

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” – UNIVEM
PROGRAMA DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

FABRÍCIO BAPTISTA

F-LIBRAS
AMBIENTE INTEGRADO DE ENSINO-APRENDIZAGEM PARA LÍNGUA
BRASILEIRA DE SINAIS

MARÍLIA
2007

FABRÍCIO BAPTISTA

F-LIBRAS
AMBIENTE INTEGRADO DE ENSINO-APRENDIZAGEM PARA LÍNGUA
BRASILEIRA DE SINAIS

Dissertação apresentada ao Programa de Mestrado do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do Título de Mestre em Ciência da Computação. (Área de Concentração: Realidade Virtual)

Orientador:
Prof. Dr. José Remo Ferreira Brega

MARÍLIA
2007

BAPTISTA, Fabrício

F-LIBRAS - Ambiente Integrado de Ensino-Aprendizagem para Língua Brasileira de Sinais / Fabrício Baptista; orientador: José Remo Ferreira Brega. Marília, SP: [s.n.], 2007.

92 f.

Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília - Fundação de Ensino Eurípides Soares da Rocha.

1. LIBRAS 2. Realidade Virtual 3. Java 3D 4. Tradutores

CDD: 004

Fabício Baptista

F-Libras ambiente integrado de ensino-aprendizagem para língua brasileira de sinais

Banca examinadora da dissertação apresentada ao Programa de Mestrado da UNIVEM / FEESR, para obtenção do Título de Mestre em Ciência da Computação.

A Comissão Julgadora:

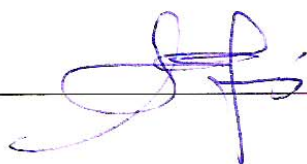
José Remo Ferreira Brega



Antonio Carlos Sementille



Romero Tori



Marília, 17 de setembro de 2007.

Uma máquina pode fazer o trabalho de
cinquenta pessoas comuns. Máquina alguma
pode fazer o trabalho de um homem incomum.
(Elbert Hubbard)

O único homem que não erra é aquele que
nunca fez nada.
(Roosevelt)

BAPTISTA, Fabrício – F-LIBRAS - Ambiente Integrado de Ensino-Aprendizagem para Língua Brasileira de Sinais. 2007. 92 f. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília. Fundação de Ensino Eurípides Soares da Rocha, Marília, 2007.

RESUMO

Grande parte dos deficientes auditivos brasileiros utiliza a Língua Brasileira de Sinais (LIBRAS) para a sua comunicação. No entanto, alguns encontram dificuldade para comunicar-se, pois o Português não é a sua língua-mãe, e possui uma ordem diferente da definida pela LIBRAS. Para minimizar este problema criou-se o F-LIBRAS, um Ambiente Virtual voltado para o domínio da LIBRAS. Ele deve suprir esta dificuldade de comunicação entre pessoas, especialmente daquelas com deficiência auditiva e pessoas que interagem de algum modo com essa comunidade. Em termos de usuários, incluem-se familiares, professores, alunos e monitores. Este ambiente deve proporcionar suporte para a integração de pessoas, sistemas computacionais e conteúdos, podendo ser utilizado tanto para aprendizagem como para treinamento e comunicação. As principais funções deste *software* são: a gravação, editoração e visualização de gestos e movimentos da Língua de Sinais, além da possibilidade de comunicação. Os gestos utilizados em LIBRAS são tridimensionais, dessa forma as técnicas de Realidade Virtual são apropriadas para a representação dos movimentos que compõe a língua de sinais.

Palavras-chave: Deficiente Auditivo, Realidade Virtual, LIBRAS, XML, Ambientes Virtuais.

BAPTISTA, Fabrício – F-LIBRAS – Including Environment of Teaching-Learning for Signal Brazilian Language. 2007. 92 f. Dissertation (Master in Computer Science) – University Center Eurípides from Marília. Teaching Foundation Eurípides Soares da Rocha, Marília, 2007.

ABSTRACT

The most of Brazilians hearing deficient uses the Signal Brazilian Language (LIBRAS) for their communication. However, some of them find difficulty to communicate each other, because Portuguese is not their native language, and it has a different position than that defined by Signal Brazilian Language (LIBRAS). For reducing this problem, it was created the F-LIBRAS, a Virtual Environment directed to dominate of Signal Brazilian Language (LIBRAS). It should provide this difficulty of communication between people, especially who suffers hearing deficiency and people who interact any way with this community. In due form of users, it includes relatives, teachers, students and monitors. This environment should provide support for including people, computer systems and contents; it can be used even for learning as training. The principal tasks of this program are: recording, publishing and visualizing of gestures and movements from Signal Language, besides the possibility of communication. The gestures used in Signal Brazilian Language (LIBRAS) are three-dimensional, this way the techniques of Virtual Reality are appropriated to represents the movements that compose the signal language.

Key-words: Hearing Deficient. Virtual Reality, LIBRAS (Signal Brazilian Language). XML (Extensible Mark up Language), Virtual Environments.

BAPTISTA, Fabrício – F-LIBRAS – Ambiente Integrado de Enseñanza-Aprendizaje para Lengua Brasileña de Señales. 2007. 92 f. Disertación (Maestrazgo en Ciencias de la Computación) – Centro Universitario Eurípides de Marília. Fundación de Enseñanza Eurípides Soares da Rocha, Marília, 2007.

RESUMEN

Gran parte de los deficientes auditivos brasileños utiliza la Lengua Brasileña de Señales (LIBRAS) para su comunicación. Entre tanto, algunos encuentran dificultad para comunicarse, pues el portugués no es su lengua-madre, y posee un orden diferente de la definida por la Lengua de Señales (LIBRAS). Para disminuir este problema se creó el F-LIBRAS, un Ambiente Virtual dirigido al dominio de la Lengua de Señales (LIBRAS). Él debe solucionar esta dificultad de comunicación entre personas, especialmente de aquellas con deficiencia auditiva y personas que interactúan de algún modo con esa comunidad. En relación a usuarios, inclúyanse familiares, profesores, alumnos y monitores. Este ambiente debe proporcionar soporte para la integración de personas, sistemas de computadoras y contenidos, pudiendo ser utilizado tanto para aprendizaje como para entrenamiento. Las principales tareas de este programa son: la grabación, editoración y visualización de modales y movimientos de la Lengua de Señales, además de la posibilidad de comunicación. Los modales utilizados en la Lengua de Señales (LIBRAS) son tridimensionales, de esa forma las técnicas de Realidad Virtual son apropiadas para la representación de los movimientos que componen la lengua de señales.

Palabras-clave: Deficiente Auditivo, Realidad Virtual, Lengua Brasileña de Señales (LIBRAS), XML, Ambientes Virtuales.

LISTA DE ILUSTRAÇÕES

Figura 1: Configuração de Mão (Brito, 2003).....	17
Figura 2: Movimento (Brito, 2003).....	17
Figura 3: Ponto de Articulação (Brito, 2003).....	18
Figura 4: Alfabeto Manual e Números em LIBRAS (Dicionário Libras, 2006).....	18
Figura 5: Esqueleto e Articulações da Mão (Deusdado, 2002).....	30
Figura 6: Interface gráfica do SkelMod3D (Martins et al., 2001).....	32
Figura 7: Hierarquia de Nós do H-Anim(HUMANOID ANIMATION WORKING GROUP, 2002).....	33
Figura 8: Arquitetura do XmlMessenger (Deitel, 2003).....	37
Figura 9: Exemplo de Documento em formato XML.....	38
Figura 10: Exemplo de DTD.....	39
Figura 11: <i>Profile</i> definidos no X3D. (<i>Consortium</i> , 2006).....	41
Figura 12: Representação Gráfica de um Grafo de Cena (Sowizral, 1999).....	44
Figura 13: Diagrama de Casos de Uso do F-LIBRAS.....	47
Figura 14: Interface Inicial do F-LIBRAS.....	48
Figura 15: Interface do módulo Editor.....	50
Figura 16: Definições de movimentos na barra de controles.....	51
Figura 17: Opções do gerador de quadros (<i>KeyFramer</i>).....	52
Figura 18: Configurações de Datilologia.....	53
Figura 19: Modularização do Ambiente.....	54
Figura 20: Grafo de cena do Ambiente.....	56
Figura 21: Nó Avatar.....	58
Figura 22: Grafo de Cena Datilologia.....	58
Figura 23: Diagrama de Classes do Aplicativo.....	59
Figura 24: Classe <i>TraduzHandler</i>	60
Figura 25: Implementação de Movimentação.....	62
Figura 26: Controle de ponto de vista.....	63
Figura 27: Interface do Módulo de Controles.....	65
Figura 28: Ações da Classe <i>Controles</i>	65
Figura 29: Gerador de Movimentos.....	66
Figura 30: Estrutura do arquivo XML.....	67
Figura 31: Exemplo de armazenamento em arquivo texto.....	68
Figura 32: Exemplo de Armazenamento em banco de dados relacional.....	69
Figura 33: Diagrama Entidade Relacionamento.....	69
Figura 34: Implementação da execução dos movimentos.....	70
Figura 35: Criação do <i>ObjectFile</i>	71
Figura 36: Exemplificação de movimento, quadros-chave principais.....	72
Figura 37: Detalhamento do Movimento: Cena 1.....	73
Figura 38: Detalhamento do Movimento: Cena 2.....	74
Figura 39: Detalhamento do Movimento: Cena 3.....	75
Figura 40: Simulação do interpretador membros superiores.....	77
Figura 41: Simulação do interpretador, Avatar completo.....	78
Figura 42: Gráfico comparativo do interpretador de sentenças.....	79
Figura 43: Representação da Datilologia.....	79

LISTA DE TABELAS

Tabela 1: Técnicas de interação em ambientes não-imersivos (Pinho 2001 apud Jacob, 2002)	23
Tabela 2: Listagem das articulações humanas (Arendi 1996 apud Maciel 2001)	30
Tabela 3: Descrição da Interface do F-LIBRAS	48
Tabela 4: Descrição da Interface de Edição	50
Tabela 5: Quadros por segundo da simulação de criação de movimento	76

LISTA DE ABREVIATURAS E SIGLAS

2D: Bidimensional

3D: Tridimensional

API: *Application Programming Interface*

AV: Ambiente Virtual

DOF: graus de liberdade, do inglês Degree of Freedom

DOM: *Document Object Model*

DTD: *Document Type Definition*

ebXML: negócio eletrônico XML

HTML: *Hyper Text Markup Language*

HV: Humanos Virtuais

IBGE: Instituto Brasileiro de Geografia e Estatística

JAXP: *Java API for XML Parsing*

JAXM: *Java API for XML Messaging*

LIBRAS: Língua Brasileira de Sinais

PC: *Personal Computer*

RV: Realidade Virtual

SOAP: *Simple Object Access Protocol*

VRML: *Virtual Reality Modelating Language*

X3D: *EXtensible 3D*

XML: *Extensible Markup Language*

XSL: *Extensible Style Language*

SUMÁRIO

INTRODUÇÃO	12
1. LIBRAS – LÍNGUA BRASILEIRA DE SINAIS	15
1.1 Sinais e os Aspectos da Sintaxe da LIBRAS.....	16
1.2 Datilologia – Alfabeto Manual.....	18
1.3 Informática na Socialização dos Deficientes Auditivos.....	19
2. AMBIENTES VIRTUAIS.....	21
2.1 Conceito.....	21
2.2 Imersão.....	22
2.3 Interação.....	23
2.4 Comunicação não verbal em Ambientes Virtuais.....	24
3. HUMANOS VIRTUAIS.....	27
3.1 Articulações Humanas e Humanos Virtuais.....	29
3.1.1 Articulações do Corpo Humano.....	29
3.1.2 Articulações em Humanos Virtuais.....	31
3.2 Controle de Movimento através da Cinemática e Dinâmica.....	34
3.3 Agentes Virtuais e Avatares.....	34
4. XML.....	36
4.1 Utilização do XML.....	36
4.2 A sintaxe XML.....	37
4.3 X3D.....	40
4.4 <i>Document Object Model</i> (DOM).....	41
5. JAVA	43
6. DESCRIÇÃO DO F-LIBRAS	46
6.1 Interface do F-LIBRAS.....	47
6.2 Desenvolvimento do F-LIBRAS.....	53
6.2.1 Modularização do Ambiente.....	54
6.2.2 Ambientes Virtuais do F-LIBRAS.....	56
6.2.3 Classes do F-LIBRAS.....	59
6.2.3.1 Classe InterpoladorBD.....	61
6.2.3.2 Classe Universo e Universo2.....	62
6.2.3.3 Classes do Módulo Editor.....	64
6.2.3.4 Classes do Módulo Interpretador.....	70
6.2.3.5 Classes do Módulo de Datilologia.....	71
6.3 Resultados.....	71
7. CONCLUSÕES.....	80
7.1 A Implementação.....	80
7.2 Formatos de Armazenamento.....	81
7.3 Trabalhos Futuros.....	81
REFERÊNCIAS	82
ANEXO A – INSTALAÇÃO DO F-LIBRAS	86

INTRODUÇÃO

Segundo Moura (2000), a educação e a inserção social dos deficientes auditivos constituem um sério problema, e muitos caminhos têm sido seguidos na busca de uma solução. A oficialização da Língua Brasileira de Sinais (LIBRAS) em abril de 2002 (Lei n. 10.436, de 24 de abril de 2002) começa a abrir novos caminhos.

Contudo, Capovilla (2004) salienta que existe certa carência de materiais de avaliação e ensino desenvolvidos especialmente para o deficiente auditivo brasileiro. Mais precisamente, pela carência de materiais instrucionais que permitam ministrar o ensino bilíngüe baseado na LIBRAS e de instrumentos de avaliação que permitam acompanhar o desenvolvimento da linguagem escrita e de sinais.

Neste contexto, surgiu a proposição de um Ambiente Virtual de ensino-aprendizagem para o domínio da LIBRAS e sua integração à sociedade. Procura-se então modelar um ambiente que permita a quebra de fronteiras do conhecimento e do preconceito na comunicação do deficiente auditivo em relação ao ouvinte normal. Um dos fatores primordiais é o desenvolvimento de uma interface para apresentar a LIBRAS da melhor forma possível, assim uma das tecnologias a ser utilizada para o desenvolvimento desta ferramenta é a Realidade Virtual (RV). A RV possui características de imersão, envolvimento e interação, utilizando um ambiente sintético tridimensional, permitem que o usuário ande pelo cenário e observe-o por ângulos variados, contribuindo para uma melhor percepção espacial.

Existem várias maneiras de se definir RV, uma definição clara e objetiva foi citada por Aukstakalnis (1992 apud, Isdale 1998), "a realidade virtual é uma maneira para que os seres humanos visualizem, manipulem e interajam com os computadores e os dados extremamente complexos". A RV proporciona uma interface mais rica em detalhes,

principalmente por proporcionar a visualização e interação em três dimensões, assim esta técnica seria apropriada para a apresentação dos movimentos que compõe a língua de sinais.

Este trabalho está estruturado em sete capítulos, além desta introdução, estes capítulos serão apresentados da seguinte forma.

No Capítulo 1, está apresentada a linguagem LIBRAS os seus aspectos, a estruturação e sua sintaxe, também é abordado como a Informática pode auxiliar na socialização dos deficientes auditivos.

O Capítulo 2 descreve as principais características dos Ambientes Virtuais, assim como alguns de seus recursos como interação e imersão, além de apresentar as formas de comunicação não-verbais em Ambientes Virtuais.

A representação da LIBRAS pode ser feita utilizando modelos humanos. No Capítulo 3 são apresentadas, características de Humanos Virtuais. Neste capítulo também são tratadas as articulações humanas em Humanos Virtuais, fator fundamental para a geração de movimentos.

Com o intuito de criar uma forma padronizada e estruturada de armazenamento e intercâmbio de informações em LIBRAS, o Capítulo 4 mostra a tecnologia XML (*eXtensible Markup Language*), seus principais conceitos e formas de utilização, além da especificação X3D, que constitui um padrão para troca de conteúdo 3D. Outro item abordado é o *Document Object Model* (DOM), uma interface de programação para arquivos XML.

No Capítulo 5 é apresentada linguagem Java e sua API¹ (*Application Programming Interface*) Java 3D, usada para desenvolvimento de programas que manipulam e mostram geometria tridimensional, uma API utilizada para transmissão de documentos XML.

O Capítulo 6 apresenta o desenvolvimento do projeto, assim como os objetivos e a modularização. Para ilustração do desenvolvimento, são apresentados diagramas, classes e

¹ API é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades.

códigos, que proporcionam uma melhor compreensão da arquitetura do F-LIBRAS. Com base nesta arquitetura é exemplificada a utilização do F-LIBRAS, onde se verifica o desempenho da ferramenta, medido através da taxa de quadros por segundo.

Partindo desta estrutura, o Capítulo 7 finaliza o trabalho, apresentado conclusões, além da sugestão de trabalhos que possam dar prosseguimento a este estudo.

1. LIBRAS – LÍNGUA BRASILEIRA DE SINAIS

Neste Capítulo será abordada a LIBRAS, sua estrutura e os aspectos que influenciarão no desenvolvimento de todo este trabalho. Mostrando como a Informática pode influenciar na socialização dos deficientes auditivos.

