorteio aleatório entre as letras do alfabeto, e uma opção de configuração, com outras opções, conforme Figura 94.

<sup>21</sup> Disponível para download em:

<[https://play.google.com/store/apps/details?id=com.teachersparadise.aslamericansignlanguage&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.teachersparadise.aslamericansignlanguage&hl=pt_PT)>

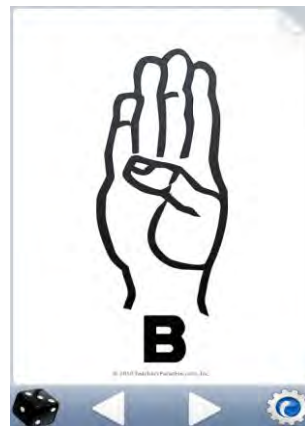


Figura 92 - ASL American Sign Language: letra "A" em ASL



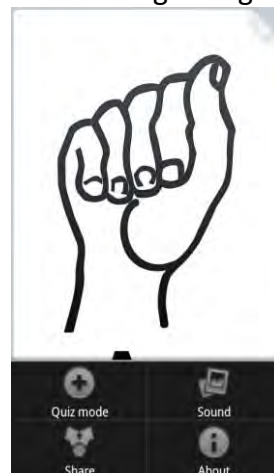
Fonte: Google Play (2012)

Figura 93 - ASL American Sign Language: letra "B" em ASL



Fonte: Google Play (2012)

Figura 94 - ASL American Sign Language: configuração



Fonte: Google Play (2012)

### 3.2. Aplicações interativas para dispositivos móveis

Além das aplicações com suporte à língua de sinais, seja na forma de vídeo, avatar ou escrita de sinais, nessa seção estão descritas outras aplicações para dispositivos móveis com o intuito de analisar características relacionadas à funcionalidade, interface e interação com o usuário em situações em que ele pode interagir livremente para construir algum desenho, por exemplo. Tal estudo se fez necessário porque o *SignSMS* permite a criação livre da escrita de sinais utilizando os diferentes símbolos do *SignWriting*. E, para isso, o usuário deverá poder escolher os símbolos e posicioná-los conforme a escrita desse sinal.

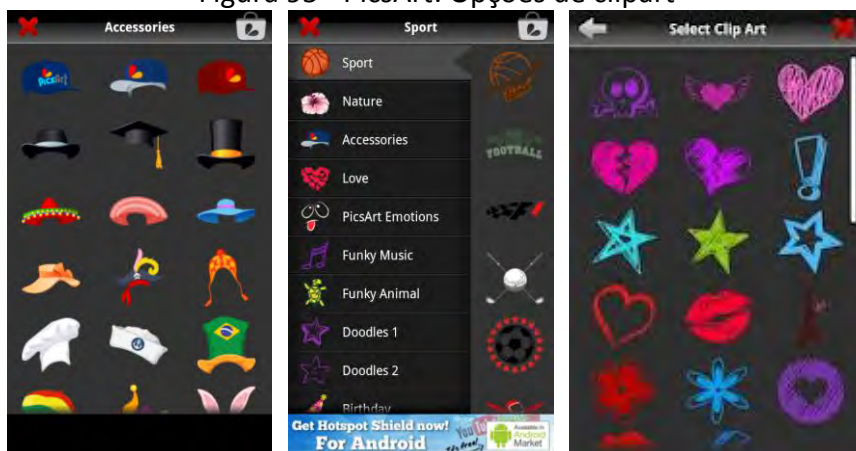
São descritas as seguintes aplicações:

- *PicsArt*
- *PicsArt for Kids*
- *PicSay*

#### 3.2.1. PicsArt

O *PicsArt*<sup>22</sup>, que requer conexão à *Internet*, é um aplicativo que combina editor de fotografias com editor de imagens. Com relação às fotos, é possível incluir efeitos gráficos de cor e textura bem como incluir objetos gráficos. Quanto ao editor de imagens, podem-se criar desenhos a mão livre ou a partir de figuras disponibilizadas na biblioteca de *cliparts*, por exemplo. A Figura 95 ilustra opções da biblioteca de *clipart*, a Figura 96 mostra efeitos de texto, a Figura 97 mostra as opções de balões de diálogo.

Figura 95 - PicsArt: Opções de clipart

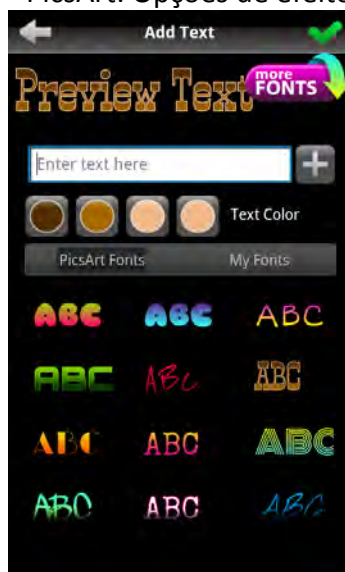


Fonte: *PicsArt* (2012)

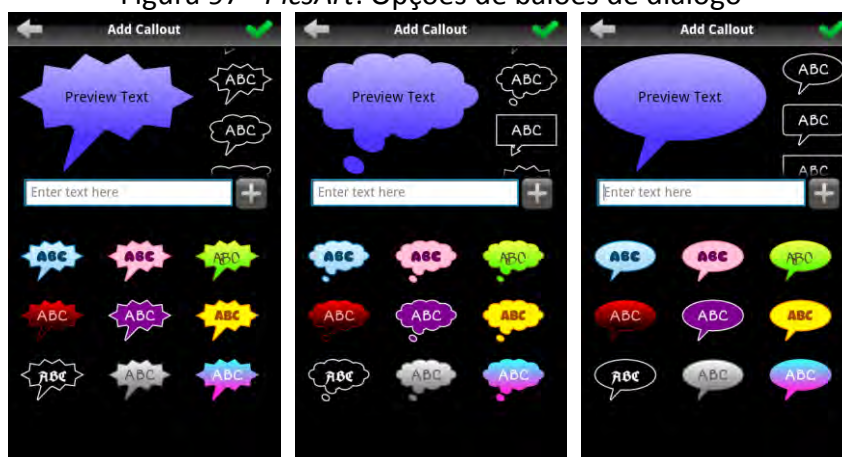
<sup>22</sup> Disponível para download em:

< [https://play.google.com/store/apps/details?id=com.picsart.studio&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.picsart.studio&hl=pt_BR) >

Figura 96 - PicsArt: Opções de efeitos de texto



Fonte: *PicsArt* (2012)

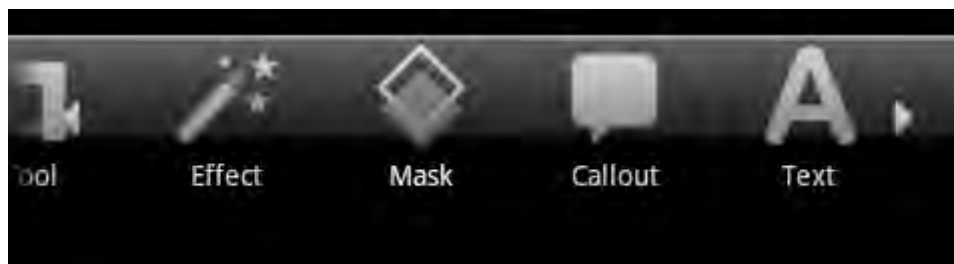
Figura 97 - *PicsArt*: Opções de balões de diálogo

Fonte: *PicsArt* (2012)

As opções de aumentar e diminuir tamanho, rotacionar, alterar cor são habilitadas nos elementos constituídos por figuras disponibilizadas nas bibliotecas da aplicação, desde que ainda estejam selecionadas. Após, o funcionamento torna-se similar ao do *Paint*, por exemplo, no qual a figura fica “carimbada”. Nesse sentido, o aumentar e diminuir passa a funcionar como o *zoom*, que aumenta ou diminui a área de desenho do aplicativo.

No que se refere à interface, as ações são dispostas em menus horizontais ou de tela cheia, quando desmembrados. A Figura 98 ilustra o menu para criar um desenho enquanto a Figura 96 e Figura 97, já citadas, mostram opções de tela cheia.

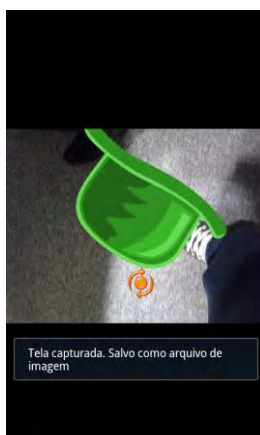
Figura 98 - *PicsArt*: Menu de Desenho



Fonte: *PicsArt* (2012)

No que se refere à interação do usuário, as opções são selecionadas com clique simples sobre o opção do menu, por exemplo. No caso de alteração no tamanho de alguma figura, essa é feita mantendo o objeto selecionado com os dedos e arrastado para dentro ou para fora, de forma similar como é feito com o *mouse*. No caso de rotação, como objeto selecionado, gira-se para direita ou para esquerda, conforme Figura 99, onde é possível verificar o símbolo que indica essa ação.

Figura 99 – *PicsArt*: Exemplo de manipulação de objetos



Fonte: *PicsArt* (2012)

### 3.2.2. *PicsArt for Kids*

O *PicsArt for Kids*<sup>23</sup> é um editor gráfico voltado para crianças. Possui opções para desenhar a mão livre dentro de um cenário, para pintar cenários/figuras, para desenhar sobre um molde, além de poder acessar sua galeria de desenhos. A Figura 100 ilustra a tela inicial do aplicativo.

<sup>23</sup> Disponível para download em:

< [https://play.google.com/store/apps/details?id=com.picsart.kids&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.picsart.kids&hl=pt_BR) >

Figura 100 - PicsArt for kids: Tela inicial



Fonte: *PicsArt for Kids* (2012)

A primeira opção do menu corresponde a construção de desenhos a mão livre. Para isso, a criança deve escolher um dentre os quatro cenários apresentados. O cenário é, então carregado, e a partir das opções de menu, a criança começa a elaborar seu desenho. A Figura 101 ilustra essa tela onde as opções de menu são dispostas verticalmente. À esquerda da tela, temos o menu de cores, que possui mais cores do que é exibido. Desta forma, com o dedo, arrastando para cima ou para baixo, podem ser acessadas as demais opções de cores. O menu da direita contém opções para desenhar e editar. Possui pincel para desenhar (Figura 102), borracha para apagar manualmente os desenhos, lixeira para apagar automaticamente todos os desenhos feitos naquele cenário, disquete para salvar o desenho na galeria. Além desses, possui um “dado” que altera aleatoriamente o tipo de “ponta do Pincel”, permitindo desenhar com efeitos diferentes. A seta para direita passa para o próximo cenário, sendo que os desenhos realizados são perdidos, caso não tenham sido salvos. Para voltar a um cenário é preciso passar por todos. Ou seja, não há opção para voltar. Também não há opção de alterar o tamanho do cenário e nem dos objetos que são desenhados pela criança tampouco de rotacionar elementos.

Figura 101 - PicsArt for kids: Cenário com desenho

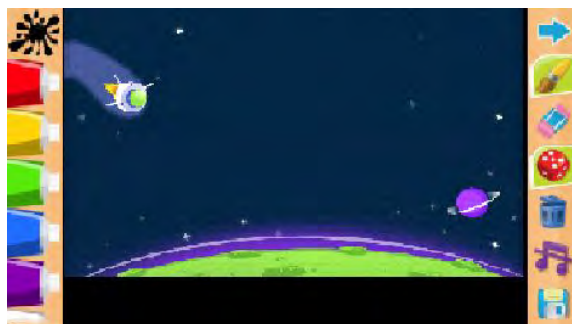
*PicsArt for Kids (2012)*

Figura 102 - PicsArt for Kids: Opções de pincel

*PicsArt for Kids (2012)*

Na opção de Pintar cenários/figuras, após escolhido o cenário/figura, deve-se escolher a cor desejada e clicar na parte do desenho para que a mesma seja pintada. A Figura 103 ilustra parte desse processo. Pode-se aplicar recurso de *zoom* clicando em qualquer parte do cenário fazendo movimentos de *pinch* e *spread*.

Figura 103 - PicsArt for kids: Colorindo cenários

Fonte: *PicsArt for kids (2012)*

Outra opção do aplicativo refere-se a desenhar sobre um molde. Nesse caso, são apresentados exemplos de desenhos e a criança pode desenhar em cima dos mesmos. As

opções de desenho são similares as já descritas na opção de desenho à mão livre (Figura 104).

