

TURNO:	NOTURNO	VERSÃO:	1	ANO / SEMESTRE:	2014.2	Nº	
---------------	----------------	----------------	---	------------------------	--------	----	--

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO — BACHARELADO
COORDENAÇÃO DE TRABALHO DE CONCLUSÃO DE CURSO

PROPOSTA PARA O TRABALHO DE CONCLUSÃO DE CURSO

TÍTULO: SIGNA: UMA APLICAÇÃO PARA AUXÍLIO NO ENSINO-APRENDIZAGEM DE LIBRAS

ÁREA: Desenvolvimento de sistemas

Palavras-chave: Libras. Leap Motion. Aprendizado de máquina. Máquina de vetores de suporte. Redes Neurais.

1 IDENTIFICAÇÃO

1.1 ALUNO

Nome: Júlio César Batista		Código/matricula: 167432-3	
Endereço residencial:			
Rua: Alice Ferreira dos Santos		nº: 260	Complemento:
Bairro: Badenfurt	CEP: 89070-600	Cidade: Blumenau	UF: SC
Telefone fixo: (47) 3334.0718		Celular: (47) 8435.4003	
Endereço comercial:			
Empresa: Ellevo Soluções em Tecnologia da Informação			
Rua: Ângelo Dias		nº: 220	Bairro: Centro
CEP: 89010-020	Cidade: Blumenau	UF: SC	Telefone: (47) 2111.5400
E-Mail FURB:		E-Mail alternativo: jcb.1611@gmail.com	

1.2 ORIENTADOR

Nome: Aurélio Faustino Hoppe	
E-Mail FURB:	E-Mail alternativo: aurelio.hoppe@gmail.com

2 DECLARAÇÕES

2.1 DECLARAÇÃO DO ALUNO

Declaro que estou ciente do Regulamento do Trabalho de Conclusão de Curso de Ciência da Computação e que a proposta em anexo, a qual concordo, foi por mim rubricada em todas as páginas. Ainda me comprometo pela obtenção de quaisquer recursos necessários para o desenvolvimento do trabalho, caso esses recursos não sejam disponibilizados pela Universidade Regional de Blumenau (FURB).

Assinatura: _____ Local/data: _____

2.2 DECLARAÇÃO DO ORIENTADOR

Declaro que estou ciente do Regulamento do Trabalho de Conclusão do Curso de Ciência da Computação e que a proposta em anexo, a qual concordo, foi por mim rubricada em todas as páginas. Ainda me comprometo a orientar o aluno da melhor forma possível de acordo com o plano de trabalho explícito nessa proposta.

Assinatura: _____ Local/data: _____

3 AVALIAÇÃO DA PROPOSTA

3.1 AVALIAÇÃO DO(A) ORIENTADOR(A)

Acadêmico(a): Júlio César Batista

Orientador(a): Aurélio Faustino Hoppe

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

Assinatura do(a) avaliador(a): _____

Local/data: _____

CONSIDERAÇÕES DO(A) ORIENTADOR(A):

Caso o(a) orientador(a) tenha assinalado em sua avaliação algum item como “atende parcialmente”, devem ser relatos os problemas/melhorias a serem efetuadas.

Na segunda versão, caso as alterações sugeridas pelos avaliadores não sejam efetuadas, deve-se incluir uma justificativa.

[illegible]

Assinatura do(a) avaliador(a):

Local/data:

3.2 AVALIAÇÃO/HOMOLOGAÇÃO DO COORDENADOR DE TCC

Acadêmico(a): Júlio César Batista

Avaliador(a): Maurício Capobianco Lopes

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

OBSERVAÇÕES:

Assinatura do(a) avaliador(a):

Local/data:

3.3 AVALIAÇÃO DO PROFESSOR DA DISCIPLINA DE TCCI

Acadêmico(a): Júlio César Batista

Avaliador(a): Roberto Heinzle

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
	ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?		
8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?				
9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?				
9.2. As informações retiradas de outros autores estão devidamente citadas?				
10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?				
10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?				
PONTUALIDADE NA ENTREGA			atraso de _____ dias	
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

OBSERVAÇÕES:

Assinatura do(a) avaliador(a):

Local/data:

3.4 AVALIAÇÃO DO(A) PROFESSOR(A) ESPECIALISTA NA ÁREA

Acadêmico(a): Júlio César Batista

Avaliador(a): _____

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

OBSERVAÇÕES:

Assinatura do(a) avaliador(a): _____

Local/data: _____

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**SIGNA: UMA APLICAÇÃO PARA AUXÍLIO NO ENSINO-
APRENDIZAGEM DE LIBRAS**

JÚLIO CÉSAR BATISTA

BLUMENAU
2014

JÚLIO CÉSAR BATISTA

**SIGNA: UMA APLICAÇÃO PARA AUXÍLIO NO ENSINO-
APRENDIZAGEM DE LIBRAS**

Proposta de Trabalho de Conclusão de Curso
submetida à Universidade Regional de
Blumenau para a obtenção dos créditos na
disciplina Trabalho de Conclusão de Curso I
do curso de Ciência da Computação —
Bacharelado.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU
2014**

1 INTRODUÇÃO

A inteligência artificial é um assunto estudado há décadas e desde o começo as pessoas se perguntam se as máquinas são realmente capazes de pensar (SMITH et al., 2006, p. 4, tradução nossa). Segundo Smith et al. (2006, p. 4, tradução nossa) "Ninguém pode refutar a habilidade de um computador processar lógica. Mas para muitos é incerto se uma máquina pode pensar". Durante a década de 1970 as atividades comerciais e científicas com inteligência artificial decaíram e este período foi denominado "inverno da IA" e um dos termos que nasceram neste período foi o "aprendizado de máquina" (SMITH et al., 2006, p. 17-18, tradução nossa).

Atualmente, o aprendizado de máquina possui três principais áreas, sendo elas: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (RUSSEL; NORVIG, 2003, p. 650, tradução nossa). O aprendizado supervisionado é composto por técnicas para realizar a classificação ou análise de regressão de padrões através de exemplos de entrada e saída de uma função, estas técnicas podem ser utilizadas para reconhecimento facial (classificação) ou até mesmo análise de valores de imóveis (análise de regressão). As técnicas de aprendizado não supervisionado são utilizadas para aprender e agrupar padrões em entradas de funções, um uso destas técnicas seria o agrupamento de artigos de jornais por categorias. O aprendizado por reforço é composto pelas técnicas mais genéricas, onde ao invés de ser dito o que fazer, o algoritmo aprende através de recompensas (RUSSEL; NORVIG, 2003, p. 650, tradução nossa), estas técnicas podem ser utilizadas na navegação de robôs, onde o robô escolhe o caminho e depois é informado se a escolha foi boa ou não.

O aprendizado de máquina supervisionado tem se mostrado interessante para a solução de vários problemas cotidianos tais como: classificação de mensagens, avaliação de preços da bolsa de valores e mineração de dados. Desta forma, um cenário que se mostra interessante é a utilização destas técnicas em conjunto do dispositivo Leap Motion para reconhecimento de sinais da Libras.

No Brasil, segundo o Instituto Brasileiro De Geografia E Estatística (2012, p. 30) cerca de 5,1% da população brasileira possui alguma deficiência auditiva, e podem depender de uma língua de sinais. Esta porcentagem implica na necessidade de existir o ensino da Língua Brasileira de Sinais (Libras), visto que esta também é uma língua oficial do Brasil. Entretanto, o aprendizado da língua por crianças e adultos pode ser complicado sem uma forma de praticar e aprimorar os conhecimentos na língua. Portanto, no ensino de Libras há uma lacuna

existente onde pode ser utilizado o estado da arte da computação para criar uma ferramenta para auxiliar no ensino da Libras.

Diante deste contexto, propõem-se neste trabalho o desenvolvimento de uma aplicação para auxílio no ensino de sinais estáticos da Libras que utilize técnicas de aprendizado de máquina supervisionado e o dispositivo Leap Motion para reconhecimento dos sinais.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver uma aplicação para auxiliar no ensino-aprendizagem da Língua Brasileira de Sinais (Libras).

Os objetivos específicos do trabalho são:

- a) reconhecer sinais estáticos da Língua Brasileira de Sinais utilizando o dispositivo Leap Motion;
- b) comparar as técnicas de redes neurais artificiais e máquinas de vetores de suporte para o reconhecimento dos sinais;
- c) desenvolver um visualizador do modelo da mão em 3D.