As linguagens oral e escrita são as formas mais comuns de comunicação, Goebel (1996) comenta que a língua utilizada por um indivíduo depende do grupo em que está inserido, para os ouvintes, a comunicação se estabelece em termos oral-auditivos, mas para os deficientes auditivos se estabelece em termos gestual-visuais, em que gestual significa o conjunto de elementos lingüísticos manuais, corporais e faciais necessários para a articulação e a significação cultural do sinal.

A linguagem de sinais deve ser considerada a língua natural dos deficientes auditivos, Dizeu & Caporali (2005) explicam que a criança com esta deficiência adquire esta linguagem de forma espontânea sem que seja preciso um treinamento específico, mas ainda é considerada por muitos profissionais apenas como gestos simbólicos.

A criança ouvinte desde seu nascimento é exposta à língua oral, dessa forma é fornecida para ela a oportunidade de adquirir uma língua natural, a qual irá permitir realizar trocas comunicativas, vivenciar situações do seu meio e, assim, possuir uma língua efetiva e constituir sua linguagem. Para a criança com deficiência auditiva deveria ser dada a mesma oportunidade, de adquirir uma língua própria para constituir sua linguagem. A linguagem de sinais no Brasil é conhecida como LIBRAS, que a partir da Lei nº 10436/2002 foi reconhecida como meio legal de comunicação entre os deficientes auditivos.

Skliar (1997 apud Pereira, 2004) comenta que a proposta do bilingüismo², não é a de isolar o deficiente auditivo numa comunidade de pessoas com esta deficiência, em que só se

² Bilingüismo do surdo, refere-se a língua oral da comunidade ouvinte (no caso do Brasil, o português) e a língua de sinais da comunidade com deficiência auditiva (Língua Brasileira de Sinais - Libras).

use a língua de sinais; a língua do ouvinte, pelo menos em sua versão escrita, é igualmente importante, mas considera necessário primeiro adquirir fluência na língua de sinais (a qual inclusive servirá de ponte para a leitura e a escrita), para depois aprender a língua dos ouvintes.

1.1 Sinais e os Aspectos da Sintaxe da LIBRAS

Segundo Quadros (1997) a LIBRAS é uma língua organizada espacialmente de forma altamente restringida. Analisar alguns aspectos da sintaxe de uma língua requer "enxergar" esse sistema que é visual-espacial não oral-auditivo.

A comunicação em qualquer língua é dotada de regras. Assim surge a necessidade de se conhecer a estrutura de formação das sentenças, com o intuito de analisar o sistemas sintático e morfológico da língua de sinais.

Na LIBRAS parece existir uma flexibilidade quanto à ordenação das palavras em uma sentença. Quanto às regras sintáticas, Quadros (1999) considera que a partir da intuição de falantes nativos da LIBRAS assume-se que há três possíveis formas de ordenar as palavras na LIBRAS: Sujeito-Verbo-Objeto(SVO), Objeto-Sujeito-Verbo(OSV) e Sujeito-Objeto-Verbo (SOV), as outras combinações são consideradas inaceitáveis.

Cusin (2004) resume que o sistema morfológico diz respeito à formação das palavras, considerando características relativas a gênero (masculino/feminino), grau (intensidade, movimento, velocidade), tempo (passado, presente, futuro) e negação.

Na língua de sinais, um sinal corresponde a uma palavra nas línguas orais-auditivas. Segundo Brito (2003), o sinal é formado a partir da combinação do movimento das mãos, com um determinado formato e em um determinado lugar, podendo este lugar ser uma parte do corpo ou um espaço em frente ao corpo.

Ferreira Brito (1995 apud Pereira 2004) estudou alguns aspectos muito importantes para a representação computacional dos sinais, referentes à constituição da LIBRAS. Ela apresentou três parâmetros primários que se combinam: a Configuração das Mãos (Figura 1), o Movimento (Figura 2), e o Ponto de Articulação (Figura 3), e também apresenta parâmetros secundários que são: a disposição das mãos, a orientação das mãos e a região de contato, além de seus componentes não manuais como a expressão facial.

De acordo com Fusco (2004) a configuração das mãos é o modo como estão posicionados os dedos, a representação da mão, ou das duas mãos (conforme o sinal). O movimento é realizado pela mão (ou mãos) ou pelos dedos, quando o sinal é realizado. O ponto de articulação é o local do corpo onde o sinal é realizado.



Figura 1: Configuração de Mão (Brito, 2003)



Figura 2: Movimento (Brito, 2003)

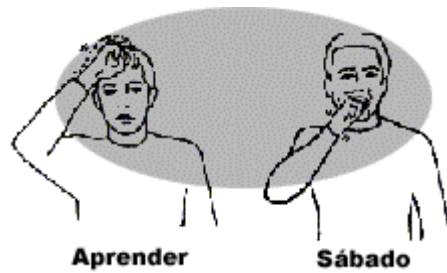


Figura 3: Ponto de Articulação (Brito, 2003)

Outras características, também muito importantes quanto à estrutura da Língua de Sinais, são os componentes não manuais da LIBRAS, tais como a expressão realizada pelo rosto, pela cabeça, pelo rosto e cabeça.

1.2 Datilologia – Alfabeto Manual

A Datilologia, ou alfabeto manual, é a forma de soletrar palavras com as mãos. Muito utilizado para nomes próprios, de pessoas, geográficos e palavras estrangeiras. No entanto, nem todos os nomes de pessoas são soletrados. Frequentemente cria-se um sinal específico para se referir a uma determinada pessoa. Não é universal. Países diferentes possuem sinais datilológicos diferentes.

O alfabeto manual, e os números utilizados pela datilologia LIBRAS são apresentados na Figura 4.

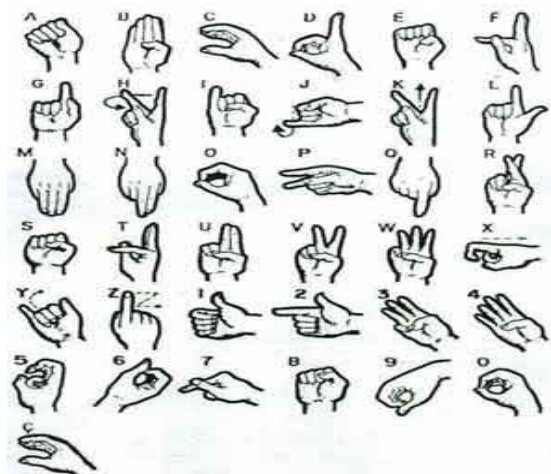


Figura 4: Alfabeto Manual e Números em LIBRAS (Dicionário Libras, 2006)

1.3 Informática na Socialização dos Deficientes Auditivos

O computador é utilizado nas diversas áreas do conhecimento para auxiliar o ser humano nas tarefas de seu cotidiano. Pode-se utilizá-lo na socialização dos deficientes auditivos, desenvolvendo ferramentas e meios para a comunicação entre eles e com indivíduos ouvintes.

Entretanto Dizeu & Caporali (2005) ressaltam que deve haver a conscientização de que toda esta tecnologia disponível não é de acesso a toda a população com deficiência auditiva, especialmente considerando-se a realidade social, cultural e econômica do país. Além do mais, ainda que toda essa tecnologia fosse acessível, ela por si não garante o desenvolvimento lingüístico e cultural desses indivíduos. Por isso é importante que esta comunidade tenha acesso a recursos tecnológicos, que possibilitem uma maior diversidade de leituras na interface no uso pelas pessoas.

Pellegrino (1996) expõe que a tecnologia, principalmente via computador, podem proporcionar:

- desenvolvimento da capacidade de comunicação do deficiente auditivo além da linguagem de sinais, estimulando a atenção necessária à expressão oro-facial do interlocutor;
- promover a integração da linguagem de sinais com a linguagem verbal;
- facilitar o acesso do deficiente auditivo e familiares a um material que possa dar continuidade em casa ao treinamento;
- observar e pesquisar a contribuição da tecnologia no atendimento aos deficientes auditivos; e

- valorizar e atualizar a pedagogia especial, comprometida com o processo de transformação social, de acordo com a necessidade real da educação dos deficientes auditivos.

Souza e Pinto (2003) concluem que em uma sociedade informatizada, todos os educandos, inclusive os deficientes auditivos, necessitam do computador presente em seu processo de aprendizagem. Para a educação dos indivíduos que se comunicam de forma visual, a utilização de ferramentas computacionais passam a ser essenciais. Dessa forma, o próximo capítulo especifica como os ambientes virtuais podem ajudar na interação.

2. AMBIENTES VIRTUAIS

Serão apresentados a seguir os conceitos de Ambientes Virtuais (AV), demonstrando os conceitos de imersão e interação, os quais possibilitam entender como esses ambientes podem proporcionar uma melhor interface entre usuários e computadores. Destacam-se também as formas de comunicação não verbais nesses ambientes.

2.1 Conceito

Rosa Junior (2003) resume, que no contexto computacional, Ambiente Virtual é um ambiente sintético composto por um conjunto de elementos que estimulam os sentidos do usuário, causando a presença virtual. Compreende uma representação, que consiste na modelagem do ambiente e no conjunto de agentes de efeito compostos de interfaces físicas e lógicas que interagem com esta representação.

Lamar (2001) comenta que a voz e os gestos são as formas mais naturais de comunicação entre os seres humanos, entretanto as interfaces ainda são muito primitivas forçando a nos adaptar aos requisitos de *hardware*, como por exemplo o teclado. Porém muitos pesquisadores têm se dedicado a criação meios mais naturais de interface homem-máquina, assim, nada melhor do que utilizar as técnicas de Realidade Virtual para atingir estes objetivos. Pinho (1996) destaca que a potencialidade da Realidade Virtual está exatamente no fato de permitir que sejam explicados alguns ambientes, processos ou objetos, não através de livros, fotos, filmes ou aulas, mas através da manipulação e análise virtual do próprio alvo do estudo.

A RV pode auxiliar o ser humano nas mais diversas áreas do conhecimento, seus recursos devem proporcionar sentimentos de imersão, envolvimento e interação aos usuários, para isto pode-se utilizar diversos dispositivos, convencionais e não convencionais, para

estimular os cinco sentidos do ser humano. A maioria dos aplicativos e dispositivos utilizam-se da visão, do tato e da audição.

2.2 Imersão

Rosa Junior (2003) explica que o fator imersão está diretamente relacionado com a capacidade do sistema em RV de envolver os sentidos do usuário, fazendo com que as ações e reações da interação entre usuário e Ambiente Virtual sejam transparentes e que o usuário tenha a sensação de “realmente” estar dentro do ambiente.

Considera-se que a imersão está relacionada aos dispositivos utilizados. Em termos de *hardware* considera-se RV imersiva, quando utiliza-se um conjunto de dispositivos chamados “imersivos” tais como: capacetes, telas de projeção ou salas de projeção, dispositivos rastreadores de movimento, som com posicionamento espacial, entre outros. A RV não imersiva é normalmente caracterizada pela simplicidade de equipamentos de interface homem-máquina, fazendo uso de monitores e dispositivos de interação simples (mouse, teclado, *joysticks* e similares).

Algumas aplicações são facilmente classificadas nestes termos, mas em diversos casos o nível de imersão varia, pois existe a utilização de combinações de equipamentos típicos de RV imersiva e não imersiva. Nos ambientes imersivos pode-se destacar como grande vantagem, a maior sensação de realmente estar no mundo virtual. Nos ambientes não imersivos, Cusin (2004) apresenta como grande vantagem a facilidade de acesso, já que utiliza-se de equipamentos mais acessíveis, de custo menor quando comparados a equipamentos para sistemas imersivos. Além disso, Rosa Junior (2003) destaca que o fator imersão não está somente relacionado ao equipamento utilizado, mas também à qualidade representativa do Ambiente Virtual (gráficos, som, tempo de resposta e elementos ativos de interface, são alguns exemplos).

2.3 Interação

Jacob (2002) explica que o processo de interação em Ambientes Virtuais consiste na capacidade reativa do sistema em detectar e responder a cada ação do usuário através de modificações instantâneas no Ambiente Virtual. Trata-se de um processo contínuo que tanto pode ser realizado através de dispositivos especiais como convencionais. A interação em ambientes tridimensionais é realizada com o objetivo de efetuar operações de seleção, manipulação e navegação no ambiente tridimensional.

Segundo Pinho (2000), por manipulação entende-se a tarefa de alterar algum parâmetro de um objeto tridimensional que acarrete mudança de suas propriedades geométricas (tamanho, posição ou orientação) ou não. Por seleção entende-se o processo de definir, dentre os objetos de um ambiente tridimensional, sobre qual deles se deseja operar. A navegação, por sua vez, é o processo de mudar a posição do observador do ambiente tridimensional.

Tabela 1: Técnicas de interação em ambientes não-imersivos (Pinho 2001 apud Jacob, 2002)

AMBIENTES NÃO-IMERSIVOS		
Seleção	Manipulação	Navegação
<p>Indireta: pode ser feita através de menus, ícones e vistas ortográficas em 2D.</p>	<p>Indireta: utiliza <i>widgets</i> 3D ou Controles comuns como botões, <i>sliders</i> e menus.</p>	<p>Indireta: pode ser realizada da seguinte forma:</p> <ul style="list-style-type: none"> ▪ com controle comum (<i>slider</i>, menus); ▪ com controles específicos como a barra de controle do VRML; ▪ dispositivos especiais: <i>joysticks</i>, direção de carro, entre outros.
<p>Direta: é necessário verificar qual região que se deseja selecionar: qual face, ou qual aresta ou vértice, ou então a seleção do objeto inteiro. A técnica para este tipo de seleção pode ser um envelope que envolve o objeto, ou também pode ser através da mudança de cor da região selecionada.</p>	<p>Direta: é realizada através de apontamento sobre a imagem. Este apontamento pode ser feito:</p> <ul style="list-style-type: none"> ▪ sobre vistas ortográficas; ▪ sobre projeção perspectiva; ▪ com manipulador tridimensional (setas tridimensionais). 	

Em ambientes tridimensionais não-imersivos, Deusdado (2002) comenta que a interação com um objeto é realizada com objetivo de efetuar operações de seleção, rotação, translação e escala sobre estes objetos ou então para realizar alterações na configuração do observador do Ambiente Virtual. As metáforas de interação em ambientes não imersivos podem ser classificadas de acordo com a forma como o usuário altera os valores que definem os parâmetros de rotação, escala, translação e navegação. A Tabela 1 demonstra as técnicas de interação em ambientes não-imersivos.

A interação em Ambientes Virtuais imersivos é realizada através de diversos dispositivos, como: capacetes, luvas, rastreadores de posição, entre outros. Mine (1995) apresenta as seguintes formas de interação em ambientes imersivos:

- direta (corpo do usuário atua diretamente sobre o objeto);
- com controles físicos (envolve o uso de botões, *joysticks*, pedais, volante de automóvel, etc.); e
- com controles virtuais (representação virtual de um dispositivo físico, tais como menus 3D, *widgets*, botões).

Dentre os modelos de interação apresentados acima, Jacob (2002) comenta que a forma básica de interação mais comum é a interação direta (utilizando a técnica da mão virtual, raio de apontamento e extensão do braço).

2.4 Comunicação não verbal em Ambientes Virtuais

A comunicação em Ambientes Virtuais não se restringe a comunicação lingüística ou verbal (texto ou voz), mas também a algumas formas não verbais da comunicação. A RV proporciona uma representação mais complexa do que as interfaces convencionais, podendo-

se explorar os modelos tridimensionais e animações a fim de proporcionar uma comunicação não verbal entre os usuários, Rosa Junior (2003) apresenta algumas formas para essa comunicação:

- a) Expressões faciais:** São muito representativas, indicando emoções e aspectos da personalidade do interlocutor. Sinalizam itens como o nível de interesse no assunto, descrença e surpresa, se devidamente sincronizada com a comunicação. Podem ser utilizados conjuntos de expressões pré-definidas associadas ao avatar do usuário, ou mesmo a utilização de equipamentos de captura de movimento facial.
- b) Olhar:** O olhar é um indicador geral de atenção, o qual representa o interesse do usuário em algum aspecto específico do ambiente ou disponibilidade de conversação. Também é utilizado como forma de sincronização da comunicação verbal em interações face a face. A representação do olhar através do avatar indica de alguma forma, a direção e sentido de interesse assumido pelo usuário. As representações auxiliares ficam por conta dos movimentos dos olhos e sobrancelhas.
- c) Gestos:** Mãos, cabeça e pés podem produzir uma vasta gama de gestos. São coordenados com a comunicação verbal e suportam várias funções de comunicação. Eles são usados como referência a conteúdo, tendo um excelente alcance social. Podem ser utilizados gestos para indicar localizações físicas no ambiente, sejam pontos de referencia ou objetos, para demonstrar algum processo ou comportamento.
- d) Postura:** É um indicativo de comportamento social associado a estados emocionais através de representações corporais. A posição e orientação do

corpo podem ser utilizadas para incluir ou excluir as pessoas em uma conversação de grupo.