Figura 104 - *PicsArt for Kids*: Desenhando a mão livre

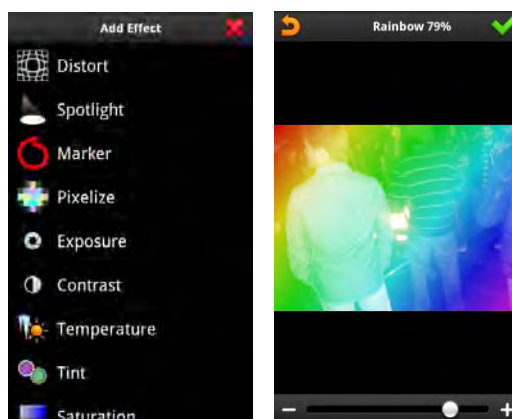


Fonte: *PicsArt for kids* (2012)

### 3.2.3. *PicSay*

O *PicSay*<sup>24</sup>, que não necessita de acesso a internet, é um aplicativo que serve para edição de fotos no qual é possível tirar uma foto ou usar alguma foto da galeria do usuário. Ele possui ferramentas para sobrepor figuras pré-definidas sobre a imagem original, ferramentas para aplicar filtros na imagem, conforme Figura 105, e recursos para rotacionar a foto, alterar contraste, distorcer a foto entre outras opções que podemos ver na Figura 106. O menu não fica visível na tela, o que evita um possível desperdício de espaço. Na Figura 107 e Figura 108 há exemplos da ferramenta de inserção de texto e de figuras que podem ser sobrepostas às fotografias.

Figura 105 - *PicSay*: Efeitos



Fonte: *PicSay* (2012)

<sup>24</sup> Disponível para download em:

<[https://play.google.com/store/apps/details?id=com.shinycore.picsayfree&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.shinycore.picsayfree&hl=pt_BR)>

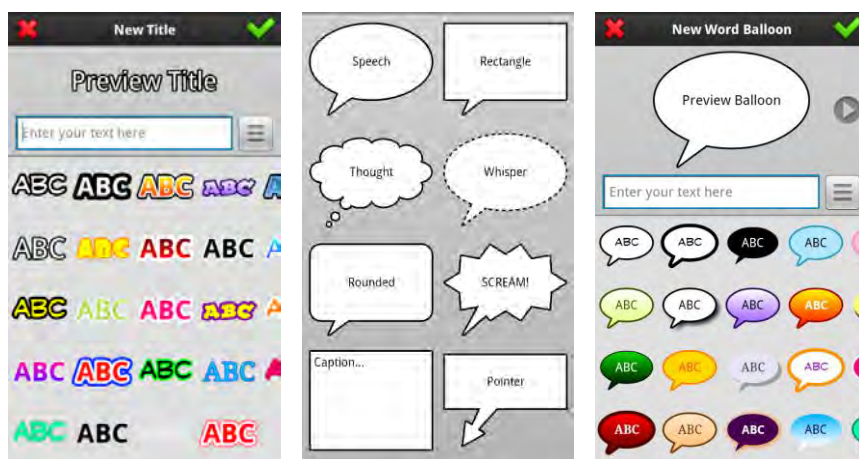


Figura 106 - PicSay: Opções do editor de imagens Opções do editor de fotos



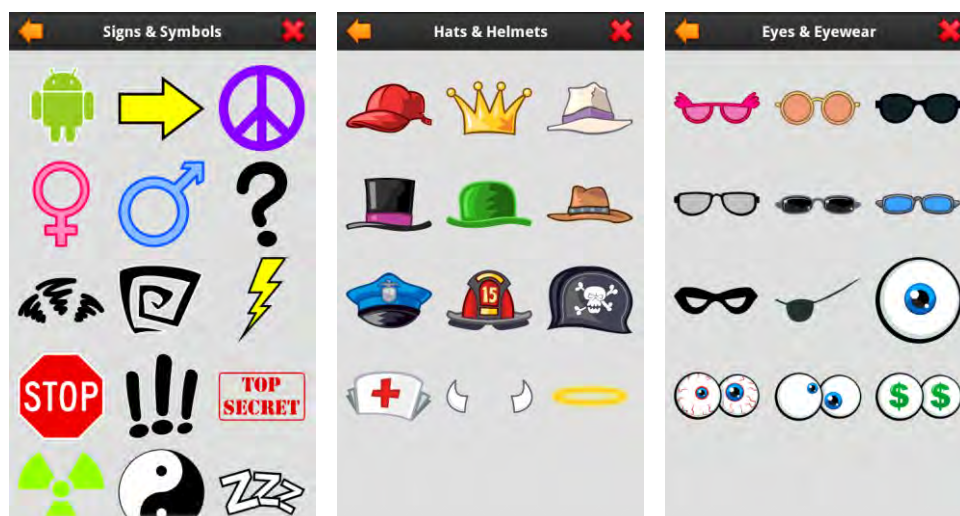
Fonte: PicSay (2012)

Figura 107 - PicSay: Inserção de texto



Fonte: PicSay (2012)

Figura 108 - PicSay: ClipArt



Fonte: PicSay (2012)



### 3.3. Padrões

Em relação aos padrões de *design* utilizados, podemos observar que, dentre os cinco principais citados na seção 2.5, foram identificados quatro em pelo menos uma das aplicações estudadas, o que ressalta a importância na utilização de padrões no desenvolvimento dos aplicativos. Segue uma lista de imagens que capturam os padrões detectados na implementação de cada aplicativo, na seguinte ordem: *Action Bar*, *Dashboard*, *Quick Actions*, *Search Bar*.

- *Action Bar*: na aplicação *Signing Savvy*, podemos perceber que as opções de *Login* e *Logout* se mantêm estáticas na *Action Bar* deste aplicativo conforme Figura 109 e Figura 110.

Figura 109 - *Signing Savvy*: *Action Bar*



Fonte: Signing Savvy (2012)

Figura 110 - signing savvy: *Action bar* 2



Fonte: Signing Savvy (2012)

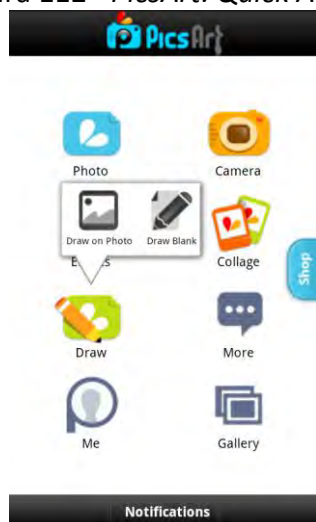
- *Dashboard*: ainda no *Baby Sign Language Dict*, a tela inicial da aplicação apresenta atalhos para as funções de consulta ao dicionário em ASL e a opção de consulta aos vídeos explicativos e ainda oferece um link para que o usuário possa adquirir a versão completa deste aplicativo. A Figura 111 ilustra o padrão *Dashboard* utilizado no aplicativo.

Figura 111 - *Baby Sign Language Dict: Dashboard*



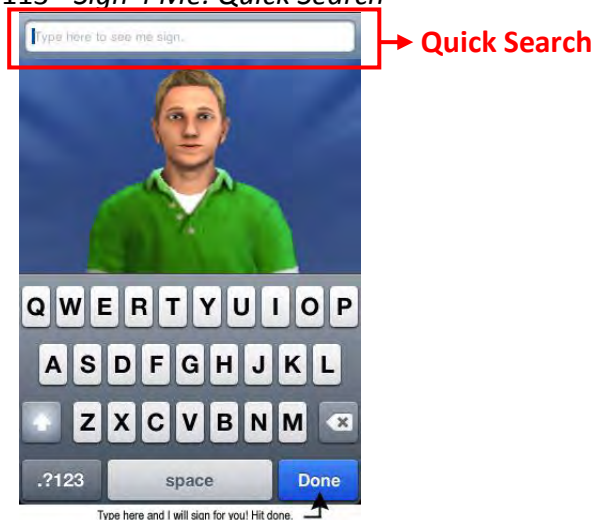
Fonte: *Baby Sign Language Dict* (2012)

- *Quick Actions*: no *Pics Art* a utilização deste padrão foi encontrado como subcomponente em um dos menus da aplicação e pode ser visto na Figura 112.

Figura 112 - *PicsArt: Quick Action*

Fonte: *PicsArt* (2012)

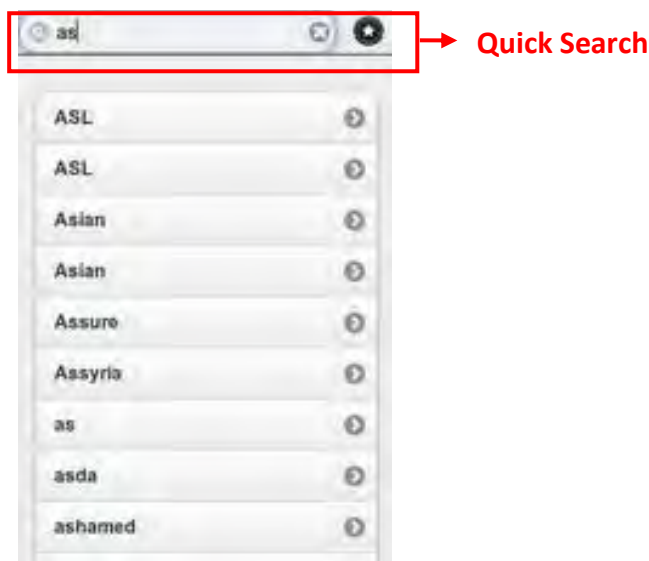
- *Quick Search*: no *Sign 4 Me* este recurso é encontrado na tela de pesquisa das palavras no idioma inglês que serão traduzidas para ASL pelo avatar. A *Quick Search* fica incorporada na *Action Bar* no topo da aplicação, como mostra a Figura 113.

Figura 113 - *Sign 4 Me: Quick Search*

Fonte: *Sign 4 Me* (2012)

No *SkySigns ASL Dictionary* este recurso é encontrado na tela de pesquisa das palavras no idioma inglês que serão mapeadas para o sinal correspondente utilizando SignWriting para ASL. Assim como no *Sign 4 Me* a *Quick Search* fica incorporada na *Action Bar* no topo da aplicação, como ilustra a Figura 114.

Figura 114 - SkySigns ASL Dictionary: Quick Search



Fonte: Skysign ASL Dictionary (2012)

### 3.4. **Comparativo das aplicações com suporte a escrita de sinais**

Nesta seção destacaremos características das aplicações analisadas como intuito de identificar recursos que foram importantes ao projeto de interface-interação do *SignSMS*.

De acordo com Souza & Pinto (2004), um dos grandes problemas enfrentados pela comunidade surda é de que:

*não poder se expressar através da escrita de sua própria língua, a língua de sinais - LS. Em função disso, sua comunicação se estabelece quase que exclusivamente de forma presencial, de forma que os surdos não podem usufruir totalmente das novas tecnologias como, por exemplo, o correio eletrônico e a Internet.*

Partindo desse desafio, a análise das principais funcionalidades e características das aplicações descritas nesta seção torna-se fundamental para que o desenvolvimento do *SignSMS* traga à comunidade surda um software não só inovador, mas que agregue valor nas atividades relacionadas à comunicação a distância e que, preferencialmente, não se restrinja às limitações presentes nos computadores e dispositivos móveis mais antigos, por exemplo. A comunicação, por meio de mensagens em escrita de sinais através das novas tecnologias, pode se tornar mais útil às pessoas surdas. Portanto, para que sejam minimizados problemas de usabilidade com relação às funcionalidades e interface do

*SignSMS*, apresentamos esse paralelo entre as funcionalidades mais recorrentes dos sistemas que utilizam *Signwriting*.

Campos (2001) e Souza & Pinto (2004) fazem uma comparação entre alguns softwares que fazem uso do *SignWriting*, conforme ilustrados na Tabela 3 (ver seção 3.1.1) e Tabela 4.

Tabela 4- comparativo das funcionalidades de softwares que fazem uso do *SignWriting*

Software / Recurso	SignWriter	SignEd SignSim SignTalk	SWEdit	SignBank	Sign WebMessage
Cria sinais	x	x	x	x	
Edita sinais	x	x	x	x	
Possui dicionário de sinais	x	x	x	x	x
Consulta o dicionário a partir da língua de sinais		x		x	x
Consulta o dicionário a partir da língua oral	x	x	x	x	x
Importa e exporta sinais em SWML			x		

Fonte: Souza e Pinto (2004)

A partir deste comparativo, podemos observar que a maioria das aplicações que faz uso do *Signwriting* e de dicionários de sinais, ainda utiliza a língua oral como meio para a realização das buscas no banco de sinais. As ferramentas do Pacote *Sign*, o *SignBank* e o *Sign WebMessage* possuem módulo de consulta também por sinais. As opções de criação, edição e armazenamento foram identificadas na maioria dos aplicativos analisados.

Ainda, no que se refere à escrita dos sinais, pode-se observar que as aplicações do Pacote *Sign* não permitem a criação direta dos sinais, de forma que a mesma ocorre em uma área de construção e depois é carregada para a área de texto em *Signwriting*. Essa estratégia foi utilizada para que os sinais pudessem ser criados e carregados somente após a escrita do mesmo estar completa. Como restrição, é que o sinal escrito não podia ser mais alterado, a menos que fosse apagado da área de texto e novamente escrito na área de construção. Assim como as aplicações do Pacote *Sign*, o *Sign WebMessage* também necessita de uma área temporária para a criação dos sinais antes de inseri-los no texto, conforme já explicado durante a descrição do aplicativo.