1.2 RELEVÂNCIA DO TRABALHO

O acesso a informação garante que as pessoas possam exercer sua cidadania (FEBRABAN, 2006, p. 3). No caso dos surdos, a língua utilizada para ter acesso a tal informação é a Língua Brasileira de Sinais (Libras), entretanto, esta língua é pouco difundida, sem contar a falta de instrutores e literatura sobre esta língua que acaba afetando o acesso a tais informações (PAGLIUCA; FIÚZA; REBOUÇAS, 2006, p. 412).

Este trabalho torna-se relevante devido ao uso do dispositivo Leap Motion e técnicas de aprendizado de máquina no desenvolvimento de uma aplicação para que crianças e adultos possam praticar e aprimorar os seus conhecimentos sobre Libras, possibilitando uma melhor integração das pessoas surdas na sociedade e acesso a informação.

1.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: pesquisar sobre língua de sinais, Leap Motion, máquinas de vetores de suporte, redes neurais e trabalhos relacionados;
- b) elicitação de requisitos: reavaliar os requisitos de acordo com os objetivos do trabalho e levantamento bibliográfico;
- c) especificação do trabalho: elaborar diagramas de casos de uso e de classes de acordo com a *Unified Modeling Language* (UML) para ter uma visão geral do

2 REVISÃO BIBLIOGRÁFICA

Este capítulo está organizado em quatro seções. A seção 2.1 aborda línguas de sinais e a Libras. Na seção 2.2 é apresentado o dispositivo Leap Motion e o seu funcionamento. A seção 2.3 apresenta os conceitos envolvendo aprendizado de máquina, seguida pela apresentação da técnica de redes neurais artificiais na seção 2.3.1 e por máquinas de vetores de suporte na seção 2.3.2. Finalmente, na seção 2.3.4 são apresentados os trabalhos correlatos com abordagens para reconhecimento de gestos.

2.1 LÍNGUA BRASILEIRA DE SINAIS

O ensino de línguas de sinais data do século XVI com o francês Pedro Ponce de Leon que fundou uma escola de professores de surdos (CARVALHO, 2011). Mais adiante, no século XVIII o também francês Charles Michel de L'Epée criou os "sinais metódicos" em uma combinação de língua de sinais com a gramática sinalizada francesa (CARVALHO, 2011).

Diferentemente do que se pensa, as línguas de sinais não são universais e diferem entre países, ou até mesmo entre regiões de um mesmo país (ARAÚJO, 2009). Essa padronização se torna ainda mais difícil em países grandes como o Brasil, que em casos de cidades grandes é possível encontrar até certos "bairrismos" (PACHECO; ESTRUC, 2011, p. 8).

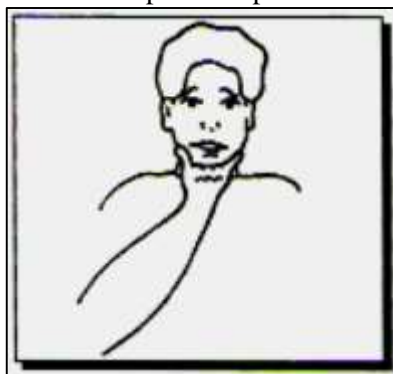
A língua de sinais não é mímica, entretanto apresenta sinais icônicos e sinais arbitrários. Em sinais icônicos segundo Pacheco e Estruc (2011, p. 14) os "[...] gestos fazem alusão à imagem do seu significado." (Figura 1). Segundo Pacheco e Estruc (2011, p. 15) os sinais arbitrários "São aqueles que não mantêm nenhuma semelhança com o dado da realidade que representam." (Figura 2).

Figura 1 - Sinal da palavra “telefone” em Libras.



Fonte: Pacheco e Estruc (2011, p. 14).

Figura 2 - Sinal da palavra “perdoar” em Libras.



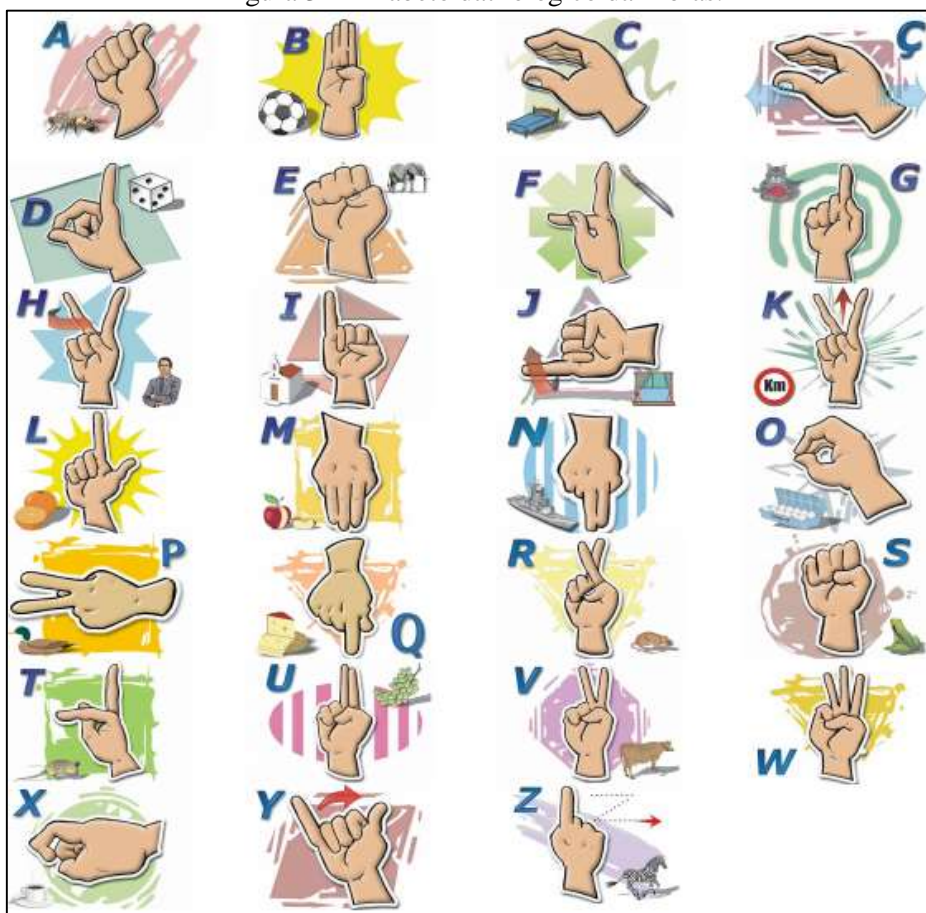
Fonte: Pacheco e Estruc (2011, p. 15).

A Língua Brasileira de Sinais (Libras) é um sistema linguístico de modalidade gestual-visual, baseado na Língua de Sinais Francesa e com estrutura gramatical independente da língua portuguesa brasileira. Entretanto, a Libras sofre influência direta da língua portuguesa brasileira por meio de adaptações por serem línguas em contato (ALBRES, 2004, p. 1). Esta língua teve origem com o professor Ernesto Huet durante o Império de Dom Pedro II e foi reconhecida como língua oficial no Brasil pela Lei Federal nº 10.436 em 24 de abril de 2002.

A Libras assim como qualquer outra língua possui estruturas sintáticas, semânticas, morfológicas e também passa por um processo de aprendizagem (ARAÚJO, 2009). O que é conhecido como uma palavra ou item léxico em línguas oral-auditivas é denominado um sinal nas línguas de sinais (SILVA, 2011, p. 1). Os sinais são criados a partir de combinações de formas, movimentos das mãos e pontos de articulação no corpo ou no espaço onde os sinais são realizados (SILVA, 2011, p. 1). Há casos específicos onde não há um sinal que corresponde a uma palavra, como no caso dos nomes. Neste caso as letras são sinalizadas separadamente (ARAÚJO, 2009). A soletração de palavras também é conhecida como datilologia.