- e) Ego-representação:** Está diretamente associada à representação pessoal, isto é, indica vários estados físicos e emocionais, por exemplo: estado civil, ocupação, personalidade, agressividade e formalidade. Esta representação é feita através da própria representação física do Avatar (geometria, cor, textura, etc.) e por representações auxiliares indicando atividade ou condições especiais. Por exemplo, indicar que o usuário se encontra ocupado em alguma atividade ou indisponível para comunicação direta.
- f) Contato Corporal:** O toque físico parece ter um significado primitivo de intimidade e produz um estímulo emocional aumentado. Esta forma de comunicações é utilizada para exprimir sentimentos ou alguma forma de saudação. O significado preciso de uma forma particular de toque depende da cultura. Normalmente não é representada em ambientes não imersivos, pois necessita obrigatoriamente de equipamento adicional. Mas em sistemas imersivos podem ser incluídas utilizando, por exemplo, luvas sensíveis ao toque.

Explorando-se então os diversos canais pelos quais a informação é trocada entre indivíduos em uma interação face a face real, chega-se a utilização da comunicação não verbal em Ambientes Virtuais. Partindo dessas definições, o próximo capítulo apresenta como os Humanos Virtuais se encaixam na idéia de proporcionar a comunicação a partir de um AV.

3. HUMANOS VIRTUAIS

Este capítulo conceitua Humanos Virtuais (HV), demonstrando como estas representações podem contribuir no desenvolvimento de aplicações que os utilizam para simulação de movimentos humanos.

Badler (1997) comenta que Humanos Virtuais são modelos computacionais de pessoas, que pode ser usados como substitutos em avaliações ergonômicas, projeto de produtos, linhas de montagem entre outros, antes da construção dos mesmos. Também podem ser representações em tempo real dos próprios usuários ou de outros, em Ambientes Virtuais.

Vilela (2002) define que HV são entidades interativas de *software* com graus variáveis de autonomia, inseridos em Ambientes Virtuais. Suas características gráficas e seus movimentos simulam basicamente o comportamento humano. Além disso, em vez de ficarem restritos a movimentos previamente definidos, Humanos Virtuais são capazes de comportar-se de forma autônoma e inteligente em ambientes dinâmicos, podendo apresentar até mesmo individualidade e personalidade.

Os HV podem ser implementados com diferentes enfoques, Torres et al. (2002) comentam que muitos projetos propostos apresentam Humanos Virtuais com aparência e funcionalidade bastante realistas, usando superfícies deformáveis e mecanismos de animação baseados em Física, mas não se importam com o comportamento do agente, o conceito de agente está caracterizado na seção 3.3. Outras abordagens propõem agentes cognitivos capazes inclusive de apresentar reações emocionais e personalidade própria, porém com representação gráfica pobre (textual em alguns casos). Existem ainda alguns trabalhos que procuram combinar a aparência realista com a credibilidade do comportamento, mas a maioria destes utiliza abordagens reativas para simular o comportamento.

Dentre essas funcionalidades os, Humanos Virtuais apresentam diversos aspectos, assim Thalmann (2001) destaca que há um grande potencial em aplicações de atividades humanas que podem utilizar-se de Humanos Virtuais, como:

- a) Interface Humano-computador: personagens com relativa autonomia e adaptabilidade ao comportamento do ser humano com o qual interage.
- b) Demonstração, Educação e Treinamento: sistemas de demonstração e treinamento conduzidos por personagens digitais.
- c) Marketing: utilização de sistemas como os já citados para atendentes virtuais (vendedores, recepcionista, assistentes para informações e instruções sobre produtos);
- d) Ergonomia e Desenvolvimento de Produtos: simulações do uso de produtos e de equipamentos de produção e manutenção dos mesmos;
- e) Medicina: Corpos Humanos Virtuais modelados com base em estudos anatômicos precisos, utilizados para treinamento em cirurgia, diagnóstico, entre outros;
- f) Forças Armadas: treinamento e simulação de batalhas e situações críticas ou rotineiras normalmente vividas por elementos da Forças Armadas;
- g) Indústria de Entretenimento: atores virtuais como personagens de animação, dublês ou figurantes, personagens fictícios mais complexos; e
- h) Pesquisa Científica: Arqueologia, Antropologia, Arquitetura, etc. podem utilizar de Humanos Virtuais para recriarem ambientes e avaliar as relações entre os seres humanos nestes espaços.

Assim verifica-se que a utilização de Humanos Virtuais para recriarem ambientes e avaliar as relações entre os seres humanos nestes ambientes, é de grande importância na criação de documentação e pesquisas que permitem avaliar o impacto da presença humana nestes espaços.

3.1 Articulações Humanas e Humanos Virtuais

Segundo Villamil (2002) as articulações são os componentes fundamentais do esqueleto humano e as estruturas responsáveis por permitir e restringir a mobilidade do corpo, acredita-se que criar um modelo de articulações humanas é o primeiro passo natural a ser dado em direção a modelar o corpo como um todo.

3.1.1 Articulações do Corpo Humano

Uma articulação é a união entre dois ou mais ossos. As articulações estão localizadas nas extremidades dos ossos, onde os ossos participantes estão em contato uns com os outros e onde pode ocorrer o movimento.

A Tabela 2 apresenta as articulações do corpo humano mostrando também os graus de liberdade de cada uma, fator importante para o desenvolvimento de movimentos em Humanos Virtuais, estas articulações são formadas por componentes complexos que permitem o seu perfeito funcionamento. A Figura 5 apresenta como exemplo, as articulações existentes na mão humana.

O movimento de uma articulação ocorre em torno de um ou mais eixos, ou simplesmente em um plano entre os ossos. São vários os fatores que fazem a restrições desses movimentos, como: a forma dos ossos, a presença de ligamentos³, musculatura, gordura, entre outros fatores como lesões e contusões que modificam a capacidade de movimentação.

³ Ligamentos: estruturas fibrosas e de pouquíssima elasticidade que tensionam um osso contra o outro de forma a mantê-los unidos no local onde se articulam.

Tabela 2: Listagem das articulações humanas (Arendi 1996 apud Maciel 2001)

Nome	Local	Tipo	DOF ⁴
Fronto-parietal	Cabeça	Sutura-serrátil	-
Interparietal	Cabeça	Sutura-serrátil	-
Parieto-temporal	Cabeça	Sutura escamosa	-
Vômer-esfenoidal	Cabeça	Esquindilese	-
Esfeno-etmoidal	Cabeça	Sincondrose	-
Entre os ossos nasais	Cabeça	Sutura homogênea	-
Dento-alveolar	Cabeça	Gonfose	-
Atlanto-ocipital	Pescoço/Cabeça	Trocóide/Pivô	1
Têmporo-mandibular	Cabeça	Condilartrose	2
Escapulo-umeral	Ombro	Esferóide	3
Úmero-ulnar	Cotovelo	Trocleartrose	1
Rádio-ulnar proximal	Antebraço	Trocóide/Pivô	1
Rádio-ulnar distal	Antebraço	Trocóide/Pivô	1
Rádio-cárpica	Punho	Elipsóide	2
Carpo-metacarpiana(polegar)	Mão	Selar	2
Carpo-metacarpiana(polegar)	Mão	Artródia	6-Plana
Metacarpofalangeanas	Mão	Condilartrose(Elipsóide)	2
Interfalangeanas da mão	Mão	Trocleartrose	1
Entre os corpos vertebrais	Coluna	Anfiartrose típica	3
Costo-vertebral	Tórax	Artródia	1
Costo-condral	Tórax	Sincondrose	-
Condroesternal	Tórax	Artródia	1
Coxofemoral	Quadril	Esferóide	3
Fêmur-tibial	Joelho	Condilartrose	2
Tíbio-fibular proximal	Canela	Artródia	3-Plana
Tíbio-fibular distal	Canela	Sindesmose	-
Calcâneo-tibial	Tornozelo	Trocleartrose	1
Tarso-metatarseanas	Pé	Artródia	6-Plana
Metatarsofalangeanas	Pé	Elipsóide	2
Interfalangeanas do Pé	Pé	Trocleartrose	1



Figura 5: Esqueleto e Articulações da Mão (Deusdado, 2002)

⁴ DOF: graus de liberdade, do inglês Degree of Freedom. O número de graus de liberdade define o número de eixos de coordenadas que podem ser manipulados simultaneamente durante o processo interativo.

Maciel(2001) conclui que, embora as articulações humanas possam ser comparadas a formas geométricas e junções mecânicas, tais como dobradiças, pinos, planos, esferas e cones, os movimentos requintados e as capacidades das articulações humanas (e dos animais em geral) ultrapassam qualquer junção fabricada pelo homem.

3.1.2 Articulações em Humanos Virtuais

Quando se fala em movimentação de Humanos Virtuais, uma área a ser estudada é a de articulações. Conforme abordado anteriormente, as articulações ou juntas, são responsáveis pela mobilidade do corpo humano. A simulação de movimento em modelos humanos pode ter vários enfoques. Alguns trabalhos primam pela construção do esqueleto, outros se concentram na modelagem de músculos, outros ainda buscam modelar a pele, o cabelo, entre outros. Outra característica a ser considerada é o tipo da aplicação:

- entretenimento em animação e jogos;
- simulação de expressões faciais;
- simulação de movimentos em atividades esportivas; e
- aplicações em ergonomia entre outras.

Maciel(2001) comenta que são muito raros na literatura de Computação Gráfica os trabalhos que abordam especificamente o problema do realismo anatômico das articulações. Podem-se destacar alguns trabalhos que de uma forma ou de outra abordaram estruturas articuladas.

Martins et al. (2001) desenvolveram um trabalho que visa a modelagem de corpos articulados, denominado SkelMod-3D, o ambiente possui uma interface que possibilita visualização e navegação em uma área 3D e uma janela 2D para configurações dos parâmetros das articulações, o usuário pode construir um esqueleto humano simples, ou seja, com informação topológica mas sem considerar formas humanas reais. Para que essa

descrição de corpo articulado possa ser posteriormente utilizada na simulação de movimento, o modelador oferecer a possibilidade de gravar o corpo modelado em um arquivo, de modo que o mesmo possa ser posteriormente importado para outros ambientes de animação. Optou-se então pelo uso de arquivos no formato padrão XML.

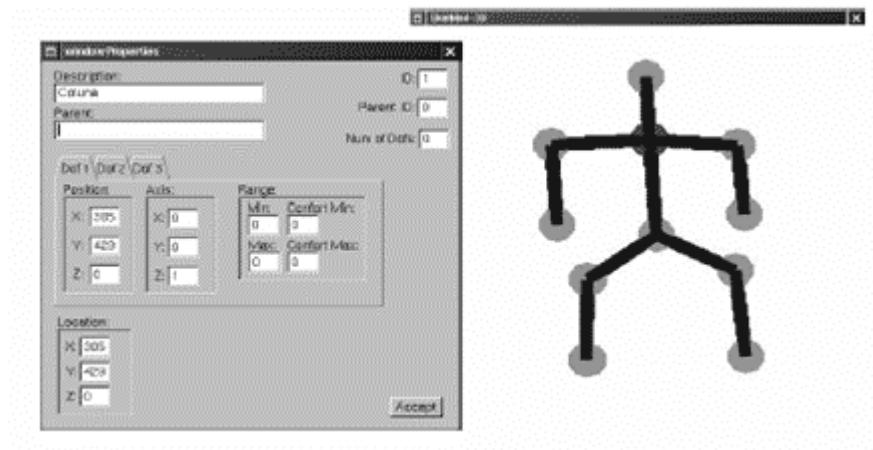


Figura 6: Interface gráfica do SkelMod3D (Martins et al., 2001)

Maciel(2001) desenvolveu um modelo anatômico articulado para a representação de corpos, portanto, neste modelo o corpo é a estrutura de mais alto nível do modelo conforme apresentado na Figura 6. Corpos são formados por um conjunto de articulações organizadas hierarquicamente. Cada uma destas, por sua vez, representa a relação entre duas partes adjacentes do corpo, e se modifica, ao longo do tempo, de acordo com os seus graus de liberdade. A cada articulação, podem ainda ser associados a objetos geométricos para representar ossos, músculos, pele, entre outros. Esses objetos, obrigatoriamente subordinados a uma articulação, têm sua posição alterada de acordo com as modificações feitas na relação representada pela articulação.

Este modelo permite a construção de sistemas gráficos para simulação de corpos humanos anatomicamente realísticos. A descrição do corpo articulado é gravada em arquivo em um formato tal que possa ser utilizada no simulador, e possa também ser importada para outros ambientes de animação e modelagem desenvolvidos futuramente.

Fusco (2004) utilizou-se da especificação de avatares H-Anim. Avatares H-Anim são representações 3D articuladas que descrevem humanóides animados, a sua hierarquia de articulações torna simples a movimentação das partes do corpo do humanóide, basta efetuar rotações nas articulações, e tudo que está ligado a ela é afetado pela transformação, possibilitando-se a movimentação das várias partes do corpo, a Figura 7 representa a hierarquia de nós da especificação H-Anim.

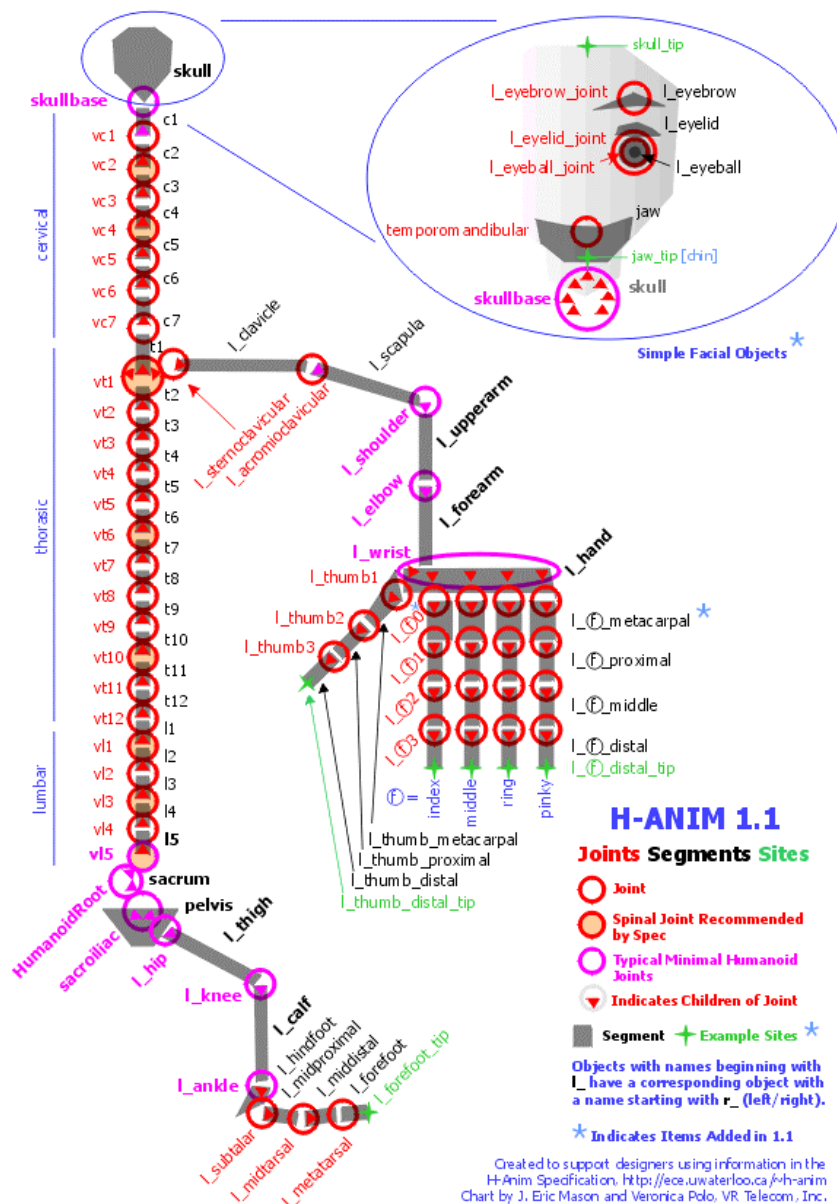


Figura 7: Hierarquia de Nós do H-Anim(HUMANOID ANIMATION WORKING GROUP, 2002)

3.2 Controle de Movimento através da Cinemática e Dinâmica

Segundo Nedel (2000), grande parte dos sistemas de animação por computador disponíveis atualmente fazem uso de modelos cinemáticos por serem de utilização mais fácil e intuitiva, além de consumir menos tempo computacional do que modelos dinâmicos. Por outro lado, o uso da dinâmica permite a geração de movimentos mais complexos, suaves e realísticos que muitas vezes são impossíveis de serem representados através da cinemática.

O uso de modelos dinâmicos, apesar de gerar movimentos bastante próximos dos reais, possui a desvantagem de não permitir que sejam parametrizadas as características do objeto no final da seqüência de animação, não podendo desta forma existir um controle rígido dos atores por parte do animador. Para resolver este tipo de problema são usados os conceitos de dinâmica inversa e cinemática inversa.

Deusdado(2002) analisa que a cinemática é o processo de cálculo da posição no espaço dos extremos de uma estrutura conectada dados os ângulos de todas a junções. Este processo é relativamente fácil e para ele há uma só solução. A cinemática inversa faz o contrário. Dados os pontos finais de uma estrutura, que ângulos estarão associados a estas junções para um posterior arquivo dos pontos finais. É de cálculo difícil, e há geralmente muitas ou até uma infinidade de soluções.

3.3 Agentes Virtuais e Avatares

Um Humano Virtual pode ser implementado basicamente de duas formas, Badler (1997) caracteriza agente como um Humano Virtual que é controlado por um computador, enquanto o avatar é controlado pelo usuário.