O *SW-Edit*, por sua vez, permite que os símbolos sejam inseridos na área de texto diretamente, sem que o usuário necessite utilizar uma área temporária. Conforme já descrito, a inserção dos símbolos pode ser feita de duas maneiras: clicando em um símbolo na área de símbolos e arrastando para uma célula no texto ou selecionando uma célula na área do texto e dando um duplo clique sobre o símbolo desejado para que este seja inserido na célula previamente selecionada.

No *SignSMS* está prevista a funcionalidade de escrita de sinais por meio do menu de grupos de símbolos, conforme organização do *SignWriting* e do menu de movimentação, que inclui arrastar um símbolo para cima, para baixo, esquerda ou direita, bem como rotacioná-lo ou espelhá-lo. Desta forma, o usuário poderá escrever os sinais no *SignSMS*.

## 4. PROJETO

Neste Trabalho de Conclusão de Curso foi desenvolvido um protótipo de ferramenta para envio de texto escrito em língua de sinais por meio de telefone móvel, denominado de *SignSMS*. Para tanto, foi definido como público-alvo, usuários que desejam se comunicar a distância por meio da escrita de língua de sinais e que estejam inseridos em contexto de mobilidade tecnológica. Podem ser conhecedores da escrita de sinais ou usuários que estejam aprendendo a se comunicar por meio dessa escrita.

A seguir são apresentadas informações sobre a construção do protótipo.

### 4.1. *Arquitetura do SignSMS*

Para a elaboração da arquitetura do projeto, inicialmente foi elaborado um estudo sobre os padrões e boas práticas de implementação, de forma a permitir uma construção do projeto mais coesa. Este estudo está apresentado nas seções 2.4 e 2.5.

Desta forma, as seções que seguem tratam dos padrões, princípios e práticas de design que foram adotados para o desenvolvimento do *SignSMS* e dos principais componentes da estrutura do projeto.

#### 4.1.1. Padrões, princípios de *design* e boas práticas no *SignSMS*

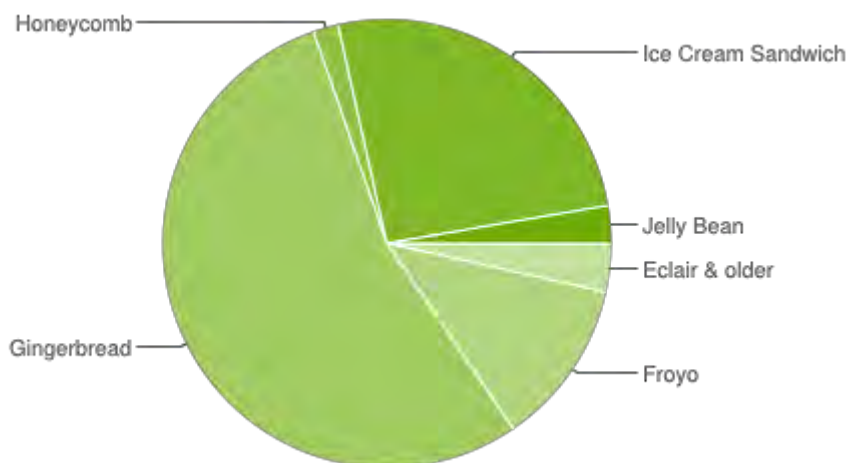
Esta seção descreve os padrões e práticas de desenvolvimentos apresentados na cartilha do W3C (seção 2.4) e os padrões de projeto (seção 2.5) que foram utilizados como base na implementação do *SignSMS*.

Sobre a cartilha do W3C (W3C MOBILE, 2012) foram considerados pertinentes ao escopo do projeto:

- Faça seu projeto para uma *Web* única: dentro deste conceito aplicamos as seguintes práticas:
  - Capacidades: consideramos que a aplicação rodará com o SO base *Froyo*, que é a versão 2.2 da plataforma *Android*, esta decisão foi baseada na representatividade de usuários que ainda utilizam esta versão, conforme Figura 115. Pode-se notar que a maior parte dos usuários utiliza a versão *Gingerbread*, que é a versão 2.3.3 do *Android*, uma característica

importante da plataforma *Android* é que uma versão superior consegue utilizar grande parte das funcionalidades das versões anteriores, porém o contrário não é possível, deste modo, ao escolhermos a versão 2.2 do *Android*, abrimos um leque para que uma gama maior de usuários tenha acesso à aplicação aqui proposta.

Figura 115 – Distribuição dos usuários por versão da plataforma Android



Fonte: Android Developers<sup>25</sup>

- Problemas na implementação: a plataforma *Android* possuiu a funcionalidade “retornar” nativa, portanto este recurso foi reutilizado para a navegação entre as telas da aplicação.
- Testes: durante o período de implementação, foram feitos testes de usabilidade tanto nos emuladores das IDEs de desenvolvimento *Eclipse* e *MotoDev Studio* emulando as APIs 2.2 e 2.3.3 quanto em *hardware* real. Dois modelos de celular foram utilizados para os testes, conforme descrito na seção que trata dos recursos utilizados para o desenvolvimento da aplicação.
- Confie nos padrões Web: dentro deste conceito aplicamos as seguintes práticas:
  - Compatibilidade de formato: este item vem de encontro ao item “Capacidades” abordado nesta mesma seção.

<sup>25</sup> Distribuição corrente dos dispositivos Android por versão de plataforma. Mais detalhes em: <http://developer.android.com/about/dashboards/index.html>



- Seja prudente com a limitação dos dispositivos: deste item consideramos os seguintes pontos de atenção:
  - Tipografia: a fonte utilizada para a escrita do texto apresentado na interface é a fonte padrão da plataforma *Android*.
  - Uso de cores: adotamos como critério para o *design* dos *layouts* da aplicação o uso das cores branco, preto e cinza, uma vez que a biblioteca de símbolos do *SignWriting* são representados em preto e branco. Desta forma o *layout* não ganha destaque em relação aos símbolos. Os botões e menus, por outro lado, apresentam cores distintas, as cores foram selecionadas para agrupar os botões por escopo de funcionalidade de forma que a cor verde foi utilizada no menu superior, que contém opções de navegação entre as páginas da mensagem, e as cores vermelho e laranja que estão no menu inferior e definem, respectivamente, opções de manipulação do símbolo (movimentação, rotação, espelhamento) e opções de ações do usuário sob os símbolos (apagar, duplicar, limpar tela, refaz/desfazer ação, trocar a ordem dos símbolos).
- Aperfeiçoe a navegação: no que se refere à navegação, destacamos:
  - Barra de navegação: no topo da tela onde a mensagem é escrita pelo usuário existe um menu para a navegação entre as páginas, existe também opções para adicionar e remover nova páginas de acordo com a necessidade do usuário.
  - Navegação: a navegação entre as telas da aplicação é feita por transição de telas, onde cada tela disponibiliza ao usuário opções de navegação para a tela anterior e fechamento da tela atual. Caso o usuário selecione a opção para escrever a mensagem e navegue por mais de um menu de símbolos é disponibilizado a opção de retornar diretamente à tela de edição da mensagem.
- Teste antes os gráficos e cores: destacamos o item abaixo:
  - Contraste de cores: vem ao encontro do item “Uso de cores” abordado nesta seção.

- Faça um tamanho reduzido: levando em conta a limitação apresentada pelos dispositivos móveis no que se refere ao tamanho da área de visualização destacamos o seguinte critério de *design*:
  - Rolagem de tela: adotamos apenas a rolagem vertical nas telas que apresentam os símbolos a serem inseridos na mensagem. Ainda, para evitar que houvesse rolagem horizontal nos menus de ações sobre os símbolos, foi adotada a estratégia de “peso”, ou seja, a partir do tamanho do espaço e dos elementos do menu, cada elemento recebeu um peso de forma que todos coubessem no espaço disponível, que pode variar de acordo com o modelo do telefone.
- Pense nos usuários da *Web* móvel: dentre os itens englobados neste item, ressaltamos:
  - Clareza: os elementos que compõem as telas do *SignSMS* foram escolhidos de forma a facilitar a navegação e o aprendizado do usuário no uso da aplicação. Para tanto, foram utilizados elementos comuns a outras aplicações, que suportam a escrita de sinais.

Já com relação aos padrões de *design* citados na seção 2.4, os elementos empregados na modelagem do projeto do *SignSMS* de acordo com o padrão utilizado foram:

- Encante-me: relacionado aos itens visuais empregados na construção da interface, tem-se:
  - Delicie-me de maneiras surpreendentes: mais do que elementos visuais bonitos, a aplicação proposta preocupa-se com a simplicidade e consistência dos layouts de interface. Por isso, a aplicação como um todo foi pensada para possuir um *layout* sem poluição gráfica, de forma que ainda houvesse uma identidade visual entre ações que tivessem funcionalidades da mesma ordem.
  - Os objetos reais são mais divertidos do que botões e menus: os botões e menus apresentam representação gráfica da funcionalidade proposta, de modo a tornar a navegação mais intuitiva e satisfatória aos usuários. Por

isso, os desenhos de ícones adotados lembram suas funcionalidades para permitir que o usuário possa lembrá-los somente visualizando-os, sem necessidade de experimentar antes.

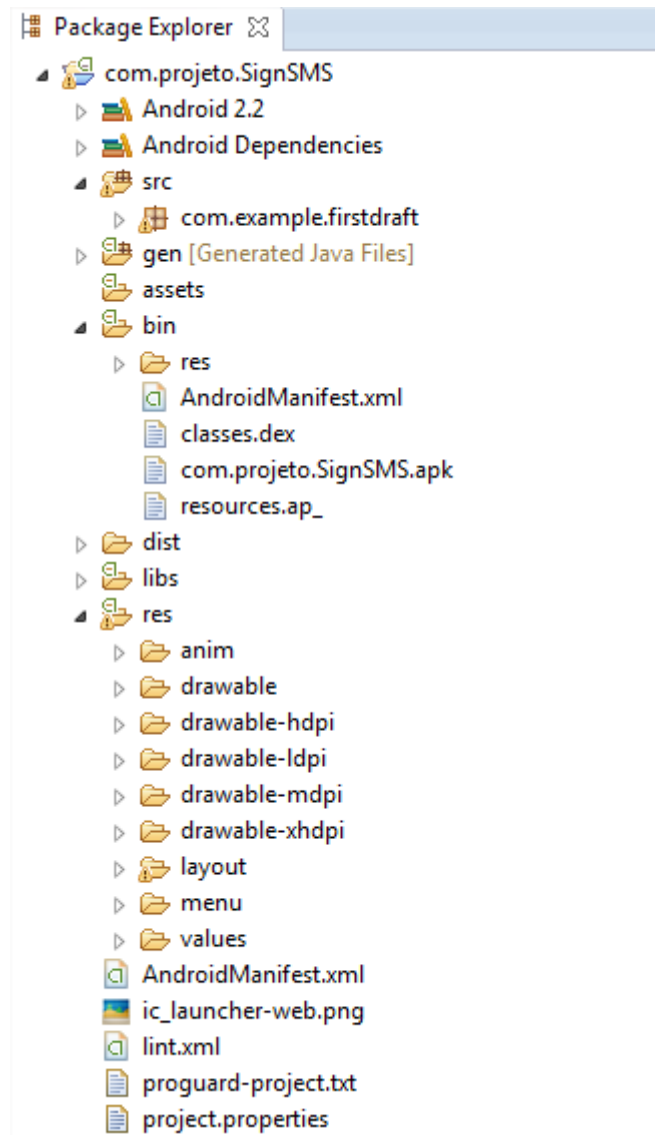
- Simplifique minha vida: quanto aos itens referentes a este padrão, ressaltamos:
  - Fotos são mais rápidas do que palavras: assumindo a ideia de foto como desenhos, este item vem ao encontro do item “Os objetos reais são mais divertidos do que botões” abordado nesta mesma seção.
  - Eu deveria sempre saber onde estou: vem ao encontro do item “Navegação”, também abordado nesta seção.

#### **4.1.2. Estrutura do projeto**

Esta seção descreve a estrutura de diretórios e os principais arquivos gerados na criação do *SignSMS* utilizando o ambiente de desenvolvimento *MotoDev Studio*.

O *Package Explorer* (localizado na perspectiva *Java* no *MotoDev Studio*) exibe todos os componentes da aplicação *Android* criada, conforme Figura 116.

Figura 116 - MotoDev Studio: perspectiva Package Explorer do projeto



Dentre os arquivos pertencentes ao projeto, os itens a serem observados incluem:

- Pasta *src*: inclui o pacote com as classes desenvolvidas para a aplicação, especificamente *com.example.firstdraft*.
- *R.java*: arquivo gerado automaticamente pelo *Android Developer Tools*, que representa as constantes necessárias para acessar diferentes recursos da aplicação *Android*.
- Pasta *bin*: gera automaticamente alguns arquivos após a compilação do projeto, dentre eles o arquivo *.APK* que é o formato de arquivo utilizado para a instalação

do aplicativo nos dispositivos que executam o sistema *Android*, especificamente *com.projeto.SignSMS.apk*.