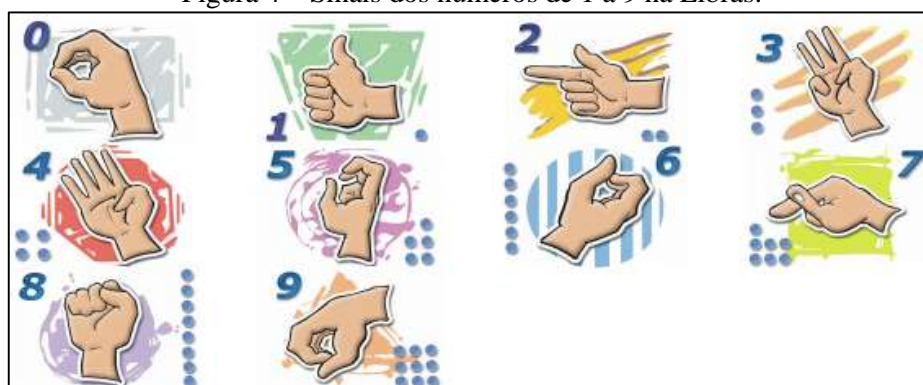
O alfabeto manual da Libras (alfabeto datilológico) é composto por 27 formatos, onde cada formato corresponde a uma letra do alfabeto do português brasileiro (GESSER, 2009, p. 30). A Figura 3 apresenta o alfabeto datilológico e a Figura 4 apresenta os sinais para representar os números na Libras. Destes 37 sinais é possível notar que apenas os sinais das letras “ç”, “h”, “j”, “k”, “y”, “z” necessitam movimento, os demais sinais são estáticos.

Figura 3 – Alfabeto datilológico da Libras.



Fonte: Pacheco e Estruc (2011, p. 37).

Figura 4 – Sinais dos números de 1 a 9 na Libras.



Fonte: Pacheco e Estruc (2011, p. 37).

Segundo Bento (2010, p. 33) "Um sinal pode ser articulado com uma ou duas mãos.

Um mesmo sinal pode ser articulado tanto com a mão direita quanto com a mão esquerda sem ocasionar mudança significativa, e portanto, não distintiva". Esta característica é possível porque os sinais são formados por unidades mínimas compostas pelos parâmetros fonológicos de Configuração de Mãos (CM), Ponto de Articulação (PA), Movimento (M), Orientação (O) e Expressão não-manual (ENM) (BENTO, 2010, p. 36).

A Configuração de Mãos (CM), segundo Bento (2010, p. 39) consiste nas "[...] formas das mãos e que podem ser da datilologia (alfabeto manual) ou demais formas feitas manualmente.". Bento (2010, p. 43) também afirma que este parâmetro "É considerado um articulador primário das línguas de sinais, sendo o parâmetro mais primitivo, pois não existe sinal sem configuração de mão."

O Ponto de Articulação (PA) é o local da mão configurada, podendo tocar o corpo ou estar em um espaço neutro vertical (BENTO, 2010, p. 39).

O Movimento (M), segundo Bento (2010, p. 46) "[...] consiste no deslocamento da Configuração de Mãos, durante a realização de um determinado sinal."

A Orientação (O) é a direção do sinal em relação aos demais parâmetros (BENTO, 2010, p. 39).

A Expressão não-manual (ENM) é o diferenciador que atua como complemento dos sinais manuais (BENTO, 2010, p. 39). Bento (2010, p. 52) complementa dizendo que "As expressões faciais/corporais ou não-manuais são de fundamental importância para o entendimento real do sinal, sendo que a entonação em Língua de Sinais é feita pela expressão facial."

Para Bento (2010, p. 39) "[...] as línguas de sinais apresentam como característica específica a simultaneidade na articulação dos fonemas. Assim, uma mesma CM e um mesmo movimento, com locação diferente, resulta em mudança de significado, formando um par mínimo."

2.2 LEAP MOTION

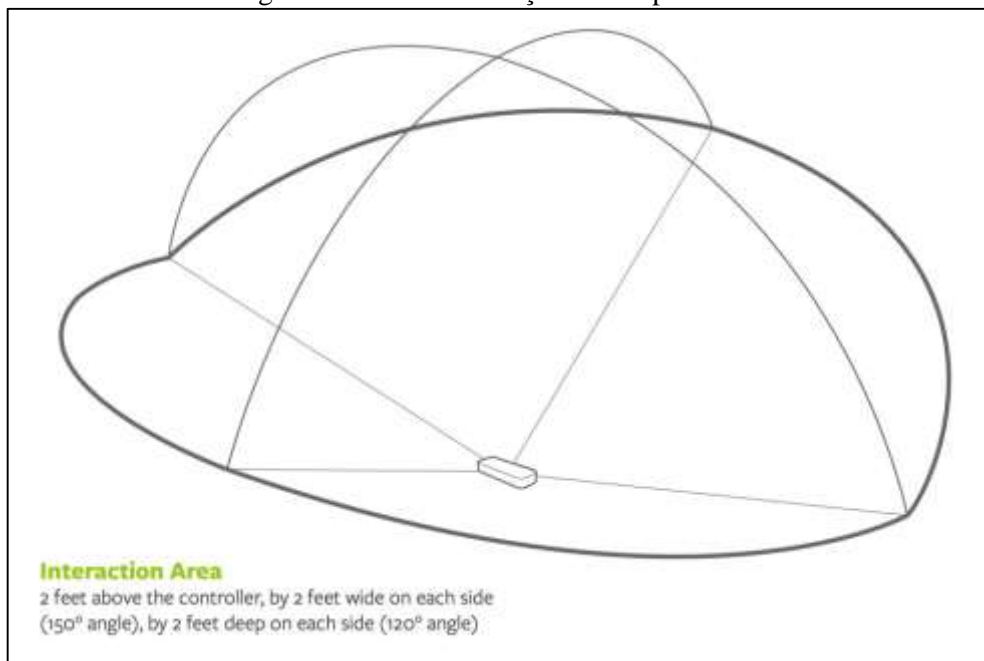
O Leap Motion (Figura 5) é um dispositivo desenvolvido pela empresa Leap Motion, Inc. e promete precisão milimétrica para detecção das mãos (WEICHERT et al., 2013, tradução nossa). O hardware do dispositivo é um tanto simples, consistindo apenas de duas câmeras estéreo e três LEDs infravermelhos (COLGAN, 2014, tradução nossa).

Figura 5 - Leap Motion.



O hardware do dispositivo oferece uma área de interação de aproximadamente 2,5 metros cúbicos (Figura 6), consistindo em 60,960 centímetros acima do controle, 60,960 centímetros ou 150° para cada lado e 60,960 centímetros ou 120° de profundidade para cada lado (COLGAN, 2014, tradução nossa).

Figura 6 - Área de interação do Leap Motion.



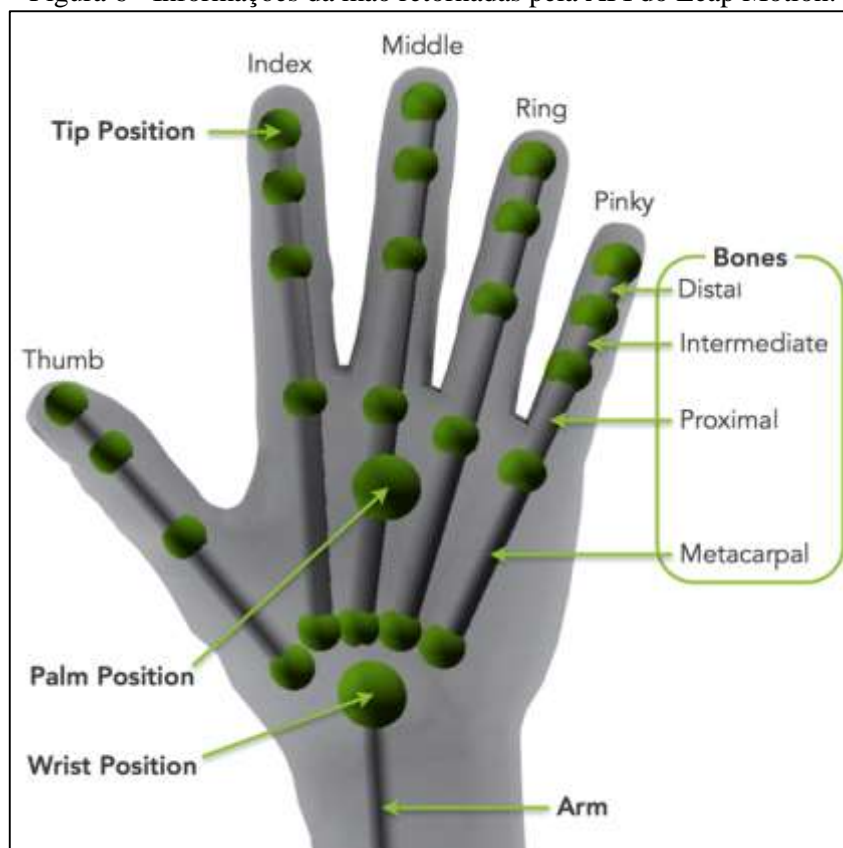
Fonte: Colgan (2014).