Segundo Torres et al. (2002) agentes inteligentes têm sido usados em diversos projetos na implementação de personagens virtuais. Um aspecto fundamental nesses sistemas

é a noção de credibilidade dos personagens, que está associada à capacidade que possuem de transmitir ilusão de vida. Em um primeiro momento, eram consideradas somente características ligadas à área de agentes, mas atualmente os projetos aliam o comportamento dos agentes a uma representação gráfica convincente dos mesmos.

Existem diversas arquiteturas disponíveis para planejar e implementar agentes, desde os reativos até os cognitivos. Os agentes reativos são muito simples e não possuem representação do ambiente nem do seu conhecimento, ficando este implícito em suas regras de comportamento. Estes agentes não possuem memória de suas ações passadas nem podem planejar seu comportamento futuro. Suas reações dependem unicamente da percepção do ambiente a cada instante. Os agentes cognitivos, ao contrário, mantêm uma representação explícita de seu ambiente e dos outros agentes presentes no sistema, têm memória do passado, podem comunicar-se diretamente e possuem um mecanismo de controle deliberativo, ou seja, raciocinam para decidir os objetivos a alcançar, os planos a seguir e as ações a executar (Alvares e Sichman, 1997 apud Torres, 2002).

Neste capítulo foi apresentada a conceituação e o estudo de Humanos Virtuais, e como as articulações são importantes na geração de movimentos, um dos focos deste trabalho. A seguir serão apresentadas as tecnologias utilizadas na implementação da ferramenta.

4. XML

O XML é um padrão mundial para troca de qualquer tipo de dado estruturado. Embora tenha sido, até o momento, muito mais usado para trocas de documentos estruturados pela *Web*, esse padrão foi desenvolvido para armazenar qualquer tipo de dado estruturado. Maciel(2001) afirma que esse padrão tem se mostrado ideal para dados estruturados hierarquicamente, como é o caso do esqueleto humano.

4.1 Utilização do XML

O XML pode ser disponibilizado de várias formas, em aplicações que utilizam-se de servidores e clientes ele pode ser implementado em ambos os lados. Dessa forma será analisada a estrutura dessa tecnologia, além dos benefícios que possibilitarão uma maior flexibilidade a esta aplicação.

O XML permite um nível de apresentação de dados flexível. Por exemplo, em dispositivos habilitados para Internet, como computadores de mão(PDAs⁵), ou telefones celulares, necessitam que as páginas sejam formatadas de maneira diferente da utilizada nos navegadores de Internet. Assim seria necessário criar uma versão totalmente diferente do site. Podem-se utilizar dados estruturados em formato XML, estes ficam separados da formatação, bastando aplicar aos dados estilos diferentes para as diferentes plataformas.

Segundo Brogden (2002), o XML está tendo maior impacto do lado do servidor. Um exemplo de aplicativo são os sistemas de mensagens, que consistem no intercâmbio de dados entre aplicativos ou computadores, ele ainda destaca que o XML não está ligado a nenhum

⁵ PDA *Personal Digital Assistants* – Assistentes Pessoais Digitais

sistema operacional ou fornecedor de *software*, e é compatível com uma grande quantidade de ferramentas e aplicativos.

Deitel(2003) apresenta um aplicativo chamado *XmlMessenger*, para troca instantânea de mensagens. Utiliza-se da arquitetura cliente-servidor, e está implementado na linguagem Java, a Figura 8 apresenta a arquitetura do *XmlMessenger*.

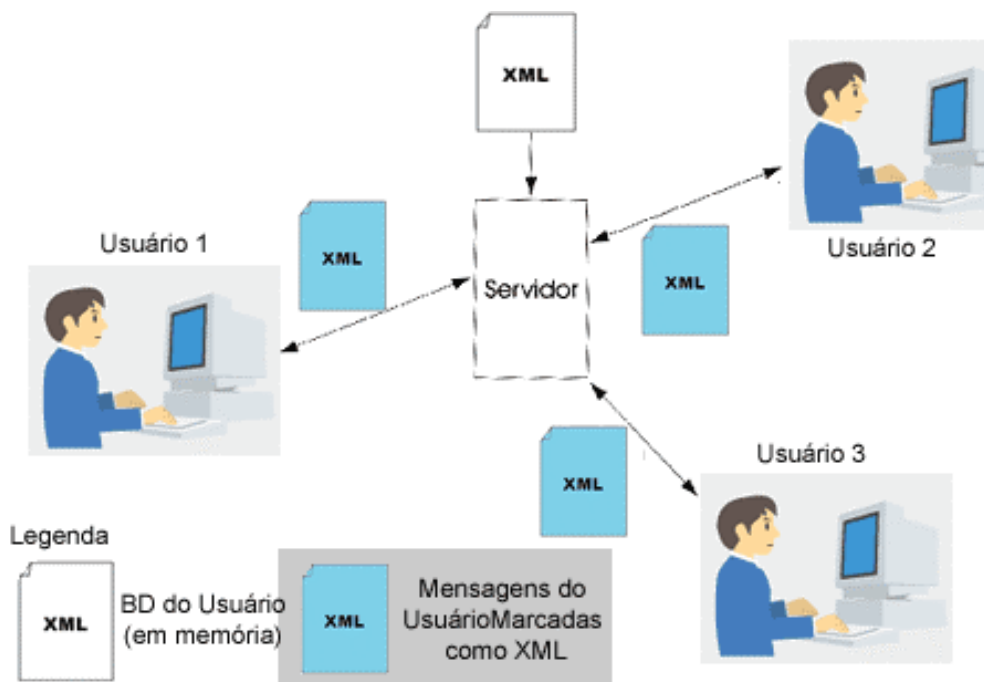


Figura 8: Arquitetura do *XmlMessenger* (Deitel, 2003)

O XML também pode ser utilizado para armazenamento de dados, embora ele não seja a maneira mais eficiente de armazenar dados, ele possui algumas vantagens, uma das mais importantes é a simplicidade, pois sua estrutura em árvore constitui uma maneira intuitiva e familiar de organização de dados, além disso, qualquer tipo de estrutura de dados pode ser representada em XML.

4.2 A sintaxe XML

Em um arquivo XML, os dados são gravados em formato de texto, com etiquetas de marcação para separar cada elemento, atributo ou objeto. Estas etiquetas também são

chamadas de *tags*. Portanto esses arquivos podem ser criados e modificados em qualquer editor de texto. Na Figura 9 um exemplo de documento no formato XML.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Laudo>
  <Paciente matricula="123">
    <Nome> João da Silva </Nome>
  </Paciente>
  <Conclusões>
    <LesãoArterial>
      <Grau> 40% </Grau>
      <Localização> 1/3 proximal
    </Localização>
      <Artéria> 2a. Marginal </Artéria>
    </LesãoArterial>
  </Conclusões>
```

Figura 9: Exemplo de Documento em formato XML

O XML é utilizado para gerar linguagens de marcação, como por exemplo um arquivo HTML⁶. O arquivo XML tem como característica marcante a possibilidade de extensão. Enquanto no HTML as tags são pré-fixadas, no XML elas podem ser criadas a vontade. Outra grande diferença, é que as tags do HTML especificam formatação e não conteúdo. Por exemplo, a tag <i> do HTML indica que o texto que virá a seguir deve ser apresentado em itálico. No XML, ao contrário, não existe informação sobre como a informação deverá ser apresentada, mas sim sobre o que a informação significa.

Toda tag XML é escrita entre os símbolos < e >. Quando o texto que compõe uma tag termina é acrescentado um indicador de fechamento, iniciando com o os símbolos </ . Por exemplo, a informação sobre o nome do paciente inicia com a tag <Nome> e termina com a tag </Nome>.

Um documento XML pode ser criado livremente, com tags quaisquer, mas na maioria dos casos, é muito importante ter um modelo que defina que tags podem ser usadas

⁶ HTML(Hyper Text Markup Language), linguagem de marcação utilizada para produzir páginas na Internet.

em um determinado tipo de documento e como elas podem ser utilizadas. Isto é feito através de um outro padrão criado pelo W3 Consortium chamado *Document Type Definition* (DTD).

NARDON (2002) comenta que uma DTD descreve a estrutura do documento e, uma vez associada a um documento XML, é utilizada para validá-lo, garantindo que as tags estão sendo utilizadas de forma correta. A criação de uma DTD é feita através da descrição dos elementos e atributos que podem aparecer no documento XML, informando também se os elementos podem se repetir, que valores possíveis um atributo pode assumir e que elementos e atributos são opcionais. Ela apresenta o seguinte exemplo de DTD para o documento XML apresentado como exemplo na Figura 10:

```

1. <?xml version="1.0" encoding="ISO-8859-1"?>
2. <IELEMENT Laudo (Paciente, Conclusões+)>
3. <IELEMENT Paciente (Nome)>
4. <IATTLIST Paciente matrícula CDATA #IMPLIED>
5. <IELEMENT Nome (#PCDATA)>
6. <IELEMENT Conclusões (LesãoArterial?)>
7. <IELEMENT LesãoArterial (Grau, Localização, Artéria)>
8. <IELEMENT Grau (#PCDATA)>
9. <IELEMENT Localização (#PCDATA)>
10. <IELEMENT Artéria (#PCDATA)>

```

Figura 10: Exemplo de DTD

A linha 1 indica que o documento é um documento XML e a seguir inicia-se a descrição dos elementos e atributos. O elemento Laudo apresentado na linha 2, por exemplo, é composto de um elemento Paciente, cuja presença é obrigatória, e de um ou mais elementos Conclusões. A possibilidade de haver um ou mais elementos Conclusões é indicado pelo sinal + colocado logo após o nome do elemento. O elemento Paciente indicado na linha 3, por sua vez, é composto de um elemento Nome e de um atributo matrícula. O atributo matrícula conterá um texto (que é o que a palavra CDATA representa) e será opcional (que é o que #IMPLIED representa). O elemento Nome será um texto qualquer e é isso que #PCDATA significa. O elemento Conclusões na linha 6, possui um elemento LesãoArterial, que é opcional. Esta opcionalidade é indicada pelo símbolo “?” colocado após o elemento.

Depois de criada a DTD, basta informar no documento XML que ela deve ser usada para validar o documento. Isto é feito acrescentando a linha `<!DOCTYPE Laudo SYSTEM "laudo.dtd">` logo após a linha `<?xml version="1.0" encoding="ISO-8859-1"?>` no documento XML. Neste exemplo, `laudo.dtd` é o nome do arquivo que contém a especificação da DTD.

Os DTDs têm algumas limitações, Fusco (2004) comenta que eles fornecem poucos controles sobre os elementos e seus relacionamentos, e para resolver estas limitações o W3C CONSORTIUM tem proposto o XML SCHEMA que pretende fornecer um mecanismo mais aprimorado de definir e validar documentos XML.

4.3 X3D

O X3D (*EXtensible 3D*) é um padrão aberto para distribuir conteúdo 3D. Ele não é uma API de programação, tão pouco um formato de arquivo para troca de geometrias. Ele combina ambos, geometria e descrições de comportamentos instantâneos em um simples arquivo que tem inúmeros formatos disponíveis para isso, incluindo o XML, (*Consortium Web3D*, 2006).

Segundo Marinho (2004) uma arquitetura modular foi empregada no X3D porque a maioria das aplicações não necessita de todos os seus recursos, tão pouco todas as plataformas suportam a gama de funcionalidades definidas na especificação. Com isso, dois novos conceitos foram adicionados ao X3D: *profiles* e *components*. Estes são as novas formas do X3D definir respectivamente extensibilidade e o conjunto de serviços que o conteúdo dos usuários necessita.

Os recursos do X3D são agrupados em componentes (*components*) que podem ser suportados pela implementação ou por uma plataforma em particular definindo uma coleção

específica de nós que possui um conjunto de funcionalidades. Um *profile* é uma coleção de *components* para um específico nível de suporte. Todos os arquivos X3D requerem a definição do *profile* que está em uso, na qual pode ser suprida com a requisição de componentes adicionais pelo usuário - ou por níveis maiores que aqueles providos pelo *profile*.

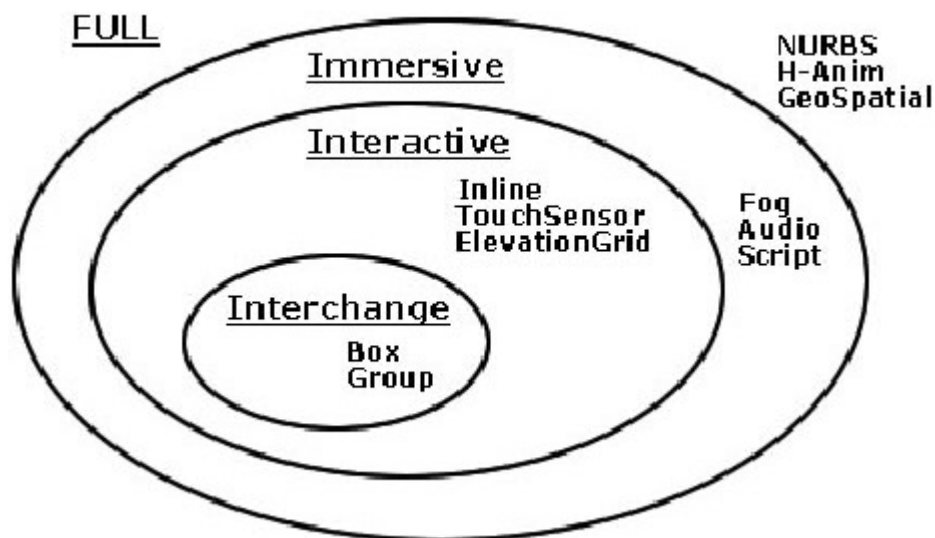


Figura 11: *Profile* definidos no X3D. (Consortium, 2006)

4.4 Document Object Model (DOM)

Consortium W3C(2006) caracteriza que o Modelo de Objeto de Documentos (DOM) é uma interface de programação de aplicativos (API) para documentos XML. É a definição da estrutura lógica dos documentos e o meio pelo qual um documento é acessado e manipulado.

Na especificação DOM, o termo "documento" é utilizado no seu sentido mais amplo - XML é usado como o meio de representação de muitos tipos diferentes de informação que podem ser armazenados em sistemas diversos e muitos seriam tradicionalmente considerados como informação no lugar de documentos. Assim mesmo, o XML apresenta estas

informações como documentos e o DOM pode ser utilizado para o gerenciamento destas informações.

Deitel et al. (2003) explica que um analisador baseado no DOM torna disponível uma biblioteca para uso de programas que permite que os dados em um documento XML sejam acessados e modificados pela manipulação dos nós de uma Árvore DOM.

Partindo deste estudo sobre a estruturação do XML, será apresentada a seguir a tecnologia JAVA, a qual pode propiciar formas de manipulação deste formato, e de outros que facilitem a integração desta ferramenta com outras aplicações.

5. JAVA

Neste capítulo será descrita de forma simplificada a linguagem JAVA, a qual servirá como ligação entre as aplicações, formatos e tecnologias a serem utilizados nos Ambientes Virtuais, além da manipulação dos Humanos Virtuais. Para a utilização e manipulação de ambientes e objetos tridimensionais será apresentada a API Java3D.

A linguagem Java foi desenvolvida pela *Sun Microsystems, Inc.* e utiliza-se dos conceitos de orientação a objetos, e é uma linguagem independente de plataforma, os programas em Java são geralmente divididos em classes, que por sua vez são compostas de métodos que retornam informações ao concluírem suas execuções. Segundo Deitel (2005), a grande vantagem de Java, é a rica coleção de classes existentes nas bibliotecas, que também são conhecidas como APIs. A API Java 3D é usada para desenvolvimento de programas Java que manipulam e mostram geometria tri-dimensional.

O modelo de programação da API Java 3D é baseado em grafos de cena. O universo virtual é composto por vários objetos hierarquicamente agrupados. Várias técnicas também foram desenvolvidas para tentar reproduzir a realidade em termos de aparência, que representam geometrias, luzes, formas, sons, comportamentos, fumaça, entre outros.

Manssour(2003) entende que para aproveitar melhor as funcionalidades do Java 3D, é importante conhecer alguns conceitos básicos de Computação Gráfica. Quando se trabalha com síntese de imagens 3D, deve-se considerar que a maioria dos periféricos de entrada e saída é 2D. Portanto, várias técnicas são utilizadas para contornar estas limitações dos dispositivos e possibilitar a visualização com realismo de um objeto modelado, por exemplo, através de uma malha de triângulos. Transformações geométricas de escala, rotação e translação, manipulação de uma câmera sintética, projeção perspectiva, iluminação, cor,

sombra e textura são algumas das técnicas utilizadas para contribuir na geração de imagens de alta qualidade.

Para que se desenvolvam mundos virtuais, o Java 3D exige que todos os objetos e componentes estejam em um universo virtual, é composto por grafos de cena, que consistem na organização dos objetos em estruturas hierárquicas (Sowizral, 1999), estrutura ideal para a representação de movimentos em Humanos Virtuais, conforme apresentado no item 3.1.2.

Um grafo de cena é criado a partir de instâncias de classes Java 3D que definem som, luz, orientação, geometria, localização e aparência visual dos objetos. Manssour(2003) explica que estes objetos correspondem aos nodos (ou vértices), e os seus relacionamentos são identificados por arcos (ou arestas) que representam dois tipos de relacionamentos: referência, que simplesmente associa um objeto com o grafo de cena; e herança (pai-filho), onde um “nodo do tipo grupo” pode ter um ou mais filhos, mas apenas um pai, e um “nodo do tipo folha” não pode ter filhos. Num grafo de cena, os nodos do tipo grupo são identificados por círculos, e do tipo folha por triângulos. A Figura 12 mostra a estrutura de um grafo de cena no Java 3D.

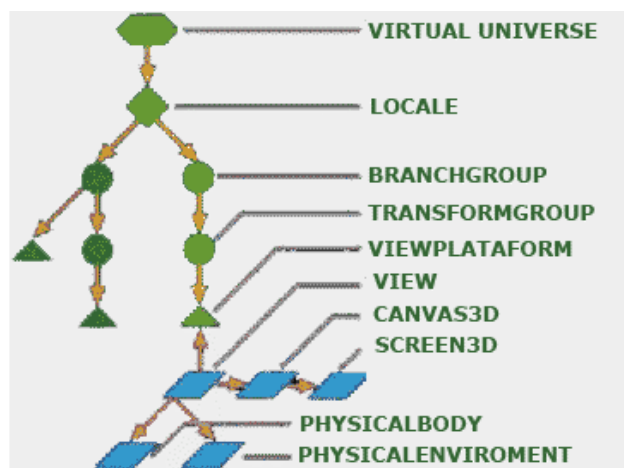


Figura 12: Representação Gráfica de um Grafo de Cena (Sowizral, 1999)

Como grande vantagem da utilização da API Java 3D, Rodello et al.(2003) comentam que ela oferece recursos para utilização de dispositivos de entrada, tanto

convencionais, como não convencionais. Outras vantagens seriam: a facilidade de implementação, o custo e a possibilidade de utilização de uma única linguagem em todo ciclo de um AV: da modelagem até a implementação de dispositivos.

Apresentados os recursos do Java e de sua API Java3D, a seguir será demonstrada a implementação do F-LIBRAS, utilizando-se dessas tecnologias.

6. DESCRIÇÃO DO F-LIBRAS

Os sistemas de RV proporcionam interações que apresentam uma interface muito próxima à realidade. O desenvolvimento de uma ferramenta que estimule o aprendizado e compreensão da língua de sinais seria apropriado, já que os sinais utilizam-se de gestos humanos. Estes seriam mais bem apresentados em 3D. Trata-se de um complemento tecnológico para a educação especial, que consiste em estimular através de imagens computadorizadas, o processo de comunicação e aprendizagem interativa.

Procura-se, desta forma, facilitar uma maior integração do deficiente auditivo na sociedade, respeitando a diversidade cultural e lingüística dos mesmos, bem como facilitar o acesso dos familiares, profissionais e pessoas que tenham interesse em conhecer a LIBRAS.

O F-LIBRAS é voltado para o domínio da Linguagem Brasileira de Sinais, destinado a dois tipos de usuários: a comunidade de deficientes auditivos e pessoas que interagem de algum modo com essa comunidade. Este ambiente deve proporcionar suporte para a integração de pessoas, sistemas computacionais e conteúdos. Em termos de usuários, incluem-se professores, alunos e monitores. As principais tarefas deste *software* seriam a gravação, editoração e visualização de gestos e movimentos da Língua de Sinais.

Para o desenvolvimento deste *software*, foi realizado um estudo sobre a linguagem de sinais e seus principais fatores. A partir disso, alguns recursos computacionais foram analisados, principalmente as características de Realidade Virtual, que auxiliariam no processo de desenvolvimento. Outro fator importante foi a caracterização de Humanos Virtuais, e os aspectos relativos à movimentação dos mesmos. Por fim foi analisada a linguagem XML para armazenamento, recuperação e transmissão dos dados, a fim de padronizar um formato onde os dados possam ser utilizados nos módulos deste sistema e em outros que possam ser desenvolvidos futuramente.

6.1 Interface do F-LIBRAS

A concepção do F-LIBRAS partiu do princípio de agrupar as funções de edição, interpretação de LIBRAS e a datilologia, em uma mesma interface. Para uma melhor compreensão, inicialmente será utilizada a UML, onde representa-se graficamente as funcionalidades através do Diagrama de Casos de Uso. O Caso de Uso é uma técnica de modelagem usada para descrever o que um sistema deve fazer. Ele demonstra as operações que cada usuário pode realizar no sistema. A Figura 13 apresenta o Diagrama de Casos de Uso do F-LIBRAS.

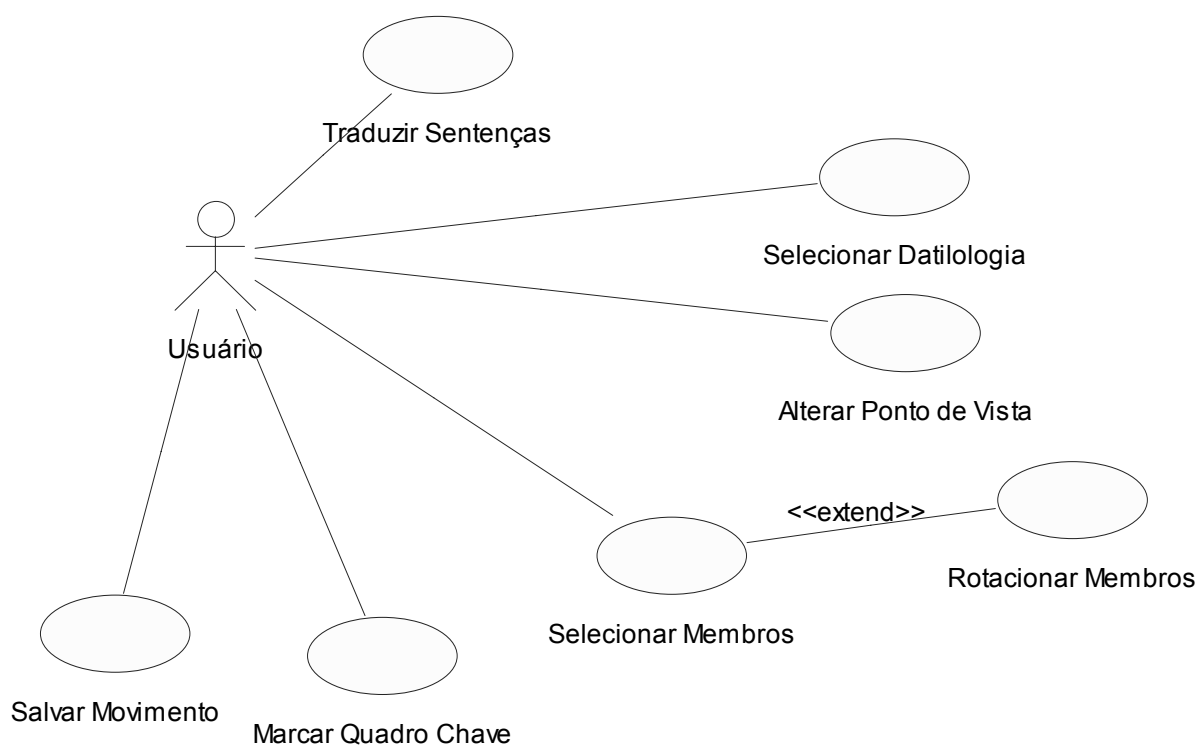


Figura 13: Diagrama de Casos de Uso do F-LIBRAS

Definidas e representadas as funcionalidades através do diagrama, existe a necessidade de demonstração de como acessar tais funções, para essas descrições a Figura 14 apresenta a interface inicial da ferramenta.

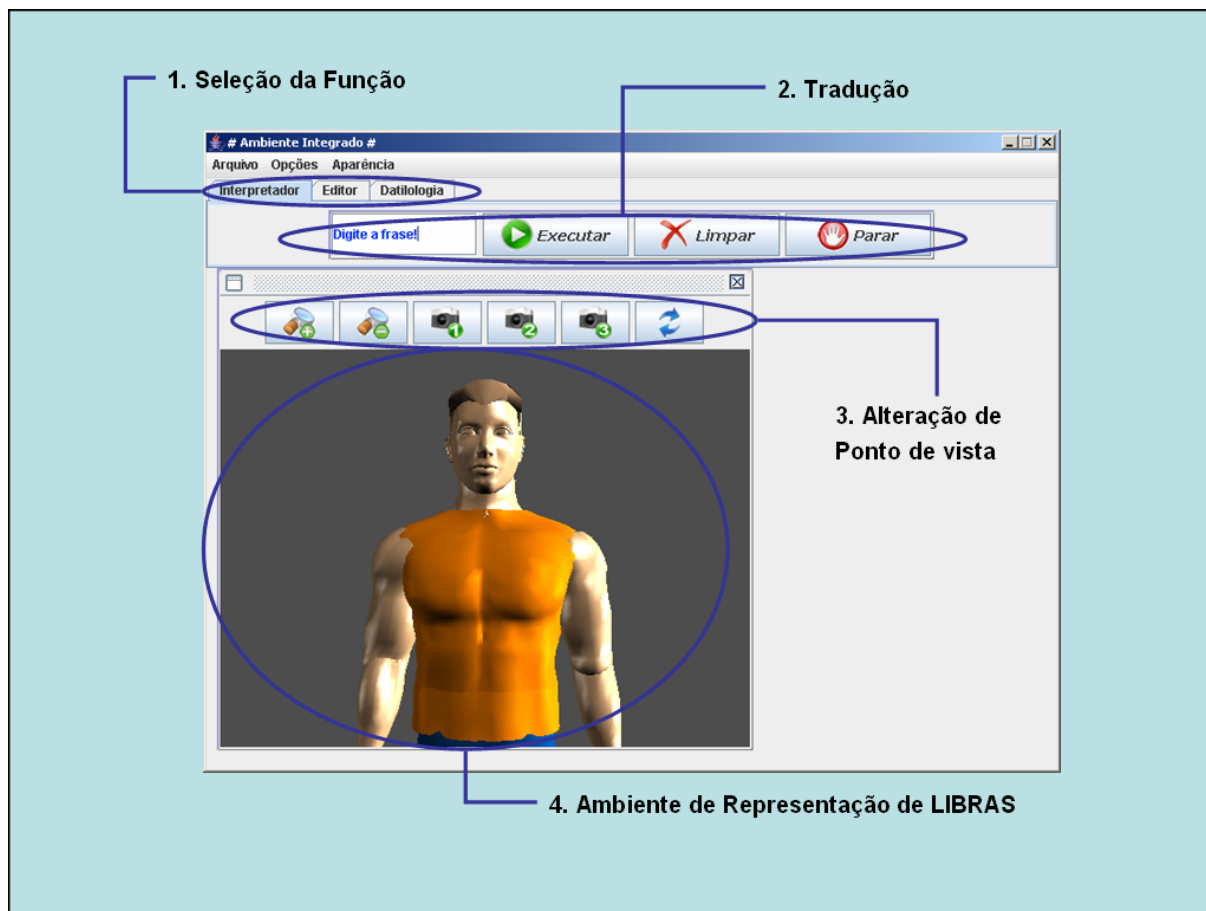


Figura 14: Interface Inicial do F-LIBRAS

A opção habilitada inicialmente é a de interpretação de sentenças em LIBRAS, a Tabela 3 descreve cada função mapeando os botões que possibilitam a utilização da ferramenta.

Tabela 3: Descrição da Interface do F-LIBRAS

Item	Descrição
1. Seleção de Função	Esta opção permite ao usuário alternar para as opções de edição e datilologia.
2. Tradução	Possibilita a digitação de uma sentença em LIBRAS, com as opções de executar, limpar e parar uma execução.
3. Alteração de Ponto de vista	Permite a escolha de três ângulos de ponto de vista, e também a possibilidade de se aproximar ou distanciar do Avatar.
4. Ambiente de Representação da LIBRAS	Ambiente Virtual responsável pela exibição dos movimentos em LIBRAS

Partindo das opções descritas, o interpretador tem a função de recuperar movimentos que estejam cadastrados no banco de dados específico da ferramenta. Dessa forma o interpretador funcionaria como um dicionário de palavras, com representação na linguagem gestual.