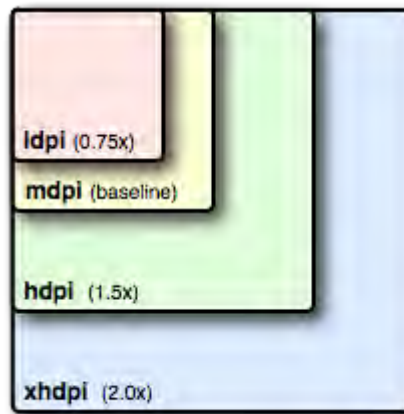
- Bibliotecas de referência: contém o arquivo *android.jar*, que é o arquivo *jar* da classe do tempo de execução do *Android*, localizado no *Android SDK*.
- Pasta *res*: contém os recursos da aplicação, como imagens, *layouts* de telas. Podendo também possuir arquivos para internacionalização da aplicação, o *SignSMS* em sua primeira versão tem somente suporte à Língua Portuguesa (Brasileira). Esta pasta possui três subpastas:
  - *drawable*: contém os arquivos de imagens que serão utilizados na aplicação, dependendo da resolução da imagem armazenada, esta se destina à uma pasta *drawable* específica, a Figura 117 ilustra a diferença entre as densidades de acordo o tamanho dos *drawables* (imagens armazenadas nas pastas).
    - *drawable-ldpi*: imagens com densidade baixa (36x36 *pixels*)
    - *drawable-mdpi*: imagens com densidade média (48x48 *pixels*)
    - *drawable-hdpi*: imagens com densidade alta (72x72 *pixels*)
    - *drawable-xhdpi*: imagens com densidade extra alta (92x92 *pixels*)

A Tabela XXX mostra a relação dos símbolos utilizados no *SignSMS*, de acordo com o grupo pertencente, no total a aplicação dispõe de 2554 símbolos:

Tabela 5- SignSMS: quantidade de símbolos na aplicação

<b>Mão – 1208 símbolos</b>	
<b>Toque – 34 símbolos</b>	
<b>Movimento – 638 símbolos</b>	
<b>Expressão facial – 158 símbolos</b>	
<b>Corpo – 466 símbolos</b>	
<b>Pontuação – 50 símbolos</b>	
<b>Total</b>	<b>2554</b>

Figura 117 - Tamanho relativo das imagens de acordo com cada densidade



Fonte: [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)

- *layout*: contém os arquivos no formato *XML* de *layout* para construir as telas da aplicação.
- *values*: contém os arquivos *XML* utilizados para armazenar as variáveis do tipo *String* que podem mudar de acordo com alguma configuração, como por exemplo, seleção de idioma.
- *AndroidManifest.xml*: é o arquivo principal do projeto e centraliza as configurações da aplicação. Neste arquivo definimos quais as atividades do projeto serão executadas, qual o nível da *API* (suporte mínimo) que a aplicação pode executar, permissões de acesso da aplicação ao dispositivo do usuário, tais como, acesso a câmera, *internet*, agenda, etc.

### 4.1.3. Descrição dos componentes

Esta seção descreve os principais componentes definidos pela *SDK Android* utilizados no *SignSMS*.

- *Activity*: é uma tela de interface com o usuário. Os aplicativos podem definir uma ou mais *activities* para lidar com as diferenças fases do programa. Cada *activity* é responsável por salvar seu próprio estado, para que ele possa ser restaurado mais tarde, como parte do ciclo de vida da aplicação.
- *Intent*: é um mecanismo para descrever uma ação específica como, por exemplo, “telefonar”, “enviar e-mail”, etc. No *Android*, uma vez que todas as

ações passam por *intents*, podem-se ter mais oportunidades para substituir ou reutilizar componentes. Por exemplo, há um *intent* de “enviar um e-mail”. Se a aplicação precisar enviar um e-mail, pode-se chamar esse *intent*. Ou se estiver escrevendo uma nova aplicação para envio de e-mail, pode-se registrar na *activity* a manipulação dessa *intent* e substituir o programa de *email* padrão. Consequentemente, na próxima vez que for enviado um e-mail, haverá a opção de usar o programa criado ao invés do programa padrão.

- *View*: este componente é uma estrutura de dados que representa uma área retangular limitada da tela do dispositivo. Por meio dele, pode-se obter informações como medidas de largura e altura, desenho, mudança de foco, capacidade de rolagem (*scrolling*) e captura de comandos (teclas dos dispositivos) percebidos por aquela área específica. A classe *View* é extensível e serve de base para outros *widgets*. Ainda, a plataforma vem equipada com vários *widgets* já implementados para o desenvolvedor utilizar em suas aplicações.
- *ImageView*: apresenta uma imagem arbitrária, como um ícone. A classe *ImageView* pode carregar imagens de várias fontes (tais como recursos ou provedores de conteúdo), cuida do dimensionamento da imagem de modo que possa ser usada em qualquer gerenciador de *layout*, e oferece opções de exibição, como escala e de tingimento.
- *ImageButton*: exibe um botão com uma imagem (em vez de texto) que pode ser pressionado ou clicado pelo usuário. Por padrão, um *ImageButton* parece um *regular Button*, com o fundo padrão do botão que muda de cor durante os diferentes estados de botão.
- *TextField*: é um campo de texto que permite que o usuário digite o texto em sua *app*. Ele pode ser uma linha única ou multi-linha. Tocar em um *TextField* coloca o cursor e exibe automaticamente o teclado. Além de digitação, campos de texto permitem uma variedade de outras atividades, como a seleção de texto (cortar, copiar, colar) .

## **4.2. Modelagem**

Esta seção descreve as principais funcionalidades do *SignSMS*. Para tal, será utilizado um método de IHC que indica a descrição de perguntas, cenários, signos e modelos de tarefas para descrever as ações do usuário sobre o sistema.

### **4.2.1. Perguntas**

De acordo com Paula & Barbosa (2003), as perguntas tem como objetivo direcionar a descrição dos cenários, evitando situações de ambiguidade ou incompletude. Foram definidas as seguintes perguntas, que foram referenciadas nos cenários.

1. Como os sinais são escritos?
2. Que símbolos podem ser incluídos na escrita de um sinal?
3. Enquanto o sinal está sendo escrito, ele pode ser modificado?
4. Há diferentes símbolos para representar, mão, toque, movimento e expressão facial. Para alguns, é preciso espelhar, rotacionar, mudar a posição em combinação com outros símbolos. Essas ações são possíveis de serem realizadas?
5. Um sinal que está sendo escrito pode ser apagado?
6. Pode-se aumentar ou diminuir o tamanho dos símbolos ou de um sinal?
7. Pode-se escrever um texto com mais de um sinal?
8. Pode-se enviar mensagem com sinais para outros usuários?
9. Qual informação é necessária para enviar uma mensagem?
10. Pode-se consultar a agenda do telefone para buscar o email do destinatário da mensagem?
11. É possível encaminhar a outros destinatários uma mensagem recebida?
12. É possível responder uma mensagem recebida?
13. É possível indicar o assunto de uma mensagem?



14. Existe algum requisito de conexão para o que o usuário saiba se tem novas mensagens?

15. Existe algum requisito de conexão para o que o usuário envie mensagens?

#### 4.2.2. Cenários

Os cenários descrevem as principais ações, que estão previstas para um sistema, do ponto de vista do usuário, e não incluem ações de interface. Conforme Barbosa & da Silva (2010, p. 183):

*(...) é basicamente uma história sobre pessoas realizando uma atividade. É uma narrativa, textual ou pictórica, concreta, rica em detalhes contextuais, de uma situação de uso da aplicação, envolvendo usuários, processos e dados reais ou potenciais.*

Para o entendimento das funcionalidades do *SignSMS*, foram descritos os seguintes cenários, nos quais já foram incluídas as perguntas da seção 4.2.1.

- Cenário 1: Escrever sinais. Ana deseja escrever o sinal da palavra CASA em *SignWriting* [1]. Para isso, ela precisa indicar os elementos que compõem o sinal, que são mãos e toque [2]. Precisa posicioná-los [3, 4]. No caso das mãos, essas devem estar rotacionadas em ângulos de 45º, sendo que uma delas deverá estar espelhada [3, 5]. Depois, verificou que precisaria escrever outros sinais [8], e viu que o *SignSMS* também permite incluir expressão facial e movimento [2]. Quando Ana estava escrevendo o sinal de CASA, verificou que o símbolo de toque ficou embaixo de um dos símbolos de mão [3]. Por sorte, o sistema permite alterar a ordem dos símbolos, sem precisar que o mesmo seja apagado [5]. Ana pensou que para melhor enxergar o sinal, poderia aumentá-lo. Mas não encontrou essa opção. Ao perguntar para outro colega, foi informada de que o *SignSMS* não permite aumentar e nem diminuir símbolos e sinais [6].
- Cenário 2: Escrever mensagem. João precisa escrever um texto em *Signwriting* para enviar para a Ana [8]. Para isso, estava escrevendo cada sinal e enviando a ela até que um colega lhe disse que poderia escrever o texto incluindo vários sinais e fazer somente um envio de mensagem [7]. Durante a escrita da mensagem, João viu que, na pressa, escreveu um sinal de forma errada. Não precisou descartar a mensagem porque o *SignSMS* permite que sinais da

mensagem sejam selecionados e atualizados [1, 3], o que inclui sua remoção da mensagem, se assim desejar [5].

- Cenário 3: Enviar mensagem. João já escreveu o texto em *SignWriting* e precisa enviar para a Ana [8]. Para isso, precisou consultar sua agenda de contatos [10] e selecionar o email dela. Ana, ao receber a mensagem, viu que ela seria útil a outros amigos. Como o *SignSMS* permite encaminhar a mensagem a outros destinatários [11], pensou que teve sorte porque não precisaria escrever cada sinal novamente. Mas, antes de fazer isso, respondeu ao João perguntando se ele autorizava o envio da mensagem [12]. Ana alterou o assunto da mensagem e enviou sua consulta ao João [13]. Após a confirmação do João, Ana perdeu a conexão com a Internet e, por isso, não pode encaminhar a mensagem [15].
- Cenário 4: Receber mensagem. Maria aguardava o recebimento da mensagem da Ana e, para isso, ficou acessando o *SignSMS* por várias vezes. Vendo a situação, um colega lhe disse que, para isso, ela precisava estar conectada à Internet [14].

### 4.2.3. Tabela de signos

De acordo com Paula & Barbosa (2003), os cenários descrevem tarefas e apresentam (ou manipulam) informações. Essas informações são denominadas de signos, que podem ser simples ou compostos. Um signo composto é aquele que é descrito por mais de um signo. Ainda, podem ser do tipo aplicação, domínio ou transformado. Com exceção dos signos que indicam os grupos de símbolos, os demais são descritos como de aplicação porque somente fazem sentido dentro do sistema, não fazendo sentido ao usuário no seu dia-a-dia. Cabe salientar que, no processo de ensino da escrita de sinais, são feitas referências aos grupos de símbolos e, por isso, esses foram definidos como signos de domínio. Essas informações são descritas na Tabela 6, que representa a tabela de signos para o *SignSMS*.

Tabela 6 - tabela de signos do SignSMS

<b>Signo composto: sinal</b>		
<b>Signo</b>	<b>Definição</b>	<b>Tipo</b>
mão	Característica da mão (signo composto)	Aplicação
toque	Característica do toque (signo composto)	Aplicação
movimento	Característica do movimento (signo composto)	Aplicação
rosto	Característica do rosto (signo composto)	Aplicação

corpo	Característica do corpo (signo composto)	Aplicação
<b>Signo composto: mão</b>		
<b>Signo</b>	<b>Definição</b>	<b>Tipo</b>
Grupo	Grupo do símbolo da mão	Domínio
Posição	Posição X e Y na tela	Aplicação
Rotação	Ângulo	Aplicação
Espelhamento horizontal	Inverte horizontalmente (s/n)	Aplicação
Espelhamento vertical	Inverte verticalmente (s/n)	Aplicação
<b>Signo composto: toque</b>		
<b>Signo</b>	<b>Definição</b>	<b>Tipo</b>
Grupo	Grupo do símbolo do toque	Domínio
Posição	Posição X e Y na tela	Aplicação
Rotação	Ângulo	Aplicação
Espelhamento horizontal	Inverte horizontalmente (s/n)	Aplicação
Espelhamento vertical	Inverte verticalmente (s/n)	Aplicação
<b>Signo composto: movimento</b>		
<b>Signo</b>	<b>Definição</b>	<b>Tipo</b>
Grupo	Grupo do símbolo do movimento	Domínio
Posição	Posição X e Y na tela	Aplicação
Rotação	Ângulo	Aplicação
Espelhamento horizontal	Inverte horizontalmente (s/n)	Aplicação
Espelhamento vertical	Inverte verticalmente (s/n)	Aplicação
<b>Signo composto: rosto</b>		
<b>Signo</b>	<b>Definição</b>	<b>Tipo</b>
Grupo	Grupo do símbolo do rosto	Domínio
Posição	Posição X e Y na tela	Aplicação
Rotação	Ângulo	Aplicação
Espelhamento horizontal	Inverte horizontalmente (s/n)	Aplicação
Espelhamento vertical	Inverte verticalmente (s/n)	Aplicação
<b>Signo composto: corpo</b>		
<b>Signo</b>	<b>Definição</b>	<b>Tipo</b>
Grupo	Grupo do símbolo do corpo	Domínio
Posição	Posição X e Y na tela	Aplicação
Espelhamento horizontal	Inverte horizontalmente (s/n)	Aplicação
Espelhamento vertical	Inverte verticalmente (s/n)	Aplicação
E-mail	E-mail do destinatário	Domínio
Assunto	Assunto da mensagem	Domínio
Nome	Nome do contato	Domínio
Notificação	Notificação de mensagem	Aplicação