O dispositivo lê os dados dos sensores em sua memória local e faz os ajustes necessários para então enviar os dados para o computador, onde são feitos os processamentos matemáticos (COLGAN, 2014, tradução nossa). Segundo Colgan (2014, tradução nossa) "Diferentemente do que se pensa, o Leap Motion não gera um mapa de profundidade – em vez disso ele aplica algoritmos avançados aos dados puros do sensor". Após o processamento dos dados do sensor, uma camada com algoritmos de rastreamento extrai informações como dedos, ferramentas e infere a posição de objetos oclusos (COLGAN, 2014, tradução nossa). Também são aplicadas técnicas de filtragem para garantir coerência temporal dos dados

(COLGAN, 2014, tradução nossa).

Os dados processados são enviados no formato de frames através de um protocolo de transporte utilizando o protocolo TCP para aplicações nativas, ou WebSocket para aplicações web. De acordo com Davis (2014, tradução nossa) "No nível mais básico, a API do Leap Motion retorna os dados de rastreamento na forma de frames". Em cada frame é possível acessar as entidades rastreadas como mãos (Figura 6), dedos e ferramentas, objetos representando gestos reconhecidos, descrição do movimento das mãos na cena e também os dados puros do sensor (DAVIS, 2014, tradução nossa).

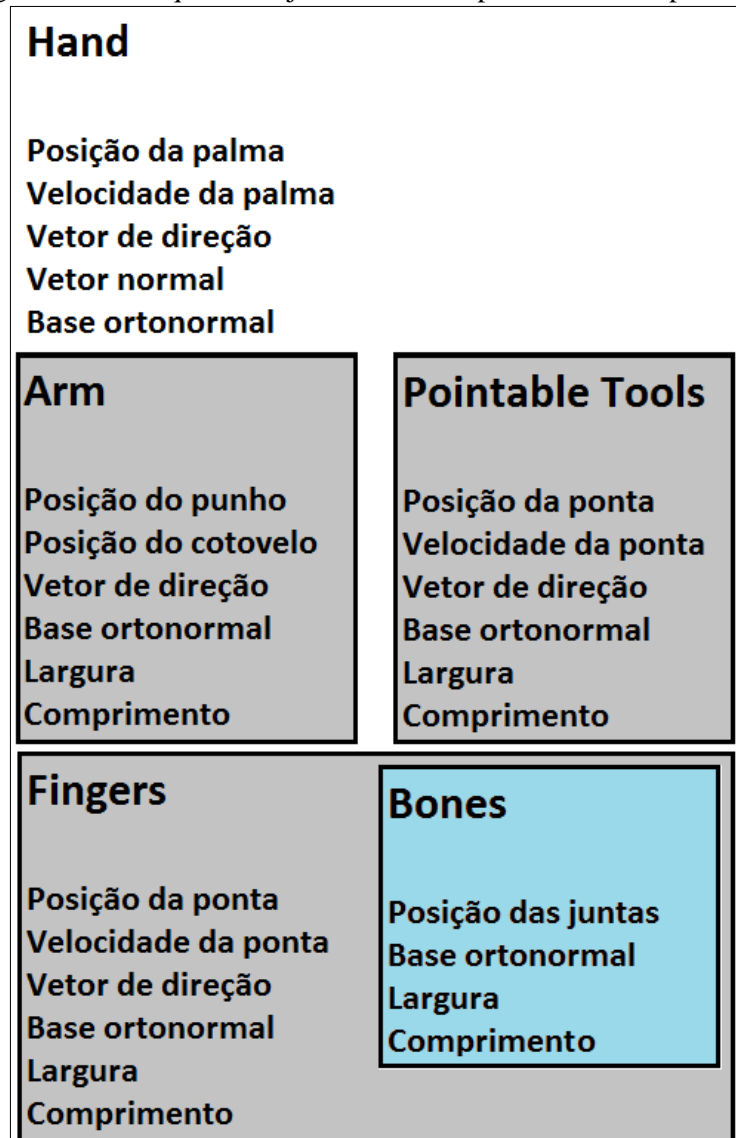
Figura 6 - Informações da mão retornadas pela API do Leap Motion.



Fonte: Davis (2014)

Toda entidade rastreada está na hierarquia do objeto Hand (Figura 7). A API também consegue distinguir entre mão direita e mão esquerda e informar a força dos movimentos agarrar e pinça (DAVIS, 2014, tradução nossa).

Figura 7 - Hierarquia de objetos rastreados pela API do Leap Motion.



A hierarquia apresentada na Figura 7 exibe como a API do Leap Motion retorna os dados do sensor. Inicialmente existe um objeto Hand com informações sobre a palma da mão. Este objeto ainda é composto pelo objeto Arm que contém informações sobre o punho do usuário e uma coleção de ferramentas apontadoras (Pointable Tools), que podem ser canetas, por exemplo. Por fim, o objeto Hand também contém uma coleção de dedos (Fingers) que traz informações sobre cada dedo de cada mão, onde cada dedo contém um conjunto de informações sobre os ossos (Bones) que formam o dedo.

2.3 APRENDIZADO DE MÁQUINA SUPERVISIONADO

Desde a invenção dos computadores, sempre se imaginou se pudéssemos programá-los para aprender, se aprimorarem automaticamente com experiência (MITCHELL, 1997, p. 1,

tradução nossa). Mitchell (1997, p. 2, tradução nossa, grifo do autor) também define o aprendizado de máquina da seguinte forma “Um programa de computador é dito que **aprende** pela experiência E com respeito a alguma classe de tarefas T e medindo o desempenho P , se o desempenho nas tarefas T , medido por P , melhora com a experiência E ”.

O aprendizado de máquina pode ser dividido nas seguintes classes: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço, onde o fator que determina a classe a ser utilizada é o tipo de feedback disponível para o algoritmo de aprendizado (RUSSEL; NORVIG, 2003, p. 650, tradução nossa).

Dada uma função f , as técnicas de aprendizado de máquina supervisionado tentam recuperar esta função através do aprendizado de uma função h , também conhecida como hipótese. O aprendizado desta função ocorre através da inferência sobre exemplos $(x, f(x))$, onde x é uma entrada de f e $f(x)$ o resultado de f para a entrada x (RUSSEL; NORVIG, 2003, p. 651, tradução nossa). Segundo Russel e Norvig (2003, p. 651, tradução nossa, grifo do autor) este procedimento pode ser descrito como "Dada uma coleção de exemplos de f , retorne uma função h que aproxime f ". Se esta função f possui saídas discretas, esta função é denominada uma função de classificação. Entretanto, se as saídas desta função são contínuas, então esta função é denominada uma função de regressão.

Os exemplos $(x, f(x))$ geralmente são representados utilizando uma tabela no formato atributo-valor como na Figura 8 (SANCHES, 2003, p. 16). Esta tabela é composta por n linhas de exemplos $E = \{E_1, E_2, \dots, E_n\}$ e m colunas de atributos $X = \{X_1, X_2, \dots, X_m\}$, onde a combinação x_{ij} representa o j -ésimo atributo do i -ésimo exemplo (SANCHES, 2003, p. 16). Por fim o valor y_i é a classe atribuída para o exemplo E_i . Desta forma o par de exemplo $(x, f(x))$ pode ser descrito como (E_i, y_i) .

Figura 8 – Exemplos na forma de uma tabela atributo-valor.

	X_1	X_2	\dots	X_m	Y
E_1	x_{11}	x_{12}	\vdots	x_{1m}	y_1
E_2	x_{21}	x_{22}	\vdots	x_{2m}	y_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
E_n	x_{n1}	x_{n2}	\vdots	x_{nm}	y_n

Fonte: Sanches (2003, p. 16).

Uma boa hipótese (h) irá prever corretamente exemplos ainda não vistos. A dificuldade em fazer esta afirmativa é uma das dificuldades do aprendizado de máquina, porque é difícil

dizer quando h é uma boa aproximação de f (RUSSEL; NORVIG, 2003, p. 651, tradução nossa). A tentativa de recuperar f pode acabar com várias hipóteses consistentes e nesse caso é preferível a escolha da hipótese mais simples consistente com os dados.