A tradução de LIBRAS inicia-se a partir da inserção de uma palavra ou sentença no local específico, descrito no item 2 da Tabela 3. Na seqüência é necessário selecionar o botão “Executar”, a partir daí as palavras são identificadas e classificadas de acordo com sua função e classe gramatical. A partir desta classificação são selecionados no banco de dados as palavras correspondentes e exibidos os movimentos associados a elas.

Os movimentos executados podem ser analisados e visualizados de diversas posições, para isto basta alterar o ponto de vista, e os posicionamentos de ponto de vista. Para executar estas mudanças, a ferramenta dispõe de uma barra de ferramentas acima do ambiente virtual, conforme o item 4 da Figura 14.

Conforme descrito anteriormente, o interpretador tem a função de buscar os movimentos em um banco de dados de gestos. A partir da necessidade de criar e armazenar novos gestos, a ferramenta dispõe do módulo denominado editor, que pode ser selecionado a partir do item 1 da Figura 14.

A interface do módulo e suas funções são visualizadas na Figura 15.

Para habilitar as opções de edição é necessário a seleção da opção 1. (Habilitação dos Controles de Edição) , dessa forma, tornam-se visíveis as barras de tarefas de edição, a Tabela 4 especifica cada função desta interface.

Cada item descrito possibilita que o usuário tenha o controle da geração do movimento. A seqüência apresentada na Tabela 4 informa quais passos seriam necessários para a criação e armazenamento de um movimento em LIBRAS.

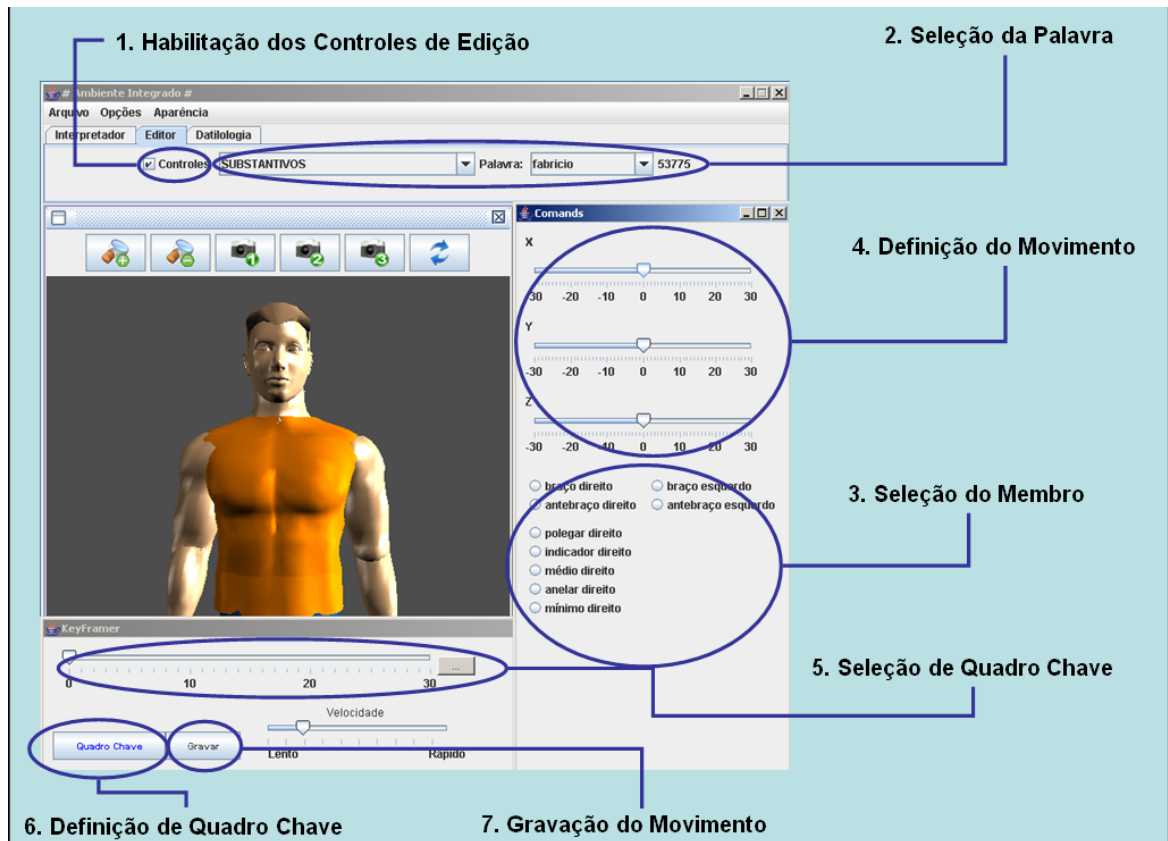


Figura 15: Interface do módulo Editor

Tabela 4: Descrição da Interface de Edição

Item	Descrição
1. Habilitação dos Controles de Edição	Habilitar as barras de tarefas de edição.
2. Seleção da Palavra	Identificação da categoria e da palavra para qual será criado o movimento.
3. Seleção do Membro	Escolha do membro do corpo a ser movimentado
4. Definição do Movimento	Definição do eixo e do ângulo de rotação do membro selecionado.
5. Seleção de Quadros-Chave	Atribuição do movimento ao quadro-chave correspondente.
6. Definição do Quadro-Chave	Gravação do quadro correspondente ao movimento.
7. Gravação do Movimento	Gravação final do movimento.

Inicialmente é necessário habilitar os controles do editor, e selecionar a palavra escolhida para geração do movimento. Assim que as barras de edição são exibidas deve ser selecionado o membro a ser movimentado, assim como o seu ângulo de rotação. Na Figura 16 estão destacados os componentes necessários para esta ação.

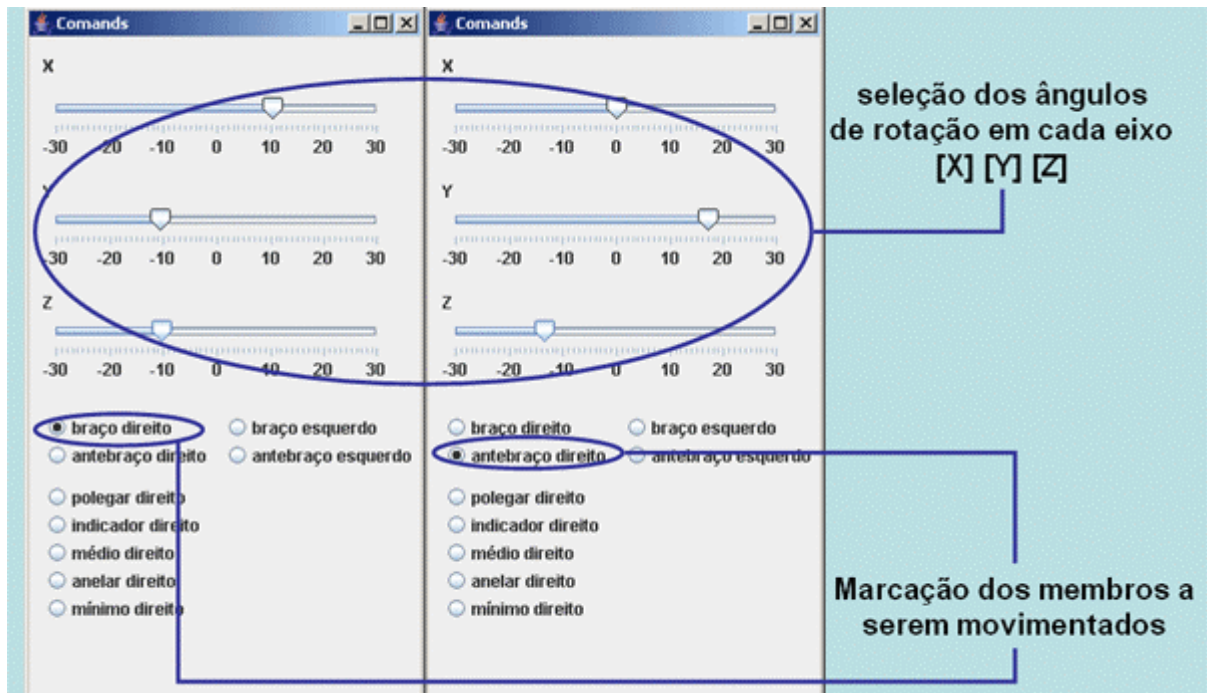


Figura 16: Definições de movimentos na barra de controles

Na seqüência deve ser realizada a definição e gravação do quadro-chave. Os quadros-chave são importantes na criação de movimentos, pois cada movimento de um membro do corpo deve ser marcado com o intuito de criar o movimento completo. A seleção e gravação dos quadros seguem até que todas as posições sejam definidas.

A Figura 17 exibe a exemplificação de um movimento composto por três posições, cada vez que é movimentado um membro deve ser definido um quadro, ou seja, a posição na animação em que o membro deve movimentar-se. No exemplo abaixo estão marcadas as posições 5, 10 e 15, como quadros-chave, isto significa que em cada uma destas posições, um membro sofreu uma alteração em sua posição.

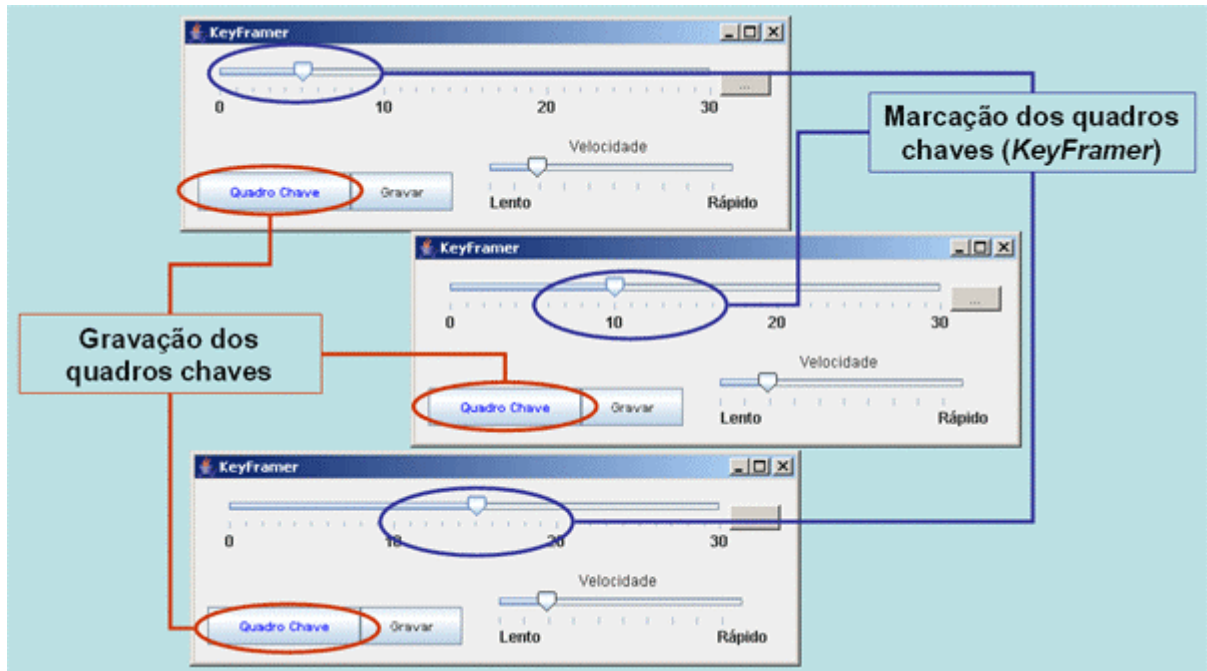


Figura 17: Opções do gerador de quadros (*KeyFramer*)

Finalizando todos os movimentos necessários, o último passo é a gravação do movimento, através da opção correspondente. No momento em que é gravado o movimento, três ações são geradas. A gravação da palavra é realizada automaticamente no banco de dados, e os arquivos XML e TXT são gerados respectivamente nas pastas “BaseDados\XML” e “BaseDados\TXT”, contidas no diretório de instalação da ferramenta, conforme descrito no ANEXO A.

Outra opção disponível na interface é a habilitação do Módulo de Datilologia, conforme descreve a Figura 18. No item 1 (1. Habilidade de Datilologia) identifica-se se o Módulo de Datilologia está habilitado, podendo-se definir a velocidade em que a mudança das letras vai ser executada. O item 2 (2. Representação de Datilologia), identifica o ambiente em que a datilologia será apresentada.

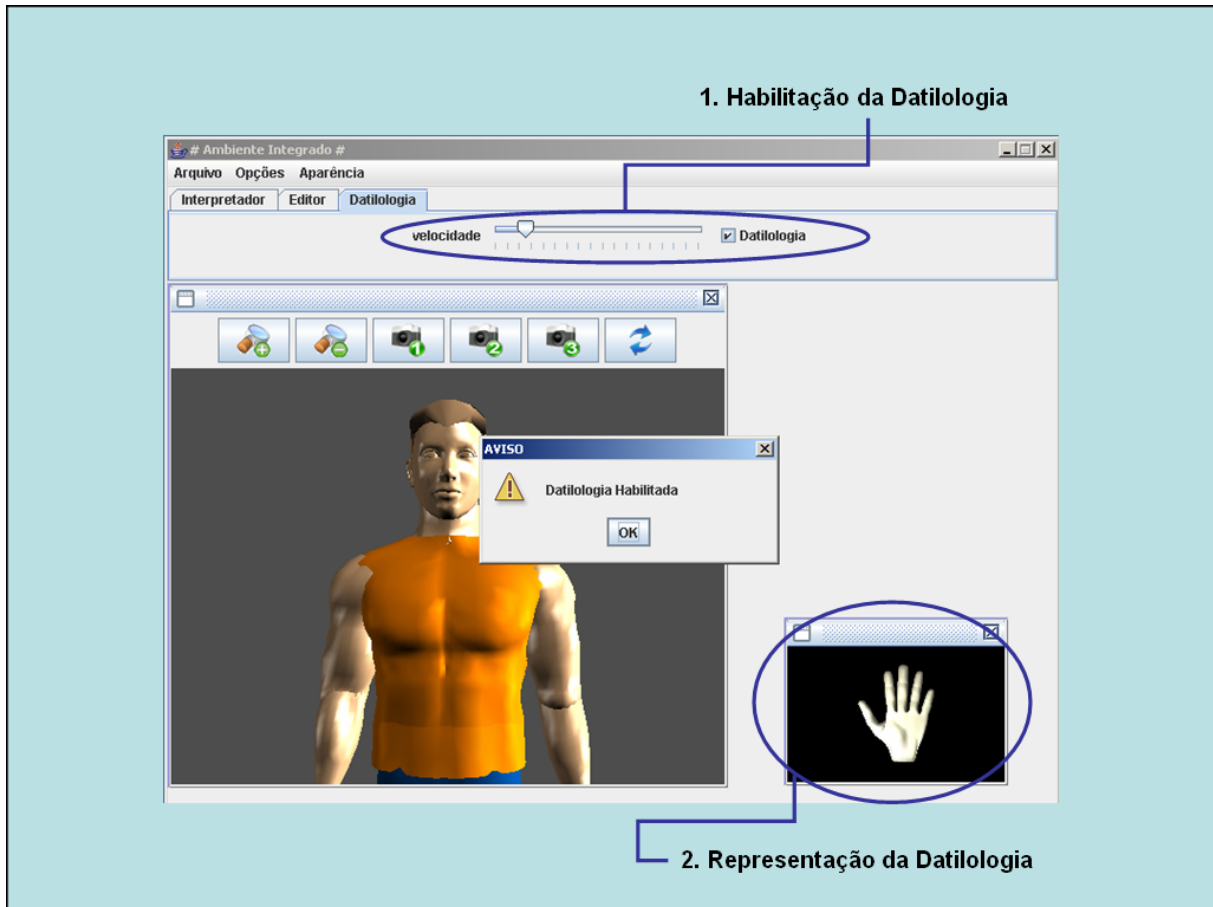


Figura 18: Configurações de Datilologia

Com base na descrição da interface e suas funcionalidades, a seguir será apresentada a estrutura da ferramenta, além das definições e descrições necessárias para sua complementação.

6.2 Desenvolvimento do F-LIBRAS

Pelo que foi apresentado nos capítulos anteriores, esta ferramenta pode proporcionar um caminho para facilitar a integração social através do aprendizado da Língua de Sinais - LIBRAS, proporcionando uma forma onde a comunidade com deficiência auditiva tenha acesso a recursos tecnológicos e que expandam a sua língua natural.

Esta seção tem o intuito de apresentar algumas das técnicas utilizadas durante o desenvolvimento, utilizando-se de diagramas, códigos e imagens. A intenção é ilustrar e demonstrar da melhor forma possível, a interligação entre módulos e classes que compõe o aplicativo.

6.2.1 Modularização do Ambiente

O desenvolvimento desta ferramenta foi dividido em módulos: Editor de Sinais, Interpretador de Sinais e Módulo de Datilologia, conforme descreve a Figura 19. Diversos fatores contribuíram para a concepção da ferramenta em módulos. Embora a idéia inicial seja de criar uma interface única para compartilhar todas as funções, esta divisão foi realizada para possibilitar que qualquer alteração realizada fique restrita apenas ao módulo correspondente.

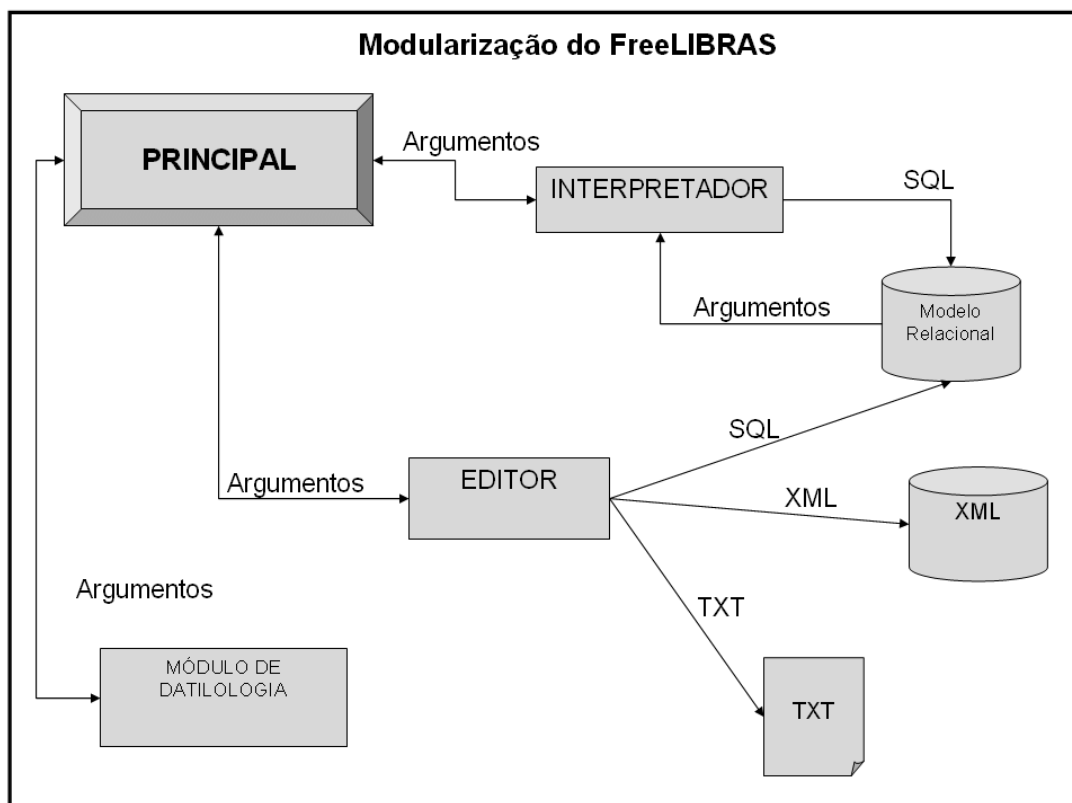


Figura 19: Modularização do Ambiente

Este formato proporciona que ferramentas independentes compartilhem um mesmo ambiente. Havendo necessidade de se utilizar algum dos módulos separadamente, é possível implementar somente as classes que o compõem, sem que todo o projeto seja incluído.

O Módulo Editor possibilita ao usuário, a geração e gravação dos sinais em LIBRAS, através de movimentos nos membros específicos do corpo humano, como: braços, antebraços, mãos, dedos, entre outros. A interface deste módulo foi apresentada na seção 6.1.

O Módulo Interpretador de Sinais tem como função receber requisições de movimentos, e caso existam no banco de dados, recuperá-los, e apresentá-los ao usuário. Estas requisições poderiam ser feitas em formato de frases da língua portuguesa. Cusin (2004) apresentou uma arquitetura de tratamento das frases separada em dois módulos: Módulo de Análise e Módulo de RV.

O Módulo de Análise é responsável por receber a sentença em LIBRAS, analisar e gerar as instruções de movimentos, essas instruções são enviadas para o Módulo de RV responsável pela execução dos movimentos.

O terceiro módulo é o de Datilologia, que é responsável pela representação das letras no alfabeto manual, ele é representado em um Ambiente Virtual próprio e age de forma independente dos módulos de edição e interpretação.

Apesar da descrição da independência das classes, formando a idéia de modularização, esses módulos compartilham de alguns pontos em comum entre si. Para o usuário da ferramenta esta modularização torna-se transparente, mantendo-se a impressão de um ambiente único. Dessa forma surge a necessidade dos módulos compartilharem algumas funções. A mais importante delas é o compartilhamento do Ambiente Virtual. A seguir são apresentados os Ambientes Virtuais que compõe o F-LIBRAS, e suas descrições.

6.2.2 Ambientes Virtuais do F-LIBRAS

Os módulos de edição, interpretação e datilologia agem diretamente sobre os Ambientes Virtuais. São esses ambientes que recebem os comandos e representam as instruções enviadas pelo usuário do aplicativo.

A modelagem de uma cena virtual é pré-requisito básico para a exibição do Avatar Humanóide onde serão executados os movimentos e também para a exibição da datilologia. Conforme especificado anteriormente, são utilizados dois AVs, um para a representação e edição dos gestos em LIBRAS e outro para a Datilologia.

Para implementação dos AVs inicialmente foi necessário definir o modelo de grafos, de cada componente, este modelo é utilizado para gerar e demonstrar as relações entre os nós do grafo (que representam os componentes responsáveis por posição, escala, cores e forma do objeto). A Figura 20 representa o grafo de cena do AV, onde são criados e representados os gestos em LIBRAS.

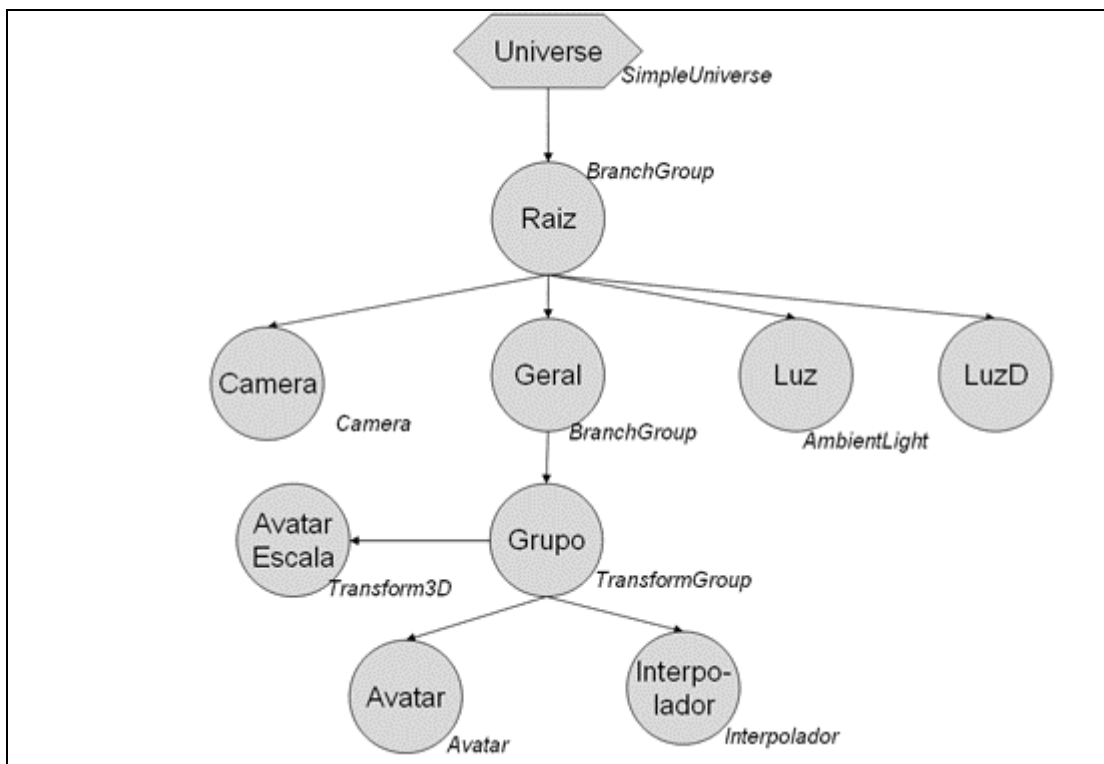


Figura 20: Grafo de cena do Ambiente

O grafo de cena apresenta resumidamente os componentes deste ambiente, é possível observar que a partir do nó *Raiz*, são ligados os componentes de iluminação (*Luz* e *LuzD*) e visualização (*Camera*).

Para inserção do *Avatar* no AV, é apresentado o nó *Geral* que possui como dependente um nó de transformação denominado *Grupo*, necessário para possibilitar a movimentação dos objetos. É neste em que estão agregados os objetos *Interpolador* e *Avatar*.

O objeto *Interpolador* é responsável por realizar todos os movimentos de interpretação e edição no Humano Virtual, que é representado pelo objeto *Avatar*.

O grafo de cena também é importante para explicar a estrutura hierárquica do objeto *Avatar*, que se divide em partes do corpo humano. Esta hierarquia determina qual componente do corpo pode ser movido sem que outras partes se movam junto (exemplo: dedo polegar) e também qual componente que se movido, ocasiona o movimento de todos os nós descendentes (exemplo: braço). Ele é formado por um conjunto de juntas representado pela junta raiz, a partir da qual pode-se obter toda a hierarquia. A partir do elemento *Tronco* são conectados os objetos que representam cada membro do corpo humano, de modo que se possa representar a hierarquia articular, cada membro possui uma referência para cada uma de suas juntas filhas. A Figura 21 apresenta o nó deste objeto, partindo do nó *Grupo* especificado na Figura 20.

Foram representados apenas os membros do lado direito do corpo (Braço e Perna), além de resumir o esquema da mão, com a representação da estrutura apenas do dedo indicador.

Por fim, a Figura 22 apresenta o grafo de cena do ambiente responsável pela representação da datilologia, em especial pode-se destacar o nó *Chave*, que é responsável por exibir o objetos correspondentes as letras do alfabeto manual.

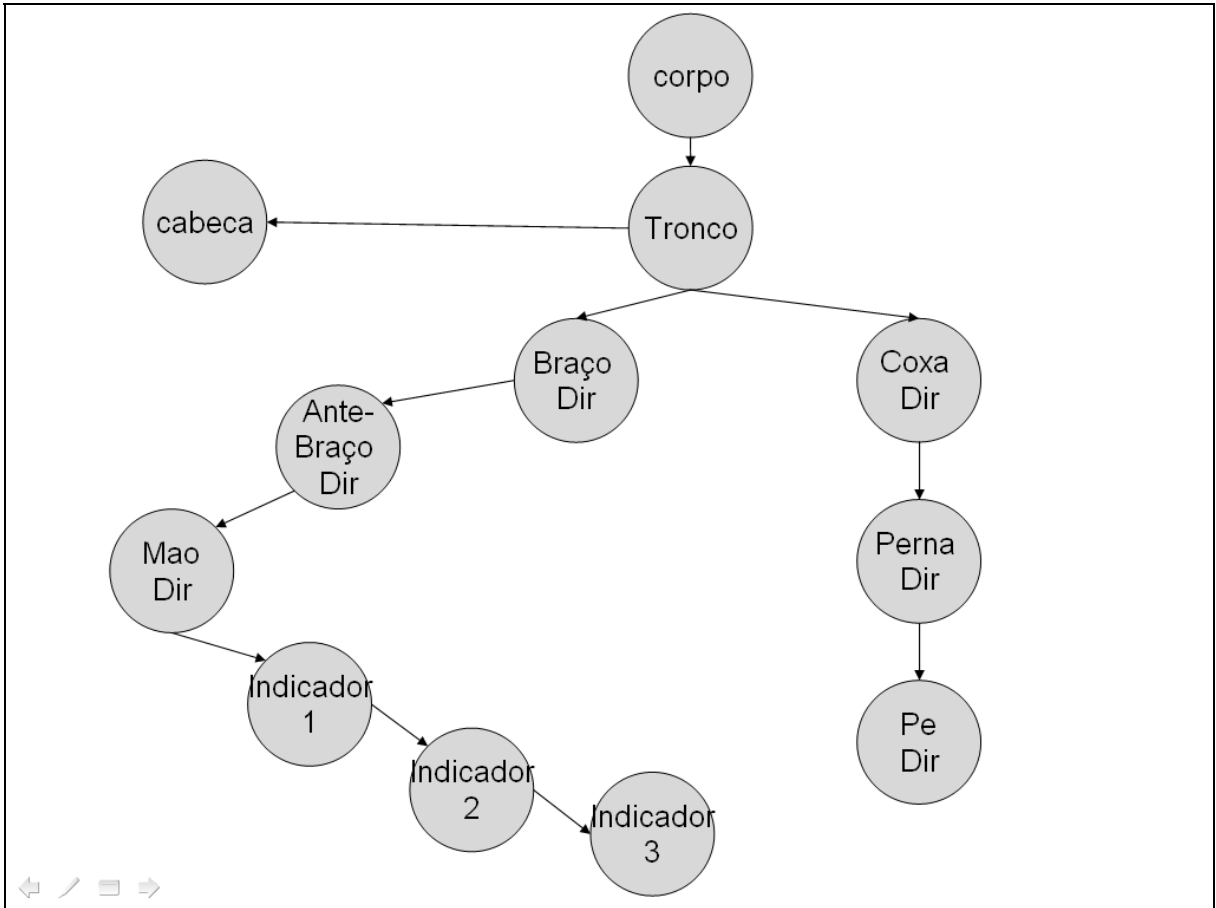


Figura 21: Nó Avatar

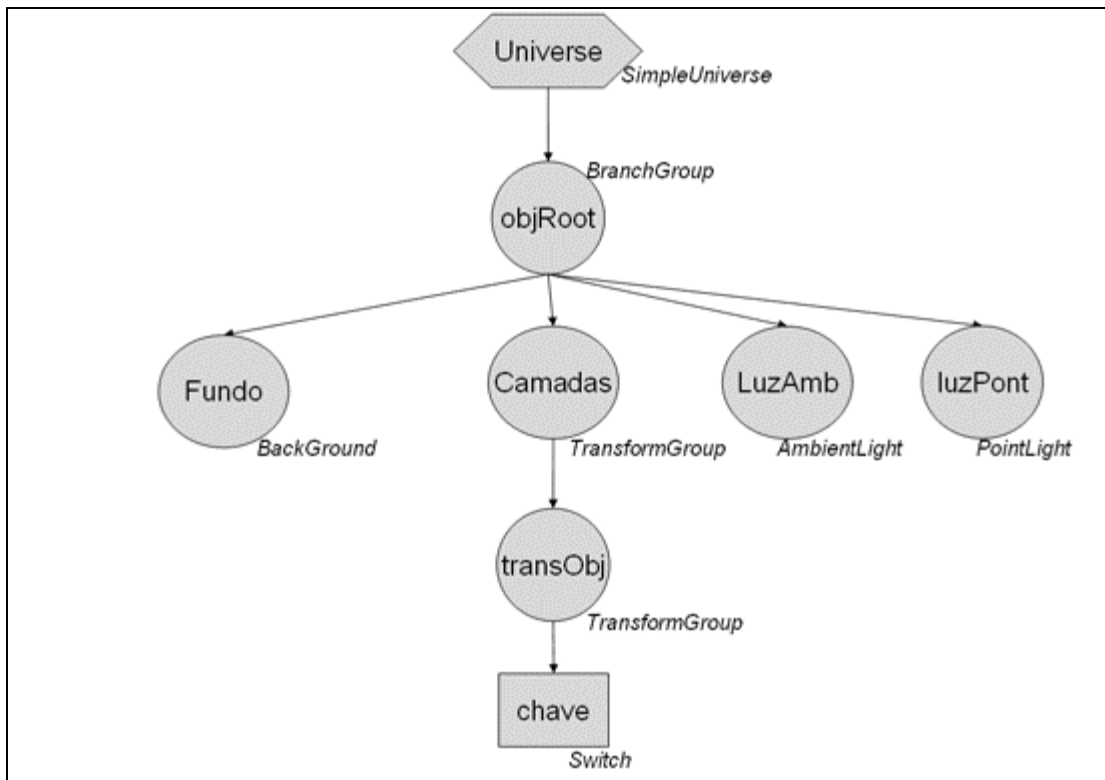


Figura 22: Grafo de Cena Datilologia

Com base nas descrições dos ambientes a partir dos grafos de cena, e da modularização da ferramenta especificada na seção anterior, pretende-se facilitar a apresentação da implementação. A seguir serão apresentadas as classes que compõem cada módulo, destacando as principais, e a forma que estão relacionadas. Tudo isto possibilita apresentar o desenvolvimento da arquitetura proposta, indicando os métodos e as técnicas utilizados para solucionar determinados problemas.

6.2.3 Classes do F-LIBRAS

A Figura 23 apresenta o diagrama de classes UML do F-LIBRAS que servirá como referência para todo este capítulo. A seguir são descritos, de forma geral, os principais pontos desta arquitetura.

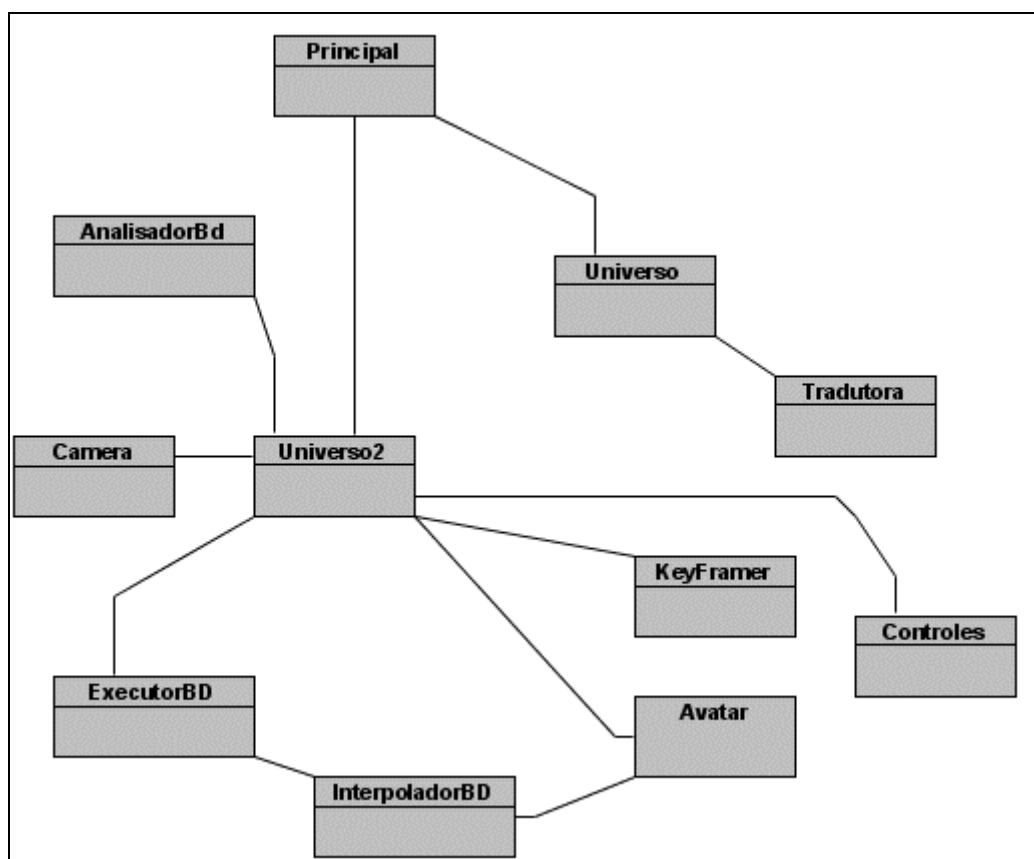


Figura 23: Diagrama de Classes do Aplicativo

A classe *Principal* é a responsável pela interface inicial do aplicativo fornece a interface gráfica para acesso às funcionalidades. A partir dela são instanciadas as classes necessárias no tratamento de eventos gerados pelo usuário. Nesta classe é implementada uma classe interna denominada *TraduzHandler*, que responde a eventos gerados pelo botão ‘Executa’ da interface com o usuário. Ela instancia os objetos da classe *Tradutora*, responsável pela datilografia (caso esta esteja habilitada), e também das classes responsáveis pela execução dos movimentos em LIBRAS. Antes do início de qualquer movimento é verificado se a sentença fornecida não está vazia, e também se não existe alguma execução em andamento. A classe *TraduzHandler* é definida dentro do corpo da classe *Principal* (Figura 24).