#### 4.2.4. Modelos de tarefas

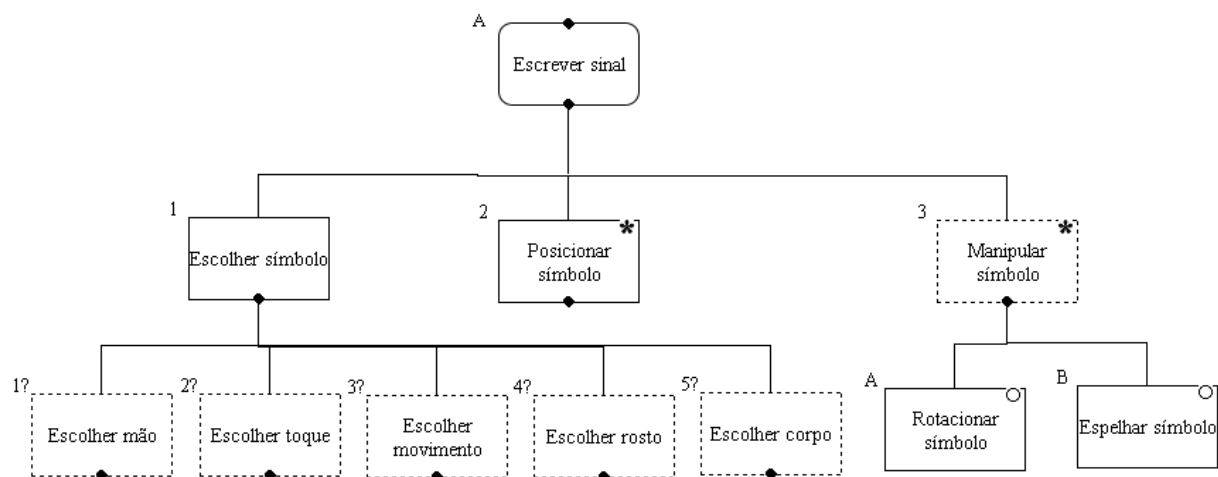
A partir dos cenários, são definidas as metas do sistema, ou seja, seus objetivos. Na descrição aqui apresentada, as metas foram nomeadas da mesma forma que os cenários,

para facilitar seu entendimento. Ainda, para cada meta, descrevemos um Modelo Hierárquico de Tarefas (MHT) que, por sua vez, descreve como as tarefas deverão ser realizadas pelo usuário. Nesse sentido, são apresentadas estruturas do tipo sequencial, independente de ordem e/ou alternada. As tarefas podem, ainda, ser opcionais, ubíquas ou iterativas e as mais simples, que remetem a descrição de ação do usuário sobre a interface do sistema, são denominadas de operadores.

A seguir, descreveremos os modelos de tarefas para cada meta utilizando o formalismo de Paula & Barbosa (2003) e Barbosa & da Silva (2010). Cabe salientar que os modelos de tarefas foram desenvolvidos utilizando a ferramenta *HTMaster* (Cavalleri, Ribeiro & Zanchin, 2012).

- Meta A: Escrever sinal (Figura 118)

Figura 118- Modelo de tarefa “Escrever sinal”



#### A..1 Escolher símbolo

SIGNOS: mão.grupo!, toque.grupo!, movimento.grupo!, rosto.grupo!,  
corpo.grupo!

##### A..1.1 Escolher mão

SIGNOS: mão.grupo?

##### A..1.2 Escolher toque

SIGNOS: toque.grupo?

##### A..1.3 Escolher movimento

SIGNOS: movimento.grupo?

##### A..1.4 Escolher rosto

SIGNOS: rosto.grupo?

## A..1.5 Escolher corpo

SIGNOS: corpo.grupo?

## A..2 Posicionar símbolo

SIGNOS: mão.grupo!, toque.grupo!, movimento.grupo!, rosto.grupo!, corpo.grupo!, mão.posição?, toque. posição?, movimento. posição?, rosto. posição?, corpo. posição?

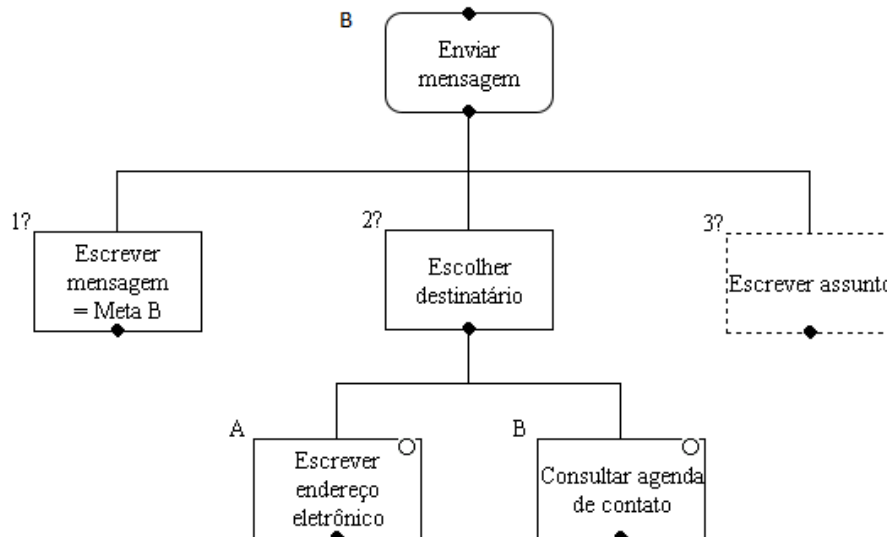
PREVENÇÃO ATIVA: os símbolos somente podem ser movimentados dentro da área de construção do sinal.

## A..3 Manipular símbolo

SIGNOS: mão.grupo!, toque.grupo!, movimento.grupo!, rosto.grupo!, corpo.grupo!, mão.rotação?, toque. rotação?, movimento. rotação?, rosto. rotação?, corpo. rotação?

- Meta B: Enviar mensagem (Figura 119)

Figura 119 - Modelo de Tarefa "Enviar Mensagem"



## B.2.A: Escrever endereço eletrônico

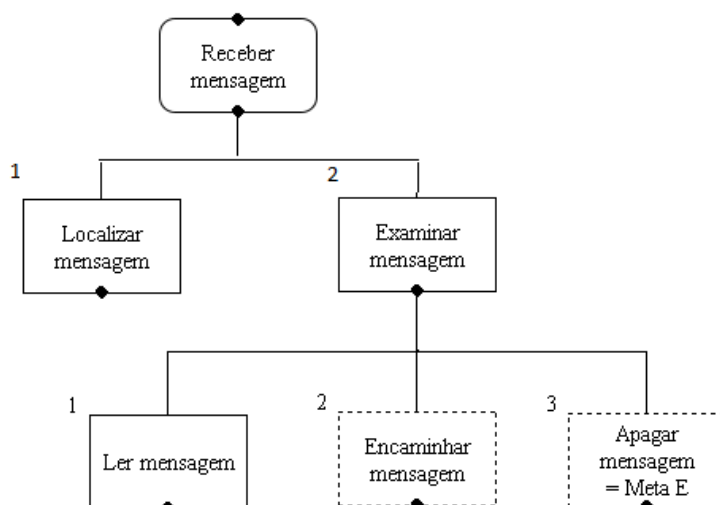
SIGNOS: *E-mail*?PREVENÇÃO ATIVA: É necessário que seja digitado um *e-mail* válido.

## B.2.B: Consultar agenda

SIGNOS: Nome?

- Meta C: Receber mensagem (Figura 120)

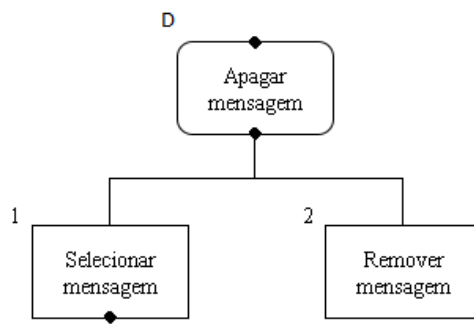
Figura 120 - Modelo de tarefa "Receber mensagem"



### C.2.2: Encaminhar Mensagem SIGNOS: Destinatario?

- Meta D: Apagar mensagem (Figura 121)

Figura 121 - Modelo de tarefa "Apagar mensagem"



D. 1: Selecionar mensagem

PREVENÇÃO ATIVA: É necessário que exista pelo menos uma mensagem.

D. 2: Remover mensagem

PREVENÇÃO ATIVA: É necessário que exista pelo menos uma mensagem.

## 4.3. Interface

Esta seção tem por finalidade identificar os principais elementos que foram desenvolvidos para a interface do aplicativo, mostrar as diferenças entre o que estava previsto e o resultado final da interface do *SignSMS*, bem como associar os padrões de interface relacionados aos componentes principais e as bibliotecas da plataforma *Android*

que foram utilizadas na implementação do protótipo. Os *mockups* gerados serviram de base para a construção da interface da aplicação. Como adotamos o método do projeto de design estrela, é justificável que, baseados em uma avaliação do resultado das tarefas obtidas, um novo design e modelagem fossem empregados. Esta etapa de avaliação foi retomada a cada passo, em um processo iterativo, até a finalização da aplicação atendendo ao disposto em Barbosa & da Silva, 2010.

Para facilitar, ilustraremos as diferenças entre as telas inicial e final justificando as alterações no design da interface das telas. Apresentaremos os elementos seguindo o fluxo de ações para o envio de uma mensagem, mostrando os componentes da tela e as funcionalidades associadas a estes componentes.

Para criar o *mockup* da interface do *SignSMS* foram utilizadas as ferramentas *Android Mock Up*<sup>26</sup>, que é uma ferramenta *on-line* gratuita para desenho de telas para aplicações web em *Android* e a ferramenta *Paint* 2007, disponível no sistema operacional *Windows* 7, para os ajustes necessários.

As figuras a seguir ilustram as diferenças entre o esboço e o resultado final de cada tela, onde o *mockup* da aplicação fica à esquerda e a tela oficial à direita. A Figura 122 ilustra a tela inicial do *SignSMS*, a qual oferece aos usuários um meio de acesso rápido às principais funcionalidades da aplicação, que são escrever ou ler mensagens. Os ícones utilizados para representar as opções de “Escrever Mensagem” e “Mensagens Recebidas”, foram elaborados de modo que pudessem ser associados a suas funcionalidades. O ícone que representa a função “Sobre” tem o mesmo padrão de diversos ícones da web relacionados a essa funcionalidade. O padrão de *design* adotado foi o *Dashboard* (seção 2.5).

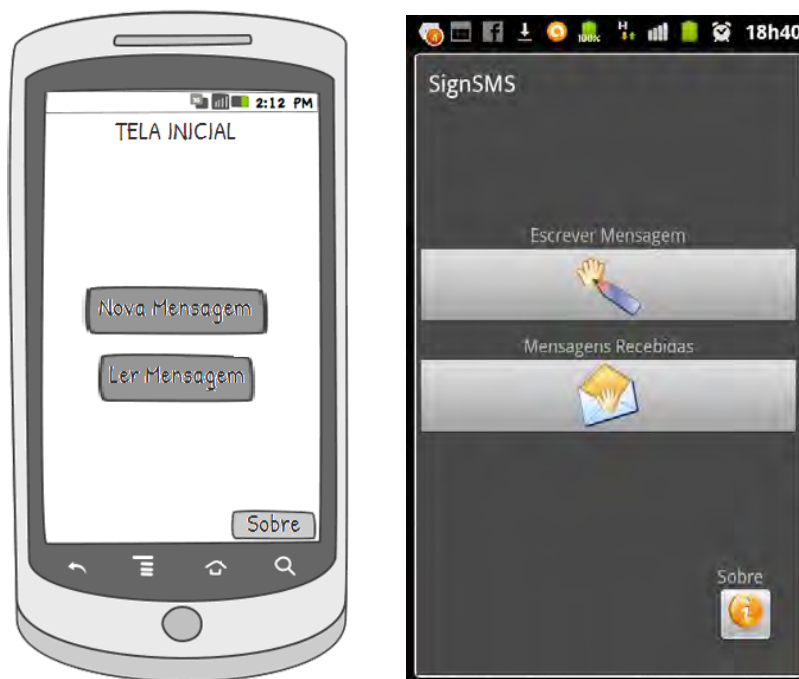
Em sua primeira versão, a tela inicial possui três botões para navegação. Os elementos que foram previstos foram mantidos nesta tela, contudo a descrição dos botões foi alterada de forma a facilitar o entendimento dos usuários sabedores da Língua Portuguesa em associar o desenho dos ícones às funcionalidades de cada botão (Figura 122):

---

<sup>26</sup> Android Mock Up. Disponível em <<http://yeblon.com/androidmockup/>>. Acesso em 23 jun. 2012.