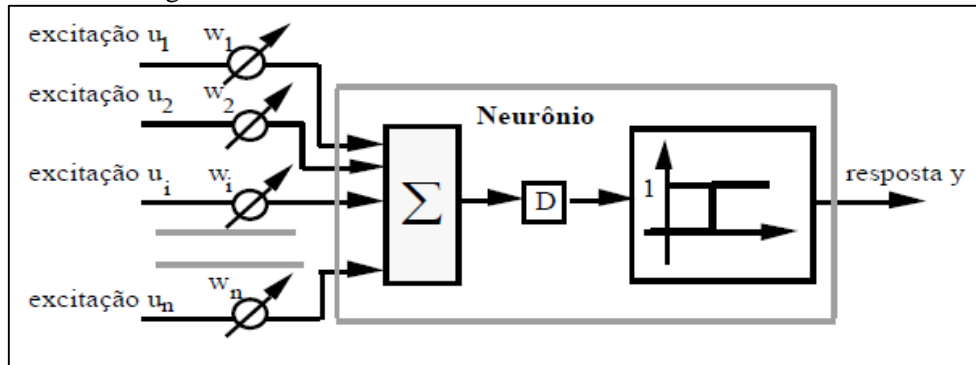
2.3.1 Redes neurais artificiais

O estudo de redes neurais artificiais surgiu na tentativa de simular o comportamento inteligente em máquinas através das conexões entre neurônios e vem desde 1943, com o desenvolvimento do primeiro neurônio biológico artificial, o *Psychon* (VELLASCO, 2007, p. 3; VIEIRA; ROISENBERG, 2004, p. 1). Uma década depois, em 1953, Frank Rosenblatt, Charles Wightman e outros pesquisadores desenvolveram o primeiro neurocomputador bem sucedido e posteriormente sustentaram os modelos *perceptron* e redes multinível (MLP) (VELLASCO, 2007, p. 3). Quase duas décadas depois dos avanços de Rosenblatt e Wightman, John Hopfield criou um modelo de rede neural com conexões recorrentes baseada em aprendizado não supervisionado (VELLASCO, 2007, p. 4). Ainda na década de 1980, os modelos de redes neurais reapareceram devido a teoria das redes multinível treinadas pelo algoritmo *backpropagation* (VELLASCO, 2007, p. 4).

O uso de redes neurais é adequado quando os dados de treinamento podem ser ruidosos e complexos, como os dados vindos de câmeras ou microfones (MITCHELL, 1997, p. 83, tradução nossa). Para Vieira e Roisenberg (2004, p. 2) as redes neurais "são dispositivos que permitem tolerância a ruídos e falhas, o que advém da capacidade de generalização das redes, sendo capaz de suportar ruídos e imperfeições no ambiente, ainda mantendo o resultado desejado." e para Vellasco (2007, p. 5) as redes neurais "[...] constituem uma importante técnica estatística não-linear capaz de resolver uma gama de problemas de grande complexidade."

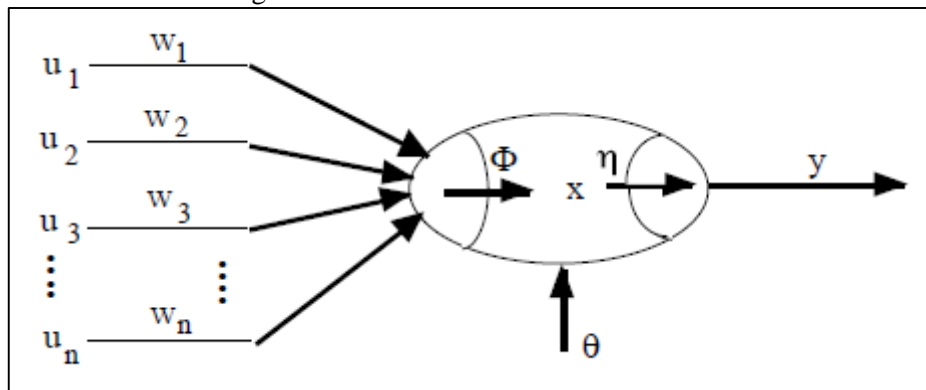
A Figura 9 apresenta o modelo de neurônio binário de McCulloch e Pitts, enquanto a Figura 10 apresenta o modelo geral de um neurônio artificial que é uma generalização do modelo de McCulloch e Pitts.

Figura 9 - Modelo de neurônio binário de McCulloch e Pitts.



Fonte: Barreto (2002, p. 14).

Figura 11 - Modelo de neurônio artificial.



Fonte: Barreto (2002, p. 15).

$u_1 \dots u_n$ são as entradas do neurônio atual, compostas pelas saídas da camada anterior de neurônios (VELLASCO, 2007, p. 5)

Os parâmetros $w_1 \dots w_n$ são os valores aprendidos após n -apresentações dos padrões que se deseja reconhecer (VELLASCO, 2007, p. 5)

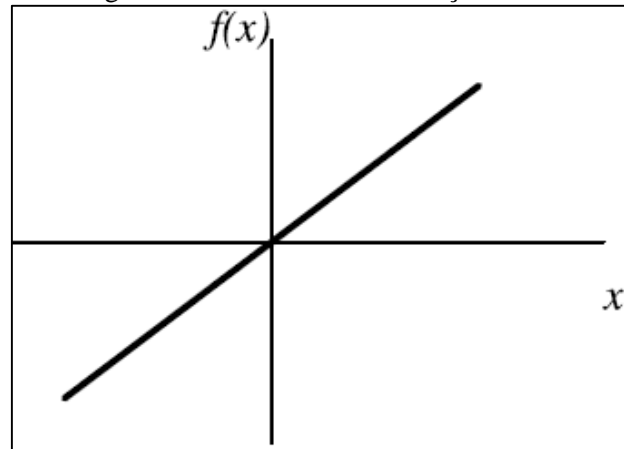
Segundo Vellasco (2007, p. 6) Φ "[...] representa a combinação linear dos pesos. Corresponde a soma ponderada da entrada pelos pesos."

θ é o termo polarizador que define o domínio dos valores de saída (VELLASCO, 2007, p. 6). Segundo Vellasco (2007, p. 6) "Como em cada neurônio chega a soma ponderada de todas as entradas, então o polarizador aparecerá associado a uma entrada fixa +1 ou -1."

y é a saída do neurônio após a aplicação da função η .

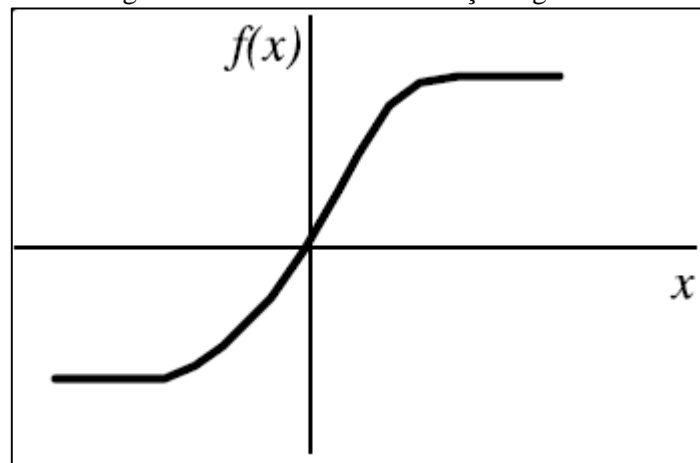
A função η também é conhecida como função de ativação e refere-se a parte não-linear de cada neurônio restringindo a amplitude do intervalo de saída do neurônio (VELLASCO, 2007, p. 6). A função linear (Figura 12) e a função sigmoide (Figura 13) são exemplos de funções de ativação.

Figura 12 - Gráfico de uma função linear.



Fonte: Cardon e Müller (1994, p. 8).

Figura 13 - Gráfico de uma função sigmoide.



Fonte: Cardon e Müller (1994, p. 10).

Existem basicamente dois tipos de arquitetura para redes neurais, as redes recorrentes e as redes não recorrentes (diretas).