```

//de tradução (acionada pelo botão TRADUZ)
private class TraduzHandler implements java.awt.event.ActionListener {
    public void actionPerformed(java.awt.event.ActionEvent evento) {
        int erro = 0;
        String conteudo = jTextField1.getText();

        if (conteudo.equals("")) {
            erro = 1;
            javax.swing.JOptionPane.showMessageDialog(Principal.this, "Insira o texto", "AVISO",
            jTextField1.setText("Seu texto aqui!!!");
        }

        if (erro == 0)
        {
            Universo2.interpolador.editor = 0;
            jInternalFrameUniverso2.Executa( conteudo );
        }

        if (datiHab ==1 && erro == 0)
        {
            if (conteudo.equals("")) {
                javax.swing.JOptionPane.showMessageDialog(Principal.this, "Insira o texto", "AVISO",
                jTextField1.setText("Seu texto aqui!!!");
            } else {
                jInternalFrameUniverso.setVisible(true); //mostra form
                if (tradutora == null || !tradutora.isAlive()) {
                    jTextArea1.setText("Traduzindo...");
                    tradutora = new Tradutora(jTextField1.getText(),
                    jInternalFrameUniverso, jTextArea1, velocidade );
                    tradutora.start();
                } else {
                    javax.swing.JOptionPane
                    .showMessageDialog(Principal.this, "Já existe uma tradução em progresso",
                    javax.swing.JOptionPane.WARNING_MESSAGE);
                }
            }
        }
    }
}

```

Figura 24: Classe *TraduzHandler*

Outra classe interna definida no corpo da *Principal* é a classe *PararHandler*, a qual é acionada a partir do botão ‘Parar’ da interface, tem a função de interromper qualquer execução em andamento.

6.2.3.1 Classe InterpoladorBD

Uma vez que se tenha definido e implementado um modelo de representação de corpos, a necessidade de aplicar movimentos sobre este corpo acaba surgindo naturalmente. Desta forma faz-se necessária a aplicação de movimentos controlados sobre ele.

Para que os módulos que necessitam implementar movimentos do ambiente possam agir sobre o Avatar, foi desenvolvida uma classe para realizar esse trabalho. A classe *InterpoladorBD* permite associar um conjunto de movimentos a um corpo e controlar a execução desses movimentos. Portanto, um objeto desta classe tem associado a ele, um corpo instanciado pela classe *Avatar* e os parâmetros necessários da animação. A classe *InterpoladorBD* representa cada uma das ordens individuais de movimentação, onde são indicados os membros do corpo, os eixos e ângulos de rotação.

Nesta classe são definidos dois objetos de transformação, ou *Transform3D* é um elemento auxiliar que, juntamente com um nó de transformação, irá aplicar sobre um conjunto de nós filhos uma operação de transformação geométrica: uma translação, rotação, redimensionamento ou uma composição destes.

Uma interação no ambiente é uma atualização do objeto de transformação de um nó de transformação pai. Os filhos deste nó de transformação irão obedecer à transformação. A Figura 25 apresenta um exemplo desta implementação na classe.