- Nova Mensagem: descrição alterada para “Escrever Mensagem”. Ao clicar neste botão o usuário é direcionado para a tela de criação de mensagens.
- Ler Mensagem: descrição alterada para “Mensagens Recebidas”. Ao clicar neste botão o usuário é direcionado para a tela que possui as mensagens recebidas.
- Sobre: direciona o usuário para a tela com informações gerais do aplicativo.

Figura 122 – SignSMS - tela inicial: mockup x resultado



Ao ser escolhida a opção “Escrever mensagem”, o usuário é direcionado para a tela de escrita da mensagem em língua de sinais. Esta tela sofreu diversas alterações com relação ao que foi previsto, conforme Figura 123. Foi previsto que essa tela seria formada por dois elementos principais: uma área no topo da aplicação onde o usuário visualizaria o texto que estaria em construção, ou seja, os sinais já incluídos na mensagem, e a área para a construção de sinais. Contudo, esta tela de pré-visualização do texto foi retirada devido à complexidade, considerando tempo de implementação. O botão para abrir o menu principal dos símbolos ficou centralizado na área de construção do texto (*canvas*).



Figura 123 - SignSMS: tela para escrita da mensagem, mockup x resultado




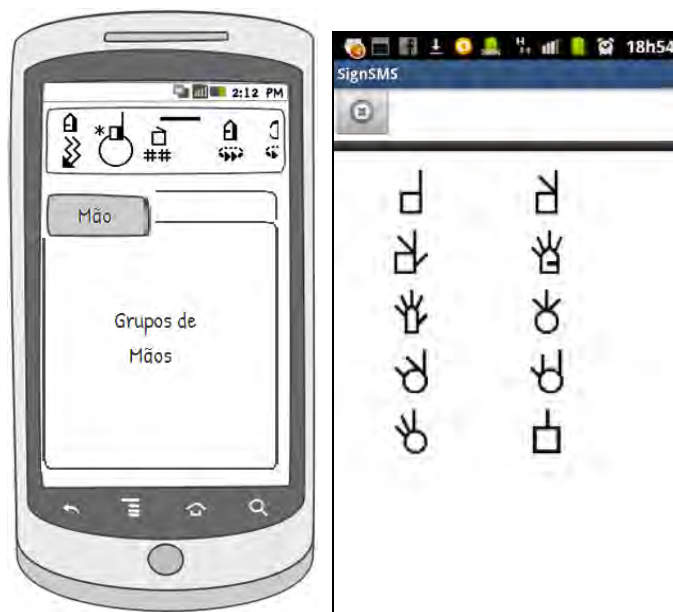
Para que o usuário comece a escrever o texto em *SignWriting*, primeiramente deve acessar o menu que contém as categorias de grupos de símbolos clicando no botão , que ativa o menu que contém os seis grupos principais de símbolos, de acordo com a estrutura do sistema *SignWriting* (Figura 124). Esse menu se desloca da direita para a esquerda para um melhor aproveitamento da área de escrita de sinais.


Figura 124 - SignSMS: menu categoria de símbolos



De acordo com o grupo selecionado, o usuário é direcionado para uma tela diferente: grupos de mãos, de toques, movimento, expressão facial, corpo ou pontuação. A Figura 125 ilustra as opções do grupo de mãos.

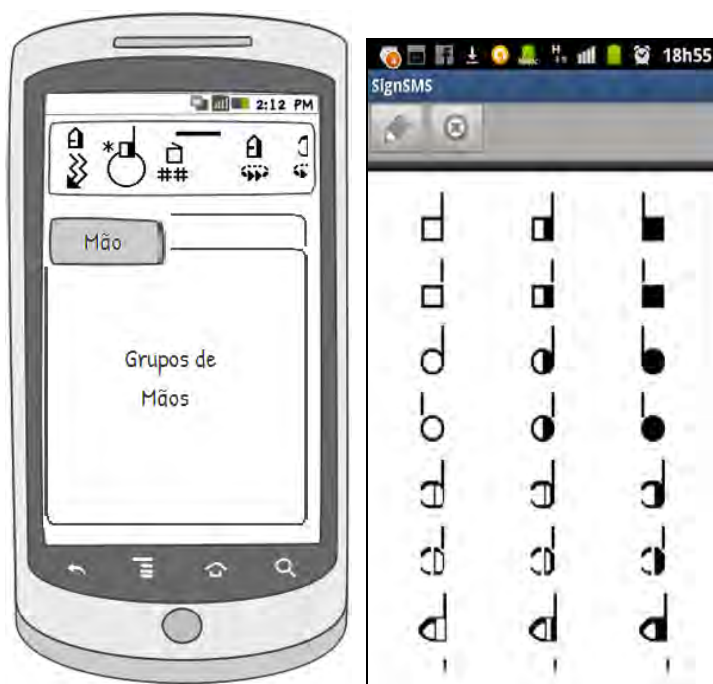
Figura 125 - SignSMS: opções do grupo de mãos



Esta atela apresenta um botão no topo, o botão  serve para fechar a tela atual e direcionar o usuário para a tela anterior. Mas se o usuário clicar sobre um símbolo, serão mostrados os símbolos que fazem parte do grupo selecionado. Supondo que o usuário tenha selecionado o botão que representa as mãos e desse tenha escolhido o grupo 1, serão mostradas as configurações do Grupo 1, conforme Figura 126.

Então, ao selecionar um dos grupos de símbolos, todos os símbolos referentes à categoria selecionada são apresentados em um novo menu, em uma nova janela. A distribuição dos símbolos e a navegação foram baseadas na funcionalidade do software *SW-Edit*, de modo que o usuário necessite navegar no máximo três níveis para encontrar um sinal desejado. Obviamente, essa distribuição também foi orientada pelos agrupamentos definidos no sistema *SignWriting* (Suton 2010 ).

Figura 126 - SignSMS: opções de configurações de mãos do Grupo 1





A tela representada na Figura 126 apresenta dois botões no topo, o botão  , já explicado, e o botão  , que direciona o usuário à área de edição de texto para que ele possa manipular o símbolo que foi inserido e compor o sinal desejado (Figura 127).

Figura 127 - SignSMS: símbolo inserido na área de construção de sinal



Ainda, como padrão de localização de inserção de um símbolo foi adotado o centro da tela de forma que minimizasse a necessidade de deslocamento pela tela. Ainda, quando o usuário seleciona um símbolo, esse fica apresentado como se estivesse dentro de uma caixa de seleção, conforme ilustrado na Figura 128. Tal recurso permite que o usuário saiba qual símbolo encontra-se selecionado. Ainda, pode-se selecionar mais de um símbolo e, nesse caso, as ações serão aplicadas a eles.
















Figura 128 - SignSMS: menu para manipulação de um símbolo selecionado



Para mover um símbolo na área de construção, o usuário deve clicar sobre o mesmo de forma que fique selecionado. A partir desse ponto, pode arrastá-lo na tela com o dedo para a posição desejada, atualmente temos uma limitação quando mais de um símbolos é selecionado, esta limitação é explicada na seção 5.1 que aborda as limitações e trabalhos futuros, o usuário também pode mover os símbolos utilizando as setas direcionais localizadas abaixo da área de edição dos símbolos, estas funcionam plenamente, independente da quantidade de símbolos selecionados. Cabe informar que em um dos primeiros esboços de funcionamento de manipulação dos símbolos estava sendo previsto um menu gráfico dinâmico, que apareceria enquanto o usuário permanece pressionando um símbolo na tela (*long click*), porém esta ação entrou em conflito com o movimento de arrastar os símbolos e foi retirada do escopo de implementação.

No que se refere ao menu de ações sobre os símbolos, que fica na parte inferior da tela do aplicativo, foram adotadas cores para diferenciar o contexto de aplicação das ações dos botões. Assim, os ícones de cor vermelha representam as ações sobre o símbolo e os ícones de cor laranja representam as ações sobre a ação do usuário. A Tabela 7 ilustra os ícones dos botões do menu de ações da tela de edição e suas respectivas funcionalidades.

Tabela 7 - SignSMS: Menu de ações

Botão	Função
	Move o símbolo selecionado para a direita.
	Move o símbolo selecionado para a esquerda.
	Move o símbolo selecionado para cima 10 pixels.
	Move o símbolo selecionado para baixo 10 pixels.
	Rotaciona o símbolo selecionado em 15° para a direita.
	Rotaciona o símbolo selecionado em 15° para esquerda.
	Espelha o símbolo selecionado horizontalmente.
	Espelha o símbolo selecionado verticalmente.
	Duplica o símbolo selecionado. O símbolo é criado no centro da área de edição do texto.
	Faz com que o símbolo selecionado seja enviado para frente dos demais independente da ordem em que tenham sido criados.
	Faz com que o símbolo selecionado seja enviado para trás dos outros independente da ordem em que tenham sido criados.
	Exclui o símbolo selecionado.
	Apaga todos os símbolos da página atual, limpando a área de edição de texto.
	Desfaz a última ação do usuário.
	Refaz a última ação desfeita pelo usuário.





Depois que o usuário escreve o texto desejado, para enviar a mensagem é necessário acessar o menu de opções através do botão menu do aparelho (Figura 129) e clicar no botão enviar, representado pelo ícone .

Figura 129 – SignSMS: menu de opções da tela de construção da mensagem



Ao clicar em enviar, o usuário é direcionado para a área onde serão informados os dados necessários para o envio da mensagem, que inclui assunto, que será um campo opcional, e o(s) destinatário(s) da mensagem, que é campo obrigatório. Esses campos são representados pelos  e  (Figura 130). O usuário pode incluir o e-mail do destinatário digitando-o ou selecionando-o via agenda de contatos do telefone. Nesse último caso, é utilizada a *API ContactsContract*<sup>27</sup>. Finalmente, para enviar a mensagem aos destinatários informados o usuário deve clicar no botão .

<sup>27</sup> *Androdi Developer. API ContactsContract*. Mais informações em



<http://developer.android.com/reference/android/provider/ContactsContract.html>

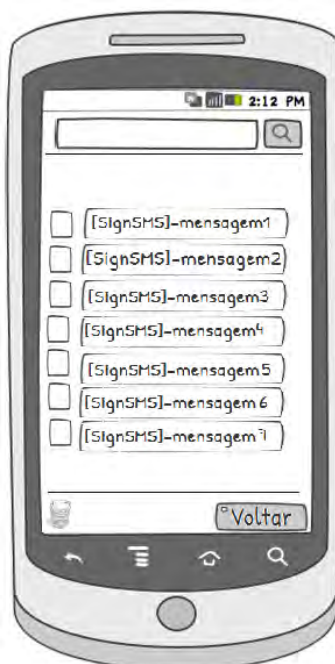
Figura 130 - SignSMS: tela para inserção dos dados de envio da mensagem



Encerra-se, assim, o processo de envio de uma mensagem utilizando o *SignSMS*.

No que se refere à leitura das mensagens recebidas, o usuário deve acessar a caixa de recebimento através do botão “Mensagens Recebidas” na tela principal da aplicação (Figura 122). Ao selecionar este botão, a tela contendo as mensagens recebidas é exibida conforme *mockup* ilustrado pela Figura 131, a tela final não foi desenvolvida até o momento.

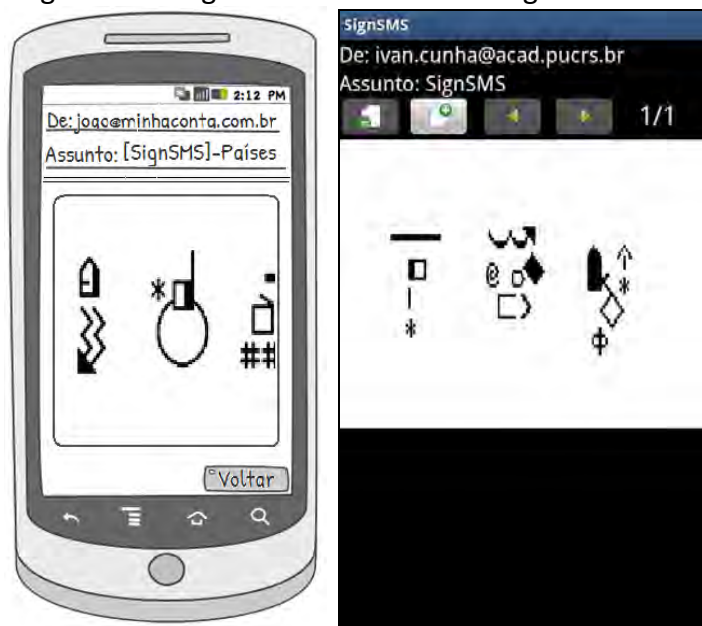
Figura 131 - SignSMS: tela de mensagens recebidas





Nesta tela representada na Figura 131, o usuário pode fazer a leitura com um clique sobre a mensagem desejada. Na área de leitura, a mensagem será visualizada utilizando a mesma navegação da tela de escrita da mensagem. No topo da tela são mostradas as informações do remetente da mensagem e o menu de navegação similar ao menu da tela de escrita da mensagem (Figura 132).

Figura 132 - SignSMS: leitura da mensagem recebida



Além da leitura, o usuário pode responder a mensagem recebida ou encaminhá-la a outros usuários. As opções de resposta e encaminhamento das mensagens podem ser acessadas através do botão menu do dispositivo (Figura 133).