As redes neurais diretas são ditas "sem memória", porque não possuem realimentação de suas saídas para suas entradas (VELLASCO, 2007, p. 10). Esta arquitetura pode ser formada por apenas uma camada ou por múltiplas camadas, contendo uma camada de neurônios para entrada, outra para saída e uma ou mais camadas ocultas (VELLASCO, 2007, p. 10). Segundo Vellasco (2007, p. 10-11) "Um exemplo desse tipo de arquitetura são as redes do tipo "feedforward", em que o sinal é sempre propagado para frente, da entrada para a saída". Segundo Vieira e Roisenberg (2004, p. 6) "Atualmente as redes neurais diretas são as mais utilizadas, principalmente pelo advento da popularização do algoritmo de treinamento backpropagation."

As redes recorrentes são ditas "com memória", porque possuem realimentação de suas

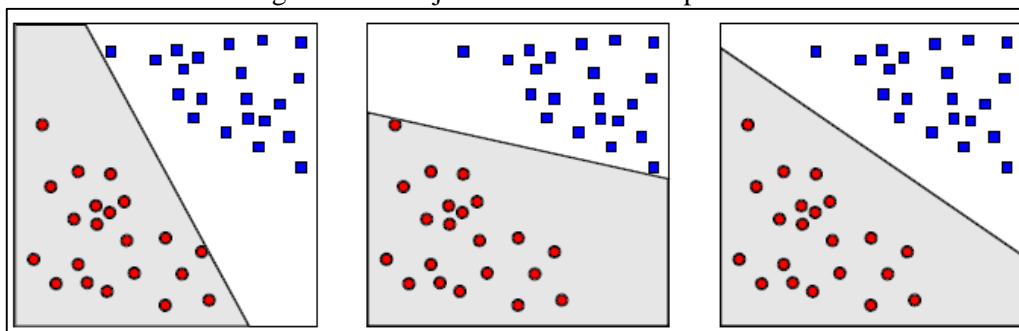
saídas para suas entradas, sendo que nesse caso, as suas saídas são determinadas pelas entradas atuais e pelas saídas anteriores (VELLASCO, 2007, p. 11). O algoritmo utilizado para treinamento é o retropropagação de erro.

2.3.2 Máquinas de vetores de suporte

As máquinas de vetores de suporte (SVM) se baseiam na teoria de aprendizado estatístico de Vapnik (LORENA; CARVALHO, 2007, p. 1). Essa teoria usa a ideia de um hiperplano que separa duas classes e maximiza uma margem com a maior distância possível entre o hiperplano e as instâncias em cada lado (KOTSIANTIS, 2007, p. 260, tradução nossa). Segundo Silva (2008, p. 37, grifo do autor) "[...] tal Teoria orienta a busca pela melhor solução para um dado problema de aprendizado através da minimização não só do erro de treinamento, mas também da complexidade do modelo obtido, o que resulta em um dos principais pontos fortes das SVMs: a **alta capacidade de generalização**".

O modelo de SVMs foi formulado para problemas binários, ou seja, conjuntos linearmente separáveis como na Figura 14 (SOUZA, 2013, p. 108). Entretanto a formulação foi estendida para trabalhar com conjuntos de dados mais gerais e também com fronteiras não lineares (LORENA, CARVALHO, 2007, p. 52-53).

Figura 14 - Conjuntos linearmente separáveis.



Fonte: Tahim (2010, p. 1).

As SVMs trazem a vantagem de terem um aprendizado convexo, onde os mínimos local e global coincidem (SOUZA, 2013, p. 108). Além disso, segundo Kotsiantis (2007, p. 261, tradução nossa), o “[...] modelo de complexidade de uma SVM não é afetado pelo número de características encontradas nos dados de treinamento (o número de vetores de suporte selecionados pelo algoritmo de aprendizado de SVM é geralmente pequeno)”. Entretanto, uma dificuldade ao trabalhar com SVMs é que todos os dados de treinamento devem estar presentes antes do aprendizado começar (SOUZA, 2013, p. 108).

As SVMs lineares utilizam como limite de decisão um hiperplano separando duas regiões no espaço (TAHIM, 2010, p. 1). Segundo Lorena e Carvalho (2007, p. 53, grifo do

autor) "Seja T um conjunto de treinamento com n dados $x_i \in X$ e seus respectivos rótulos $y_i \in Y$, em que X constitui o espaço dos dados e $Y = \{-1, +1\}$ ". Assim é possível definir uma SVM como linear se T puder ser classificado como $+1$ ou -1 .

O Quadro 2 mostra a equação de um hiperplano para separar X linearmente em duas regiões: $w \cdot x + b > 0$ e $w \cdot x + b < 0$. De acordo com Lorena e Carvalho (2007, p. 53) "[...] $w \cdot x$ é o produto escalar entre os vetores w e x , $w \in X$ é o vetor normal ao hiperplano descrito [...]".

Quadro 2 – Equação de um hiperplano.

$$f(x) = w \cdot x + b = 0$$

Fonte: Lorena e Carvalho (2007, p. 53).

Dada a equação do hiperplano (Quadro 2), é possível expressar uma função de decisão (Quadro 3) para o vetor x que retorna o rótulo $+1$ ou -1 predito para o vetor x (TAHIM, 2010, p. 1).

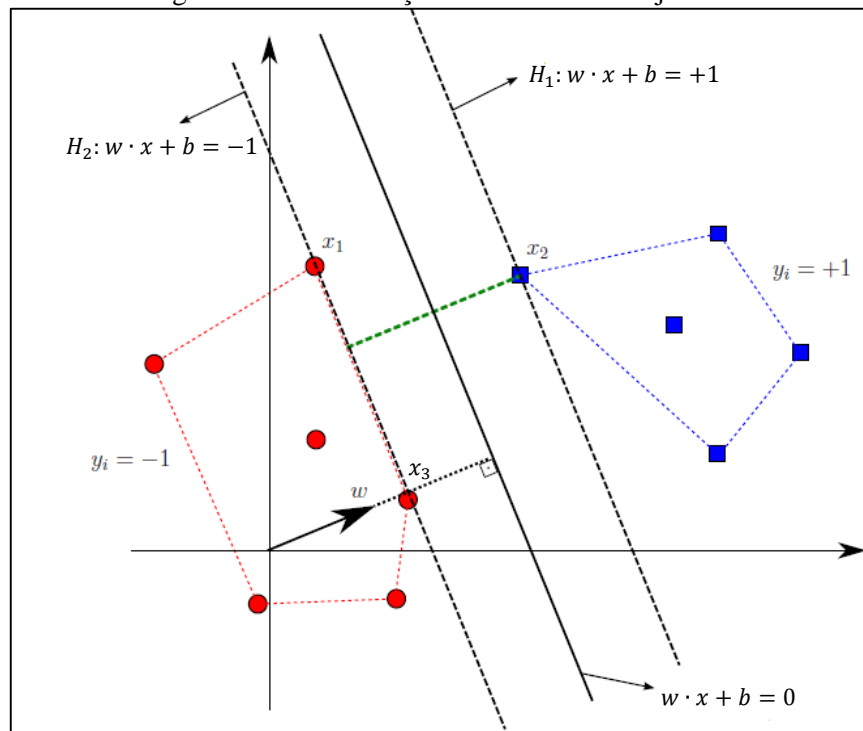
Quadro 3 – Função de decisão sobre um hiperplano

$$g(x) = \text{sgn}(f(x)) = \begin{cases} +1 & \text{se } w \cdot x + b > 0 \\ -1 & \text{se } w \cdot x + b < 0 \end{cases}$$

Fonte: Lorena e Carvalho (2007, p. 53).

A Figura 15 apresenta um exemplo de classificação linear. Neste exemplo é possível notar que o hiperplano ótimo ($w \cdot x + b = 0$) é sempre perpendicular ao centro da reta que liga os hiperplanos H_1 e H_2 (TAHIM, 2010, p. 2). Os pontos x_1 , x_2 e x_3 são os pontos de vetores de suporte e a solução é representada como uma combinação linear destes pontos apenas, os outros pontos são ignorados (KOTSIANTIS, 2007, p. 261, tradução nossa). Devido a escolha destes poucos vetores de suporte, segundo Kotsiantis (2007, p. 261, tradução nossa), as "[...] SVMs são bem adaptadas para lidar com tarefas de aprendizado onde o número de características é grande com respeito ao número de instâncias de treinamento."

Figura 15 - Classificação linear de dois conjuntos.



Fonte: Adaptação de Tahim (2010, p. 3).