Figura 133 - SignSMS: menu de opções da tela de leitura da mensagem



Na próxima seção faremos uma relação entre as funcionalidades e os recursos disponíveis na plataforma *Android* para o desenvolvimento do *SignSMS*.

#### 4.4. Funcionalidades x classes *Android*

A partir dos estudos e discussões já realizados, destacamos as características do *SignSMS* e já as relacionamos com classes públicas do *Android framework*<sup>28</sup>, utilizadas no desenvolvimento e gerenciamento das informações na aplicação aqui proposta.

- O *SignSMS* permite que os usuários interajam com a aplicação por meio da tela sensível ao toque, podendo o usuário utilizar o recurso de arrastar para compor a escrita de um sinal, de modo que o usuário possa construir os sinais desejados necessários à formação do texto. O *Android framework* define a classe *Touch*<sup>29</sup>, que permite que sejam programadas as ações de toque na tela.
- Considerando que a tela dos telefones é maior na altura do que na largura, depois de feitas avaliações sobre a usabilidade das telas e menus, a rotação da tela foi bloqueada devido à disposição dos botões de ação, e do tamanho do *canvas* (tela de edição).

<sup>28</sup> *Android framework* – Mais informações em <http://developer.android.com/>

<sup>29</sup> Classe *Touch* – Mais informações em <http://developer.android.com/resources/articles/touch-mode.html>

- Para o desenvolvimento do envio da mensagem, utilizamos a *API JavaMail*<sup>30</sup>, que possibilita o envio e recebimento de *E-mails*.
- Para podermos movimentar, rotacionar ou espelhar um sinal, usaremos uma classe do *Android framework* chamada *Canvas*<sup>31</sup>, que nos permite desenhar os símbolos e movimenta-los livremente na tela.

#### **4.5. Recursos de desenvolvimento**

Considerando o desenvolvimento de uma aplicação para *Web*, estão previstos os seguintes recursos de software:

- IDE Eclipse Java (Recomendada versão 3.7 "Indigo" ou mais recente)
- Java Platform (Java Development Kit, JDK) 7u3
- Android Development Toolkit (SDK, versão mínima 14, recomendada 16)
- Android SDK (versão mínima 2.2)
- ADT Plugin for Eclipse

No que se refere aos recursos de hardware, foram utilizados no desenvolvimento e testes:

- Um computador com Microsoft Windows 7.
- Um celular Samsung Galaxy modelo W com uma versão do *Android* 2.3.3.
- Um celular Samsung Mini-Galaxy modelo GT-S5570B com versão do *Android* 2.2.1.

---

<sup>30</sup> *API JavaMail* – Mais informações em <http://www.oracle.com/technetwork/java/javamail/index.html>

<sup>31</sup> Classe *Canvas* – Mais informações em <http://developer.android.com/reference/android/graphics/Canvas.html>

## 5. CONCLUSÃO

Como vimos, este trabalho é resultado de um estudo sobre diferentes ferramentas, sistemas de comunicação no sistema de escrita de sinais, editores gráficos, modelos de interação humano-computador, padrões e princípios de design para implementação de aplicativos para dispositivos móveis e aplicações web.

Esta pesquisa tornou possível consolidar a importância da utilização de mecanismos de comunicação nos quais as pessoas possam se comunicar na sua língua materna (no caso de pessoas surdas) e possam (pessoas surdas e ouvintes) melhor aprender esse processo de escrita. Também mostrou que a plataforma *Android* vem em constante crescimento no mercado consumidor tendo sua abrangência e importância ressaltada em diversos fóruns e *sites* tanto por ser uma plataforma gratuita quanto pelo crescente número de aplicações que são colocadas no mercado de aplicações disponibilizado pela *Google* conhecido como *Google Play*<sup>32</sup> para os diferentes nichos de necessidade.

Esse estudo confirmou a opção do grupo pelo desenvolvimento de um protótipo de ferramenta para comunicação assíncrona por meio de telefones móveis, a partir do sistema de escrita de sinais *SignWriting* na plataforma *Android*. Também é importante ressaltar que pesquisas realizadas na Internet não indicaram a existência de uma ferramenta que atenda a esse propósito. Portanto, tem seu caráter de inovação ampliado, e de desafios também de forma que houve muitas lições aprendidas durante o processo de TC I e TC II.

Para responder a questão de pesquisa do trabalho, que está relacionada às características que deve ter uma aplicação para dispositivos móveis com suporte à construção de mensagens utilizando o sistema de escrita de sinais *SignWriting*, considerando seu uso tanto por pessoas surdas quanto pessoas ouvintes interessas em aprender e/ou desenvolver seus conhecimentos sobre esse processo de escrita, foram analisadas diferentes ferramentas que utilizam tanto o *SignWriting* quanto outros métodos de escritas de sinais para ambientes *web* e/ou para *Desktop*, aplicações com recursos de edição de imagens, dicionários e tradutores mensagens escritas utilizando algum sistema de escritas de sinais. Embora não seja possível testar ferramentas similares à proposta neste

---

<sup>32</sup> Google Play. Mais detalhes em: <https://play.google.com/store>

TCC, as análises indicaram características importantes de interface e de interação do usuário com o sistema, bem como trouxeram informações relevantes sobre tecnologias de desenvolvimento, limitações e o potencial de uso.

Realizada a análise das ferramentas similares ao proposto nesse TCC foi observado que algumas funções que estavam presentes nas ferramentas atuais de mercado eram pertinentes para o uso na ferramenta proposta, como por exemplo, envio e recebimento de mensagens utilizando apenas os símbolos do *SignWriting*. Outro item que foi observado nas ferramentas é a dificuldade de se navegar nas aplicações para selecionar um símbolo desejado devido à quantidade de elementos que compõem a biblioteca de símbolos do *SignWriting*. Tal ponto foi levado em consideração para a criação do *SignSMS*, como uma preocupação com a usabilidade da ferramenta.

Então, considerando a questão de pesquisa deste trabalho, podemos assim nos posicionar:

*Que características deve ter uma aplicação computacional interativa para permitir uma comunicação a distância entre pessoas surdas, no contexto de uma educação bilíngue? E entre pessoas surdas e pessoas ouvintes, considerando o contexto de mobilidade tecnológica? Deve permitir a interação utilizando a língua materna de seu público-alvo, que primeiramente são pessoas surdas usuárias das línguas de sinais. Portanto, deve permitir a escrita de mensagens a partir da língua de sinais. Ainda, para que possa ser utilizada por pessoas surdas de diferentes países, a aplicação deve fazer uso de dicionários de idiomas de LS ou permitir a escrita livre. Para o *SignSMS* a escolha foi pela escrita livre porque desejamos que o aplicativo também possa ser utilizado por aqueles que estão em processo de alfabetização da escrita da LS. Ainda, no que se refere à escrita, a opção foi pelo sistema *SignWriting* porque o mesmo vem sendo utilizado em escolas e universidades brasileiras. Inclusive o curso de Letras/LIBRAS possui disciplinas específicas que tratam da escrita de sinais por meio do *SignWriting*. Ainda, devido ao *SignWriting* ser um sistema universal de escrita de sinais, ele já vem sendo utilizado em diferentes países, conforme indica *SignWriting* (2012).*

No que se refere aos objetivos, consideramos que a realização deste Trabalho de Conclusão possibilitou construir mais conhecimento sobre temas relacionados à Interação humano-computador, relacionando-os à modelagem e ao desenvolvimento de sistemas interativos, e a padrões de design, que até então conhecíamos de forma não tão aprofundada. Também aprendemos sobre as línguas de sinais, que não são universais, e que se configuram como língua materna das pessoas surdas. A descoberta de que podem ser representadas de forma escrita permitiu verificar novas possibilidades de desenvolvimento de aplicações de sistemas e ao mesmo tempo o desafio de centralizarmos para as questões pertinentes ao *SignSMS*. O estudo do *Signwriting* e a utilização do mesmo no *SignSMS* foi um desafio devido à quantidade de símbolos e às especificidades de cada grupo. Por fim, é importante ressaltar que o desenvolvimento de aplicações para tecnologia móvel já era um desafio ao grupo, que não tinha experiência nessa área. Assim, buscamos cursos de extensão e também informações de colegas que pudessem discutir conosco problemas que já sabíamos que iríamos enfrentar. Alguns, ainda permaneceram em aberto, outros foram resolvidos, mas quem sabe podem ser mais bem otimizados em uma próxima versão do *SignSMS*.

Concluimos, por fim, que o *SignSMS* não deve se encerrar neste trabalho. Deve ter continuidade e, por isso, na seção que segue apresentamos algumas limitações e sugestões de trabalhos futuros.

### **5.1. Limitações e trabalhos futuros**

Esta seção apresenta as funcionalidades que foram previstas no TC I e não foram desenvolvidas devido à relação dificuldades do grupo com relação à tecnologia x tempo disponível. Também são incluídas funcionalidades que o grupo considera importantes para uma próxima versão do *SignSMS*.

No que se refere à proposta de TC e a entrega do TC II, cita-se:

- Foi previsto o uso de um banco de sinais, que seria construído pelo usuário na medida em que o mesmo construísse sinais utilizando o *SignSMS*. Tal recurso permitiria maior produtividade do usuário porque os sinais poderiam ser reaproveitados.

- Foi previsto o envio de mensagens no formato português escrito, mas, no decorrer do TC II o grupo optou por direcionar o *SignSMS* para o envio e leitura de mensagens baseadas em escrita de sinais, já que há outras ferramentas que suportam o envio e leitura de mensagens escritas em língua oral.
- Foi previsto o uso do *long click* para diferenciar ações de escolher e de ativação de menus, por exemplo, para que submenus pudessem ser visualizados. Esta ação foi retirada em tempo de implementação porque estava entrando em conflito com o movimento de arrastar os símbolos. Desta forma, foi priorizado o desenvolvimento de outros recursos.

Como trabalhos futuros, indicamos:

- Melhoria na qualidade dos símbolos, pois essa se difere entre aplicações com resoluções diferentes. Assim, os símbolos devem considerar a densidade suportada por cada dispositivo, conforme densidades recomendadas no site de desenvolvimento *Android Developers*. Importante ressaltar que esse recurso foi adotado nas imagens dos menus, incluindo o dos símbolos, mas não aplicado à área de edição dos sinais.
- Botões de ação na tela de edição poderiam ficar ocultos quando não estivessem em uso. Mas essa questão poderia ser confirmada durante os testes com os usuários finais.
- A interação com o movimento de arrastar os símbolos perde qualidade quando há mais de um símbolo na área de edição, o que traz certa instabilidade e insegurança ao usuário, que por vezes consegue selecionar mais de um símbolo e arrastá-los e em outras, não. Essa questão também deveria ser observada junto aos usuários finais para verificar sua relevância.
- A figura de seleção deve ser redimensionada de acordo com a imagem selecionada, pois, atualmente, quando o símbolo é menor que a imagem de seleção, a seleção não é escalada para se ajustar ao símbolo, somente quando o símbolo possui o tamanho maior ou igual à imagem de seleção.

Essa pendência deveria ser solucionada antes da realização dos testes com os usuários.

- Havendo o uso de um banco de sinais, sugere-se o uso do recurso de predição de sinais, descrito no *Pacote Sign*, para que o usuário possa, durante a composição de um sinal, verificar os sinais já criados com aqueles símbolos. Tal recurso permitiria maior produtividade ao processo de escrita das mensagens.
- Extender a compatibilidade do *SignSMS* para outras plataformas como, por exemplo, *iPhone* e *Windows Phone*, permitindo uma maior interação entre as pessoas, independente do tipo de dispositivos utilizado.
- Validação do protótipo junto a usuários reais com a finalidade de avaliar sua interface, funcionalidades e interação.

## REFERÊNCIAS

ACESSIBILIDADE BRASIL. **O que é acessibilidade**. Disponível em:

<http://www.acessobrasil.org.br/index.php?itemid=45>. Acesso em: 09 abr. 2012 (a).

ACESSIBILIDADE BRASIL. **Recomendações W3C**. Disponível em:

<http://www.acessobrasil.org.br/index.php?itemid=41>. Acesso em: 03 abr. 2012 (b).

ANDREW, Rachel. *WEB STANDARDS PROJECT (WaSP)*. Disponível em:

<<http://www.Webstandards.org>>. Acesso em: 09 abr. 2012.

**Android Developers**. Design Principles. Disponível em:

<<http://developer.android.com/design/get-started/principles.html>>. Acesso em: 11 abr. 2012 (a).

Android Developers. **What is Android?** Disponível em:

<<http://developer.android.com/guide/basics/what-is-android.html>>. Acesso em: 11 abr. 2012 (b).

Android Patterns. **Quick Actions**. (2011). Disponível em:

<[http://www.androidpatterns.com/uap\\_pattern/quick-actions](http://www.androidpatterns.com/uap_pattern/quick-actions)>. Acesso em: 23 jun. 2012.

ASL AMERICAN SIGN LANGUAGE. Disponível em: <[http://pt.appbrain.com/app/asl-](http://pt.appbrain.com/app/asl-american-sign-language/com.teachersparadise.aslamericansignlanguage)

[american-sign-language/com.teachersparadise.aslamericansignlanguage](http://pt.appbrain.com/app/asl-american-sign-language/com.teachersparadise.aslamericansignlanguage)>. Acesso em: 08 abr. 2012.

BABY SIGN LANGUAGE DICT. Disponível em: < [http://pt.appbrain.com/app/baby-signing-](http://pt.appbrain.com/app/baby-signing-lite-msh/com.rbi.msh_dictionary_lite)

[lite-msh/com.rbi.msh\\_dictionary\\_lite](http://pt.appbrain.com/app/baby-signing-lite-msh/com.rbi.msh_dictionary_lite)>. Acesso em: 07 abr. 2012

BARBOSA, Simone Diniz Junqueira; da SILVA, Bruno Santana. **Interação Humano-Computador**. Editora Campus-Elsevier, 2010.

BOHZ, Rafaela. **Possibilidades de uso de software educacional no processo de ensino e**

**aprendizagem do aluno surdo**. 2011. Monografia (Especialização em Informática na Educação – 2ª Edição) – Faculdade de Informática e Faculdade de Educação, PUCRS, Porto Alegre, 2011.

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. Disponível em:

<[http://www.planalto.gov.br/ccivil\\_03/\\_Ato2004-2006/2005/Decreto/D5626.htm#art1](http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2005/Decreto/D5626.htm#art1)>. Acesso em: 04 mar. 2012

BRASIL. Lei nº 10.436, de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais - Libras e dá outras providências. Disponível em:

<[http://www.planalto.gov.br/ccivil\\_03/Leis/2002/L10436.htm](http://www.planalto.gov.br/ccivil_03/Leis/2002/L10436.htm)>. Acesso em: 11 abr. 2011.



C.E.S.A.R. **Android: Padrões de Projeto para Interface Usuário. 2011.** Disponível em <<http://techne.cesar.org.br/android-padroes-de-projeto-para-interface-usuario/>>. Acesso em: 20 jun. 2012.

CAMPOS, Márcia de Borba. **Ambiente Telemático de Interação e Comunicação para Suporte à Educação Bilíngüe de Surdos.** 2001. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação, UFRGS, Porto Alegre, 2001.

CAMPOS, Marcia de Borba. **O uso das TIC na Educação de Surdos.** 2007. Disponível em: <[http://www.inf.pucrs.br/~marciabc/20072/libras/inf\\_surdo\\_libras.pdf](http://www.inf.pucrs.br/~marciabc/20072/libras/inf_surdo_libras.pdf)>. Acesso em: 15 abr. 2012.

CAMPOS, Márcia de Borba; MAIOCCHI, Daniela; BORTOLIN, Michael. SIGNHQ: sistema de autoria para criação e leitura de Histórias em Quadrinhos com suporte à Língua de Sinais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 15., 2004, Manaus. **Anais...** Manaus: SBC/SBIE, 2004. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/371/357>>. Acesso em: 15 abr. 2012.

CAMPOS, Márcia de Borba; OLIVEIRA, Dalvares; SANTOS, Guilherme. SignWebEdit: Uma oportunidade para a criação coletiva de textos escritos em língua de sinais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 17., 2006, Brasília. **Anais...** Brasília: SBC/SBIE, 2006. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbie/2006/030.pdf>>. Acesso em: 14 abr. 2012.

CAMPOS, Márcia de Borba; SILVEIRA, Milene Selbach. Promoção da Cidadania da Comunidade Surda: o uso das TICs na apropriação de sua língua materna. In: Seminário Integrado de Software e Hardware, 34., 2007, Rio de Janeiro. **Anais...** Rio de Janeiro: SBC/SEMISH, 2007. Disponível em: <<http://www.de9.ime.eb.br/~sousamaf/cd/pdf/arq0234.pdf>>. Acesso em: 10 mar. 2012.

CGI.br. Comitê Gestor da Internet no Brasil. **Dimensões e características da Web brasileira: um estudo do .gov.br.** CGI.br: 2010. Disponível em: <<http://www.cgi.br/publicacoes/pesquisas/govbr/cgibr-nicbr-censoWeb-govbr-2010.pdf>>. Acesso em: 09 abr. 2012.

COSTA, Antônio Carlos da Rocha; TORCHELSEN, Rafael Piccin. **SW-Edit.** Disponível em: <[http://rocha.c3.furg.br/index.php?Itemid=513&option=download\\_categoria&task=detalhe&id\\_site\\_componente=707&id=31](http://rocha.c3.furg.br/index.php?Itemid=513&option=download_categoria&task=detalhe&id_site_componente=707&id=31)>. Acesso em: 16 abr. 2012.

JEMNI, Mohamed. **SMS to MMS sign language for mobile phones.** 2010. Disponível em: <<http://www.youtube.com/watch?v=9mDkNppWb1M>>. Acesso em: 20 mar. 2012.

KARNOPP, Lodenir Becker. **Aquisição do parâmetro Configuração de Mão na língua Brasileira dos Sinais (LIBRAS):** Estudo sobre quatro crianças surdas, filhas de pais surdos. Porto Alegre: PUCRS, 1994. 154 f. Dissertação (Mestrado em Letras) - Instituto de Letras e Artes, Pontifícia Universidade Católica do Rio Grande do Sul, 1994.

LAHTI'S, Bill. **How To Build A Dashboard User Interface In Android** (2011). Disponível em: <<http://blahti.wordpress.com/2011/03/14/build-dashboard-ui-for-android/>>. Acesso em: 12 jun. 2012.

LECHETA, Ricardo R. **Google Android**: aprenda a criar aplicações para dispositivos móveis com Android SDK. São Paulo: Novatec Editora, 2010.

MOBILE WEB INICIATIVE – W3C. **Boas Práticas em Web móvel**. Disponível em: <<http://www.w3c.br/divulgacao/pdf/guia-boas-praticas.pdf>>. Acesso em: 04 mar. 2012.

MOURA, Ana Lúcia Damasceno. **Tecnologias Móveis e Vida Pessoal**. São Paulo, 2008.

NIELSEN, J. **Ten usability heuristics**. In: NIELSEN, J.; MACK, R. (eds) Usability inspection methods. New York, John Wiley & Sons, 1994. Disponível em <[http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)>. Acesso em 29 mai. 2012.

PAULA, Maíra Greco de; BARBOSA, Simone Diniz Junqueira. **MOLIC** (apêndice). Rio de Janeiro, 2003. Disponível em: <[http://www.inf.pucrs.br/~milene/materiais\\_IHC/MOLIC.pdf](http://www.inf.pucrs.br/~milene/materiais_IHC/MOLIC.pdf)>. Acesso em: 14 ago. 2011.

PONTES, Adéle Malta; ORTH, Afonso Inácio. Uma Proposta de Interface de Software Orientada à Linguagem de Sinais. In: Workshop sobre fatores humanos em sistemas computacionais, 2., 1999, Campinas. **Anais...** Campinas: SBC/IHC, 1999. Disponível em: <<http://www.unicamp.br/~ihc99/Ihc99/AtasIHC99/art31.pdf>>. Acesso em: 14 abr. 2012.

PRATES, Raquel Oliveira; SOUZA, Clarice S.; BARBOSA, Simone D. J. . **A Method for Evaluating the Communicability of User Interfaces**. New York, NY: ACM Press, 2000.

PREECE, Jennifer; ROGERS, Yvonne; SHARP, Helen. **Design de interação: além da interação homem-computador**. Porto Alegre: Bookman, 2005.

QUADROS, Ronice Müller de. **Um capítulo da história do SignWriting**. Disponível em: <<http://www.SignWriting.org/archive/docs1/sw0065-BR-Historia-SW.pdf>>. Acesso em: 24 mar. 2012.

ROSA, Emiliana Faria. **Olhares sobre si: a busca pelo fortalecimento das identidades surdas**. Salvador: UFBA, 2009. 146 f. Dissertação (Mestrado em Educação) – Faculdade de Educação, Universidade Federal da Bahia, 2009.

SANTOS, Beatriz Souza. **Introdução à Interação e Usabilidade**. 2008. Disponível em: <<http://www.ieeta.pt/~bss/aulas/Introdu%C3%A7%C3%A3o-IHC-08-h.pdf>> Acessado em: 06 de abr. 2012.

SBC, Sociedade Brasileira de Computação. **Grandes Desafios da Pesquisa em Computação no Brasil – 2006 – 2016**. Maio, 2006. Disponível em: <[http://www.sbc.org.br/index.php?option=com\\_jdownloads&Itemid=195&task=finish&cid=11&catid=50](http://www.sbc.org.br/index.php?option=com_jdownloads&Itemid=195&task=finish&cid=11&catid=50)>. Acesso em: 28 mar. 2012.

- SIGN 4 ME: Disponível em: <<http://itunes.apple.com/us/app/sign-4-me-for-ipad-signed/id383462870?mt=8>>. Acesso em: 10 abr. 2012.
- SIGNING SAVVY. Disponível em: <<http://pt.appbrain.com/app/signing-savvy-member-app/com.signingsavvy.mobile>>. Acesso em: 20 mar. 2012
- SIGNWRITING*. Disponível em: <<http://www.SignWriting.org/>>. Acesso em: 25 mar. 2012.
- SILVEIRA, Milene Selbach; CAMPOS, Márcia de Borba. **Design de interface**. Slides. 2011.
- SKYSIGNS ASL DICTIONARY. Disponível em: <http://pt.appbrain.com/app/skysigns-asl-dictionary/us.skyb.skysigns>>. Acesso em: 07 abr. 2012.
- SOUZA, PINTO. (2003) XIV Simpósio Brasileiro de Informática na Educação - NCE - IM/UFRJ 2003 Disponível em <<http://www.nce.ufrj.br/sbie2003/publicacoes/paper42.pdf>>. Acesso em 30 mai. 2012.
- SOUZA, Vinícius Casta de. **Sign WebMessage: um ambiente para comunicação via Web baseado na escrita da Língua Brasileira de Sinais**. Trabalho de conclusão. Graduação em Análise de Sistemas. Universidade do Vale do Rio dos Sinos - Unisinos. São Leopoldo, 2002. Disponível em: <[http://professor.unisinos.br/wp/crespo/files/2011/07/tc\\_vinicius\\_souza.pdf](http://professor.unisinos.br/wp/crespo/files/2011/07/tc_vinicius_souza.pdf)>. Acesso em: 20 jun. 2012.
- SOUZA, Vinícius Costa de; PINTO, Sérgio crespo C. da Silva. Sign WebMessage: uma ferramenta para comunicação via web através da Língua Brasileira de Sinais – Libras. In: Simpósio Brasileiro de Informática na Educação, 13., 2003. **Anais...** SBIE, 2003. Rio de Janeiro, 2003. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/270/256>>. Acesso em: 16 abr. 2012.
- SOUZA, Vinícius Costa de; PINTO, Sérgio Crespo Coelho da Silva. O Aprimoramento do Sign WebMessage como Base para o Desenvolvimento da SWService: uma Biblioteca para a Escrita da Libras na Internet baseada em Web Services. In: Simpósio Brasileiro de Informática na Educação, 14.,2004. **Anais...** SBIE, 2004. Disponível em <<http://www.br-ie.org/pub/index.php/sbie/article/view/326>>. Acesso em 29 mai. 2012.
- STUMPF, Marianne Rossi. **A Estrutura do Sistema SignWriting**. Disponível em <<http://www.SignWriting.org/archive/docs6/sw0569-BR-2008-Stumpf-ELSIII.pdf>>. Acesso em: 23 mar. 2012.
- STUMPF, Marianne Rossi. **Aprendizagem de escrita de Língua de Sinais pelo Sistema SignWriting: Línguas de Sinais no papel e no Computador**. 2005. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação, UFRGS, Porto Alegre, 2005. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/5429/000515254.pdf?sequence=1>>. Acesso em: 25 mar. 2012.

STUMPF, Marianne Rossi. **Lições sobre *SignWriting***. Disponível em: <<http://www.SignWriting.org/archive/docs5/sw0472-BR-Licoes-SignWriting.pdf>>. Acesso em: 11 abr. 2012.

SUTTON, Valerie. **Manual do *SignWriting* em Português**. Disponível em: <<http://escritadesinais.wordpress.com/2010/08/24/manual-de-SignWriting-em-portugues/>>. Acesso em: 25 mar. 2012.

TECMUNDO. **Android conquista quase 50% do mercado de smartphones**. Disponível em: <<http://www.tecmundo.com.br/android/12071-android-conquista-quase-50-do-mercado-de-smartphones.htm>>. Acesso em 14 abr. 2012.

VILCHEZ, Angel. Android: Características y Aplicaciones. Disponível em: <<http://www.configurarequipos.com/doc1107.html>>. Acesso em: 16 abr. 2012.

W3C MOBILE. **Guia de referência - Boas práticas em Web móvel**. Disponível em <<http://www.w3c.br/divulgacao/guia-boas-praticas2008/>>. Acesso em: 04 mar. 2012.

W3C PADRÕES. **Padrões**. Disponível em: <<http://www.w3c.br/Padroes>>. Acesso em: 04 mar. 2012.