2.4 TRABALHOS CORRELATOS

O reconhecimento de sinais da Libras tem se tornado um tema de pesquisa que chama a atenção de pesquisadores nos últimos anos. É possível citar o trabalho de Souza (2013), onde o objetivo é reconhecer sinais estáticos, sinais dinâmicos (com movimento) e soletração na Libras. O trabalho de Santin (2013) também objetiva o reconhecimento do alfabeto datilológico na Libras. Para o dispositivo Leap Motion, é possível destacar a biblioteca LeapGesture desenvolvida por Nowiki et al. (2014).

2.4.1 Gesture recognition library for Leap Motion

Nowiki et al. (2014) apresenta o desenvolvimento de uma biblioteca para reconhecimento de gestos estáticos e dinâmicos para o dispositivo Leap Motion. Foram utilizadas máquinas de vetores de suporte para o reconhecimento de gestos estáticos e modelos ocultos de Markov para reconhecimento de gestos dinâmicos.

Segundo Nowiki et al. (2014, p. 19, tradução nossa) "As máquinas de vetores de suporte foram escolhidas porque existe uma fundamentação matemática sólida apoiando a simples ideia de maximização da margem entre classes". Como implementação de SVM foi utilizada a biblioteca libSVM, entretanto, devido a problemas de desempenho durante o treinamento para classificação dos gestos, também foram feitos testes com a biblioteca

libLinear. A troca da biblioteca melhorou o desempenho do treinamento, que passou de aproximadamente 12 horas, para 5 segundos com 5000 exemplos. Entretanto, a troca de biblioteca implicou em uma perda de aproximadamente 5% à 17% na precisão do reconhecimento.

Os modelos ocultos de Markov foram implementados de forma que cada estado sempre tem uma transição para ele mesmo e para o próximo estado. Segundo Nowiki et al. (2014, p. 35, tradução nossa) "as transições para si mesmo foram utilizadas para modelar diferentes velocidades dos gestos e permite conseguir um sistema mais robusto".

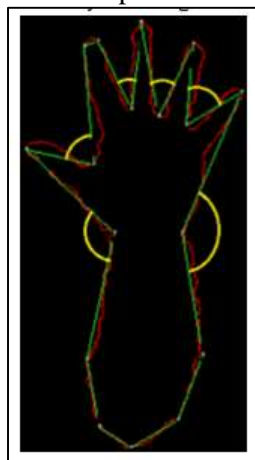
A biblioteca atingiu 99% de precisão no reconhecimento 5 gestos estáticos e 85% de precisão para o reconhecimento de 10 gestos estáticos. Na tarefa de reconhecimento de gestos dinâmicos, a precisão foi de 85% para um conjunto com seis gestos dinâmicos.

2.4.2 Ferramenta para transcrição do alfabeto datilológico para texto utilizando Microsoft Kinect

Santin (2013) desenvolveu uma ferramenta para reconhecimento dos símbolos do alfabeto datilológico utilizando o sensor *Microsoft Kinect*. A ferramenta armazena em uma matriz as distâncias capturadas pelos sensores de profundidade e câmera RGB do Microsoft Kinect. Após a montagem desta matriz, são extraídos e ordenados os pontos do contorno da mão. Para simplificar os segmentos de reta do contorno da mão, foi utilizado o algoritmo de Ramer-Douglas-Peucker.

Após a simplificação dos segmentos de reta, são extraídos os ângulos entre segmentos de reta que formam os dedos para o reconhecimento dos sinais (Figura 16). Inicialmente, o reconhecimento deveria ocorrer utilizando a visibilidade e oclusão dos dedos. Entretanto, como alguns sinais obrigam os dedos a ficarem unidos não foi possível a identificação precisa. Devido a este problema, optou-se por fazer o reconhecimento através da comparação dos ângulos extraídos com os ângulos de sinais previamente cadastrados em um arquivo XML. Desta forma, um sinal é detectado quando os valores extraídos do sensor forem o mais próximo possível dos valores cadastrados.

Figura 16 – Ângulos utilizados para reconhecimento de um sinal.



Fonte: Santin (2013, p. 40).

Nos testes executados, a ferramenta atingiu cerca de 70% de precisão na transcrição dos símbolos cadastrados separadamente. Este conjunto de sinais possuía os números de um à nove e as letras de A à Z, exceto as letras H, J, K, W, X e Z porque necessitavam de movimentação da mão ou tinham ângulos muito parecidos com outros símbolos. Entretanto, quando os símbolos foram cadastrados simultaneamente a precisão caiu para 12%, devido ao fato do crescimento da possibilidade de existirem símbolos concorrentes.

2.4.3 Reconhecimento de gestos da Língua Brasileira de Sinais através de Máquinas de Vetores de Suporte e Campos Aleatórios Condicionais Ocultos

Souza (2013) desenvolveu uma ferramenta para reconhecimento de sinais estáticos, dinâmicos e soletração da Libras. Para isto, foram utilizadas máquinas de vetores de suporte para reconhecimento de sinais estáticos e campos aleatórios ocultos condicionais para reconhecimento de sinais dinâmicos.

Esta ferramenta foi desenvolvida em duas camadas. A primeira camada extrai informações como: configuração das mãos, face e posição das mãos em relação a face. A segunda camada é responsável pela contextualização e classificação do sinal.

Para treinamento do reconhecimento de sinais estáticos foram utilizadas 300 amostras de 46 poses fundamentais da Libras, totalizando 13.800 amostras. Para treinamento do reconhecimento de sinais dinâmicos foram utilizados 939 quadros de 13 palavras do vocabulário da Libras, totalizando 139.154 quadros. Segundo Souza (2013, p. 155) "As sequências foram divididas em 10 conjuntos mutuamente exclusivos em antecipação ao uso de validação cruzada".

Os experimentos para reconhecimento de sinais estáticos utilizando máquinas de

vetores de suporte utilizaram as funções kernel Gaussianas, Quadráticas e Lineares. O desempenho dos classificadores, segundo Souza (2013, p. 163), foi medido "em termos do coeficiente Kappa (k) de Cohen, o número total de vetores de suporte necessários para a estratégia de votação e o número médio de avaliações de vetores de suporte encontrados pelo caminho de votação do DDAG".

Além dos testes com SVMs, também foram realizados testes utilizando redes neurais ($k = 0,9248$) que acabaram demonstrando um desempenho similar as SVMs ($k = 0,9268$). Entretanto o uso de redes neurais apresentou um custo computacional elevado para treinamento. Segundo Souza (2013, p. 167) "Os melhores valores para k foram encontrados entre 300 e 500 neurônios na camada intermediária".

O classificador escolhido como tendo o melhor desempenho foi o classificador DDAG composto por SVMs lineares. Este classificador foi escolhido devido ao aprendizado convexo e por possuir tempo de avaliação constante, visto que a avaliação não depende do número de vetores de suporte em cada máquina.

3 REQUISITOS DA APLICAÇÃO A SER DESENVOLVIDA

A aplicação de ensino-aprendizagem deverá:

- a) exibir um modelo 3D da mão do usuário a partir de uma interface gráfica (requisito funcional - RF);
- b) reconhecer os sinais estáticos do alfabeto datilológico (RF);
- c) possibilitar o treinamento dos valores para reconhecimento dos sinais (RF);
- d) utilizar a linguagem javascript para o desenvolvimento da aplicação para navegadores web (requisito não-funcional - RNF);
- e) utilizar o dispositivo Leap Motion com a biblioteca LeapJS para entrada de dados (RNF);
- f) utilizar a biblioteca three.js para exibir o modelo 3D da mão (RNF);
- g) utilizar a linguagem C# para reconhecimento dos sinais (RNF);
- h) utilizar a biblioteca Accord.Net que implementa algoritmos para máquinas de vetores de suporte e o algoritmo backpropagation para redes neurais (RNF);
- i) utilizar a API WebSocket do HTML 5 para integrar a camada javascript com a camada C# (RNF).

4 CONSIDERAÇÕES FINAIS

As pesquisas envolvendo acessibilidade, neste caso a Libras, tem se mostrado uma área interessante para aplicação de métodos computacionais e utilização de dispositivos que mudam a forma como interagimos com os computadores, como o caso do Leap Motion e do Microsoft Kinect.

Na literatura relacionada, é possível destacar a biblioteca LeapGesture desenvolvida por Nowiki et al. (2014) devido a alta precisão para classificação dos gestos e também a possibilidade de treinamento para reconhecimento de gestos estáticos e gestos dinâmicos. A ferramenta desenvolvida por Santin (2013) apresenta o uso do sensor Microsoft Kinect que evita a necessidade do tratamento de imagens para extração de informações sobre a mão do usuário. A ferramenta também demonstra a necessidade de utilizar algum algoritmo de classificação para obter resultados satisfatórios na tarefa de reconhecer os sinais da Libras. Já a ferramenta de Souza (2013) tenta unir a utilização do sensor Microsoft Kinect com o uso do algoritmo SVM e campos aleatórios ocultos condicionais para reconhecimento dos sinais da Libras. Esta abordagem se mostrou muito boa, atingindo altos valores de reconhecimento com uma curva de aprendizado convexa.

Como proposta, será utilizado o dispositivo Leap Motion como entrada de dados para as técnicas de máquinas de vetores de suporte e redes neurais na tarefa de reconhecimento de sinais da Libras. O Leap Motion foi escolhido devido à precisão do reconhecimento das mãos do usuário que evita a necessidade do processamento de imagens, possibilitando o foco da aplicação nas técnicas utilizadas para reconhecimento de sinais.

Com o desenvolvimento desta aplicação espera-se ajudar a preencher a lacuna existente no ensino da Libras existente atualmente, assim como possibilitar uma melhor integração de pessoas surdas a sociedade.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBRES, Neiva de Aquino. **História da Língua Brasileira de Sinais em Campo Grande - MS**. Petrópolis, 2005. Disponível em: <<http://www.editora-arara-azul.com.br/pdf/artigo15.pdf>>. Acesso em: 28/08/2014.

ARAÚJO, Ana Paula de. **Língua Brasileira de Sinais (LIBRAS)**. [S.l.], [2009]. Disponível em: <<http://www.infoescola.com/portugues/lingua-brasileira-de-sinais-libras/>>. Acesso em: 28/08/2014.

BARRETO, Jorge M. **Introdução às Redes Neurais Artificiais**. Florianópolis, 2002. Disponível em: <<http://www.inf.ufsc.br/~barreto/tutoriais/Survey.pdf>>. Acesso em: 12/09/2014.

BENTO, Nanci A. **Os parâmetros fonológicos**: Configuração de mãos, ponto de articulação e movimento na aquisição da Língua Brasileira de Sinais – um estudo de caso. 2010. 143f. Dissertação (Mestrado em Letras e Linguística) – Programa de Pós-Graduação em Letras e Linguística, Universidade Federal da Bahia, Salvador.

CARDON, André.; MÜLLER, Daniel M. **Introdução às Redes Neurais Artificiais**. Porto Alegre, 1994. Disponível em: <http://www.inf.ufrgs.br/~danielnm/docs/intro_rna.pdf>. Acesso em: 12/09/2014.

CARVALHO, Rodrigo J. Língua de Sinais Brasileiras e Breve Histórico da Educação Surda. **Revista Virtual de Cultura Surda e Diversidade**, [S.l.], n. 7, 2011. Disponível em: <<http://editora-arara-azul.com.br/novoeaa/revista/?p=466>>. Acesso em: 28/08/2014.

COLGAN, Alex. **How Does the Leap Motion Controller Work?**. [San Francisco], 2014. Disponível em: <<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>>. Acesso em: 30/08/2014.

DAVIS, Alan. **Getting Started with the Leap Motion SDK**. [San Francisco], 2014. Disponível em: <<http://blog.leapmotion.com/getting-started-leap-motion-sdk/>>. Acesso em: 30/08/2014.

FEBRABAN. **Gestão de pessoas com deficiência no ambiente de trabalho**: Coleção FEBRABAN de inclusão social. São Paulo, 2006. Disponível em: http://www.febraban.org.br/arquivo/cartilha/cartilha_gestao.pdf>. Acesso em: 22/09/2014.

GESSER, Audrei. **Libras?**: que língua é essa? : crenças e preconceitos em torno da língua de sinais e da realidade surda. São Paulo : Parábola, 2009. 87 p, il.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo Demográfico 2010**: Características gerais da população, religião e pessoas com deficiência. Rio de Janeiro, 2012. Disponível em: <
http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf>. Acesso em: 13/09/2014.

KOTSIANTIS, S. B. Supervised Machine Learning: A Review of Classification Techniques. **Informatica**, [Liubliana], v. 31, n. 3, 2007. Disponível em: <
http://www.informatica.si/PDF/31-3/11_Kotsiantis%20-%20Supervised%20Machine%20Learning%20-%20A%20Review%20of...pdf>. Acesso em: 12/09/2014.

LORENA, Ana C. CARVALHO, André C. P. L. F. de. Uma Introdução às Vector Support Machines. **Revista de Informática Teórica e Aplicada**, [Porto Alegre], v. 14, n. 2, 2007. Disponível em: <
http://www.seer.ufrgs.br/index.php/rita/article/view/rita_v14_n2_p43-67/3543>. Acesso em: 12/09/2014.

MITCHELL, Tom M. (Tom Michael). **Machine learning**. New York : McGraw-Hill, 1997. 414 p, il. (McGraw-Hill series in computer science).

NOWIKI, Michal et al. **Gesture recognition library for Leap Motion controller**. 2014. 65f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de ciência da computação, Universidade de Tecnologia de Poznan, Poznan.

PACHECO, Jonas; ESTRUC, Ricardo. **Curso Básico da LIBRAS (Língua Brasileira de Sinais)**. [S.l.], 2011. Disponível em: <
<http://www.surdo.org.br/Apostila.pdf>>. Acesso em: 12/09/2014.

PAGLIUCA, Lorita M. F.; FIÚZA, Nara L. G.; REBOUÇAS, B. de A. Aspectos da comunicação da enfermeira com o deficiente auditivo. **Revista da Escola de Enfermagem da USP**, [São Paulo], v. 41, n. 3, 2007. Disponível em:
http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0080-62342007000300010&lng=en&nrm=iso>. Acesso em: 21/09/2014.

RUSSELL, Stuart J. (Stuart Jonathan); NORVIG, Peter. **Artificial intelligence: a modern approach**. 2nd ed. Upper Saddle River, N.J : Prentice-Hall, 2003. xxviii, 1080p, il. (Prentice-Hall series in artificial intelligence).

SANCHES, Marcelo Kaminski. **Aprendizado de máquina semi-supervisionado**: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. 2003. 120f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2003. Disponível em: <
<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-12102003-140536/>>. Acesso em: 2014-09-19.

SANTIN, Diego M. **Ferramenta para transcrição do alfabeto datilológico para texto utilizando Microsoft Kinect**. 2013. 53f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SILVA, Regiane Ferreira da. **Língua Brasileira de Sinais**. [S.l.], [2011]. Disponível em: <http://academico.fead.br/informativo/html/pdf/003_6_0.pdf>. Acesso em: 28/08/2014.

SILVA, Marcelo M. **Uma abordagem evolucionária para o aprendizado semi-supervisionado em máquinas de vetores de suporte**. 2008. 106f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte.

SMITH, Chris et al. **The History of Artificial Intelligence**. [Washington], 2006. Disponível em: <<http://courses.cs.washington.edu/courses/csep590/06au/projects/history-ai.pdf>>. Acesso em: 13/09/2014.

SOUZA, César R. de. **Reconhecimento de gestos da Língua Brasileira de Sinais através de Máquinas de Vetores de Suporte e Campos Aleatórios Condicionais Ocultos**. 2013. 218f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de São Carlos, São Carlos.

TAHIM, André P. N. **Máquinas de vetores de suporte (SVM)**. Florianópolis, 2010. Disponível em: <<http://www.andretahim.com/publicacoes/Tutoriais/SVM/principalSvmTutorial.pdf>>. Acesso em: 12/09/2014.

VELLASCO, Marley M. B. R. **Redes neurais artificiais**. Rio de Janeiro, 2007. Disponível em: <<http://www2.ica.ele.puc-rio.br/Downloads/33/ICA-introdu%C3%A7%C3%A3o%20RNs.pdf>>. Acesso em: 12/09/2014.

VIEIRA, Renato C.; ROISENBERG, Mauro. **Redes neurais artificiais: um breve tutorial**. Florianópolis, [2004]. Disponível em: <<http://www.inf.ufsc.br/~l3c/artigos/TutorialRNA.pdf>>. Acesso em: 12/09/2014.

WEICHERT, Frank et al. Analysis of the Accuracy and Robustness of the Leap Motion Controller. **Sensors**, Basel, v. 13, n. 5, 2013. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690061/>>. Acesso em: 12/09/2014.