```

/*Rotação Braço direito*****
public void rotBracoD(int eixo, float angulo)
{
    if (eixo == 0)
    {
        parteBracoD.getTransform(t1);
        t2.setIdentity();
        t2.rotX(-(float)Math.toRadians(angulo));
        t1.mul(t2);
        parteBracoD.setTransform(t1);
    }
    else if (eixo == 1)
    {
        parteBracoD.getTransform(t1);
        t2.setIdentity();
        t2.rotY(-(float)Math.toRadians(angulo));
        t1.mul(t2);
        parteBracoD.setTransform(t1);
    }
    else if (eixo == 2)
    {
        parteBracoD.getTransform(t1);
        t2.setIdentity();
        t2.rotZ(-(float)Math.toRadians(angulo));
        t1.mul(t2);
        parteBracoD.setTransform(t1);
    }
}

```

Figura 25: Implementação de Movimentação

Com base nos métodos da classe *InterpoladorBD*, é que os módulos de interpretação e de edição agem sobre o a classe *Avatar*, a fim de realizar a criação e execução dos movimentos desejados.

6.2.3.2 Classe Universo e Universo2

Para a apresentação dos Ambientes Virtuais de movimento e datilologia são instanciadas duas classes, são elas *Universo* e *Universo2*, as quais são subclasses de *JInternalFrame* que possibilita que dois Ambientes Virtuais possam estar contidos na interface principal do aplicativo.

Na classe *Universo* é instanciado o *SimpleUniverse*, onde as mãos em datilologia são exibidas, nesta classe também são definidos todos os componentes do ambiente, tais como, iluminação e posicionamento.

A classe *Universo2* é responsável pela criação do ambiente onde agem os módulos de edição e interpretação da LIBRAS. Nesta implementação, a cena (ou modelo) é uma instância da classe. Para formação da cena são utilizados objetos gráficos, um conjunto de luzes e um conjunto de câmeras virtuais.

Os atributos de cada fonte de luz definem a forma como ela influencia na iluminação de todos os objetos da cena. Sejam quais forem, as fontes de luz têm em comum o atributo intensidade, que é dado pelas três componentes de cor, R, G e B, variando de 0 a 1.

A luz direcional é aquela que se encontra posicionada em uma determinada direção no infinito, fazendo com que seus raios sejam todos paralelos. As cores e a direção em que ela incide são implementadas nesta classe e os parâmetros de iluminação são adicionados ao grafo de cena.

As câmeras virtuais são usadas para simular o ponto de vista de um observador dentro de uma cena virtual, e obter imagens correspondentes à sua visão. Para isso, foi implementada a classe *Camera*. Esta classe armazena os parâmetros de diversas câmeras, com ângulos e posições variadas. As mudanças de posicionamento e ponto de vista são implementadas em métodos da própria classe *Camera*, essas funções são invocadas a partir dos botões de controle contidos na parte superior da janela conforme mostra a Figura 26, e implementadas através dos métodos *mousePressed()* e *mouseReleased()* da classe *MouseListener* contida na classe *Universo2*.



Figura 26: Controle de ponto de vista

A classe *Universo2* instancia a classe *Avatar*, responsável por implementar a representação humana. É importante ressaltar que os objetos geométricos associados a um corpo, antes de tudo, precisam fazer parte de uma cena; as classes que representam cada membro possuem apenas referências para objetos da cena. Cada uma dessas classes contém as propriedades e os métodos necessários para movimentá-las.

6.2.3.3 Classes do Módulo Editor

O Módulo Editor possui três principais classes que implementam as funcionalidades necessárias para a criação e gravação dos movimentos, são elas *Controles*, *KeyFramer* e *GravaMovimento*.

A classe *Controles* é a responsável pela exibição desta interface, utilizando-se do componente *JSlider* para cada eixo (X, Y e Z) é definido o ângulo de movimento. Para seleção do membro a ser movimentado, é utilizado o componente *JRadioButton*, limitando a seleção de um membro de cada vez, a Figura 27 apresenta esses componentes.

A partir do momento em que é selecionado o membro a ser movimentado, são escolhidos os ângulos desejados, a cada alteração da movimentação de qualquer um dos eixos, são acionadas as classes *Interpolador* e *KeyFramer* conforme a apresenta a Figura 28.

A classe *Interpolador* está especificada no item 6.2.3.1, ficando a cargo da classe *Controle* apenas indicar o membro, ângulo e eixo de movimento. Ao mesmo tempo são enviados os ângulos de rotação para os métodos *passagemX()*, *passagemY()* e *passagemZ()*, para posterior utilização no armazenamento dos dados.

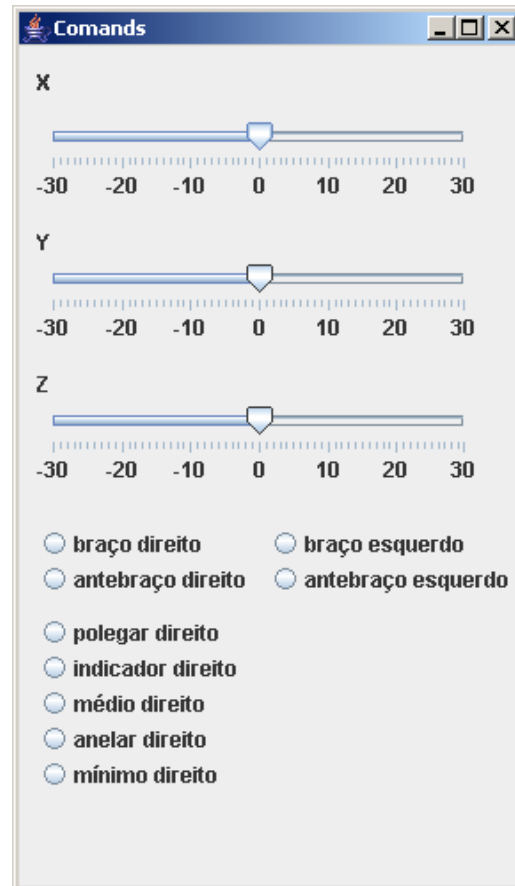


Figura 27: Interface do Módulo de Controles

```

if(btnBracoDireito.isSelected()== true)
{
    float auxX;
    float auxY;
    float auxZ;

    if(eixoX.getValueIsAdjusting()==true)
    {
        auxY=0.0f;
        auxZ=0.0f;
        auxX =(float) eixoX.getValue();
        valorbracodireitoX=eixoX.getValue();
        Universo2.interpolador.rotBracoD(0,auxX);
        Universo2.interpolador.rotBracoD(1,auxY);
        Universo2.interpolador.rotBracoD(2,auxZ);
        -----
        keyFramer.passagemX(auxX);
        keyFramer.passagemY(auxY);
        keyFramer.passagemZ(auxZ);
    }
}

```

Figura 28: Ações da Classe *Controles*

A partir da classe *KeyFramer* são definidos os quadros-chave (posição final), e a velocidade do movimento. O movimento completo pode possuir um ou vários quadros-chave,

onde cada quadro deve representar uma parte do movimento. Para definição das posições de cada movimento (quadros-chave) e velocidade também é utilizado o componente *JSlider*, conforme pode ser observado na Figura 29.

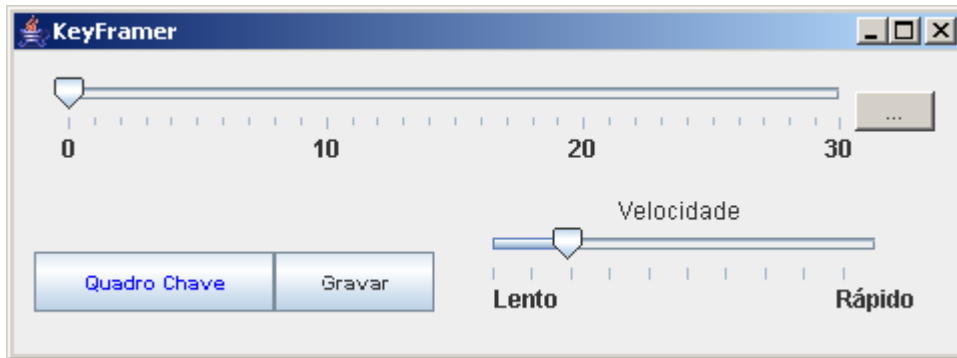


Figura 29: Gerador de Movimentos

Durante uma simulação, todos os parâmetros e estruturas que definem o movimento são mantidos na memória RAM, porém, a informação armazenada nessa memória deixa de existir quando o programa termina de executar. Para que as informações do movimento possam ser reutilizadas é necessário armazenar todos os dados referentes ao mesmo, como: posição dos membros, ângulos de rotação e quadros-chave. Estes dados são enviados para a classe *GravaMovimento* que é responsável pela padronização de armazenamento, possibilitando a recuperação do mesmo. Para uma posterior reutilização é necessário que, no momento de gravação, seja definida a palavra e a categoria a que se refere, a fim de se evitar que o movimento seja utilizado fora de contexto.

A descrição do movimento precisa ser gravada em um formato tal que possa ser utilizada no interpretador, e também ser importada para outros ambientes de animação e modelagem desenvolvidos futuramente. A solução proposta para este problema foi armazenar os movimentos em três formatos diferentes, arquivos padrão XML, arquivos texto e banco de dados relacional.

Para definir a arquitetura do documento XML, utilizou-se o esquema apresentado na Figura 30. É um modelo para descrever a estrutura da informação para toda uma classe de documentos.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <palavra>
  <categoria>ESTADO/CONDIÇÃO</categoria>
  <classegram>SUBSTANTIVO</classegram>
  <dspalavra>AMIGO</dspalavra>
- <movimento>
  - <membro>
    <dsmembro>rotBracoD</dsmembro>
    <eixox>15</eixox>
    <eixoy>00</eixoy>
    <eixoz>00</eixoz>
  </membro>
  - <membro>
    <dsmembro>rotAntebracoD</dsmembro>
    <eixox>12</eixox>
    <eixoy>00</eixoy>
    <eixoz>12</eixoz>
  </membro>
  + <membro>
  + <membro>
  + <membro>
  + <membro>
  </movimento>
+ <movimento>
+ <movimento>
+ <movimento>
</palavra>

```

Figura 30: Estrutura do arquivo XML

O objeto principal ou raiz é o nó ‘palavra’, que é composto pela categoria, classe gramatical, descrição da palavra e os movimentos. Cada movimento possui a descrição dos membros a serem movimentados e também os ângulos de rotação nos eixos X, Y e Z.

Para gravação do arquivo em formato texto, manteve-se o esquema utilizado inicialmente no módulo editor de sinais, que serviu como base para criação deste ambiente. O arquivo texto recebe instruções dos ângulos de rotação dos membros. Estas instruções estão delimitadas por ponto e vírgula (;), e cada linha do arquivo corresponde a posição de todos os membros em cada eixo (X, Y e Z), um exemplo deste formato está exposto na Figura 31.

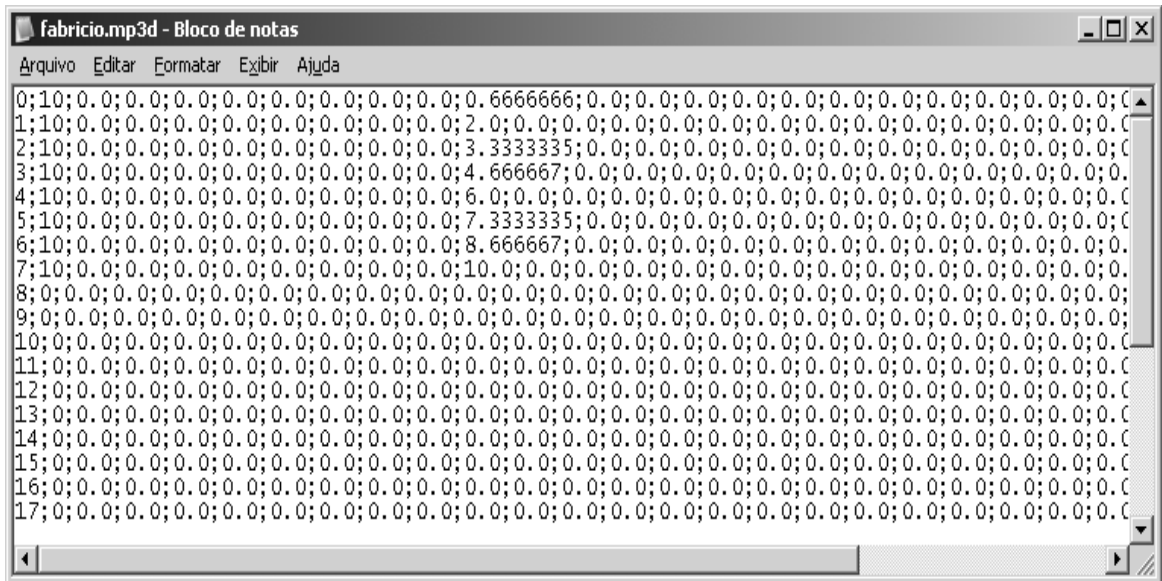


Figura 31: Exemplo de armazenamento em arquivo texto

A concepção deste arquivo tem o intuito de facilitar a integração de captação de movimentos através de um rastreador de movimentos, já que grande parte utiliza os arquivos texto para armazenamento dos mesmos. Partindo deste arquivo seria possível a edição e a criação dos movimentos captados por estes dispositivos.

Nesta ferramenta o modelo de banco de dados relacional serve como base para o interpretador das sentenças. Atualmente o banco de dados *Access* está sendo utilizado, mas como a linguagem utilizada para as consultas e gravações está no padrão SQL (*Structured Query Language*), a mudança pode ser feita sem complicações para qualquer outro banco de dados compatível com esta estrutura. A Figura 32 demonstra um exemplo de armazenamento de movimentos utilizando esta tecnologia.

movimento : Tabela						
idMovimento	copiaPalavra	indice	membro	eixo	angulo	
88	vi	5	14	2	-15,0	
89	vi	6	11	2	-15,0	
90	vi	7	15	2	-10,0	
91	vi	8	12	2	-10,0	
92	vi	9	16	2	-5,0	
93	vi	10	13	2	-5,0	
94	vi	11	17	2	-10,0	
95	vi	12	17	0	-15,0	
96	vi	13	18	0	-15,0	
97	vi	14	1	2	-2,0	
98	vi	15	1	0	10,0	
99	vi	16	2	2	-10,0	
100	vi	17	2	2	-7,5	
101	vi	18	1	0	-10,0	
103	fabricao	1	1	0	-15,0	
104	fabricao	2	1	2	25,0	

Registro: 1 de 103

Figura 32: Exemplo de Armazenamento em banco de dados relacional

A estrutura do banco de dados possui diversas tabelas que definem a classe gramatical da palavra, categorias, sinônimos e os movimentos correspondentes a cada palavra, a Figura 33 demonstra o diagrama de entidade relacionamento.

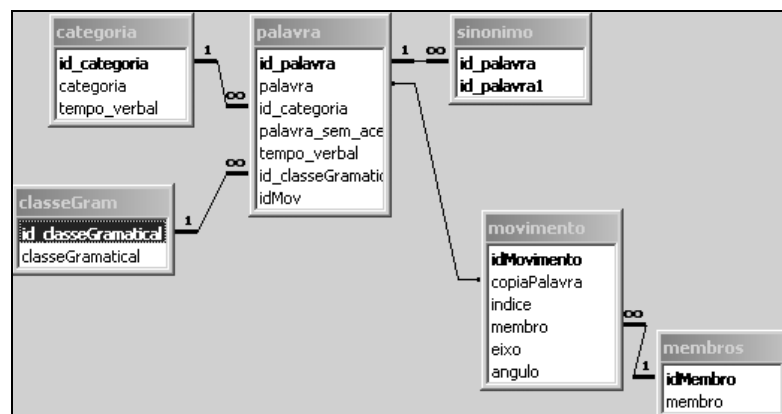


Figura 33: Diagrama Entidade Relacionamento

Apresentados os três formatos de armazenamento de dados, a classe *GravaMovimento* possui três métodos distintos que são invocados a partir da classe *KeyFramer*, são eles *GravaXml()*, *GravaTxt()* e *GravaBd()*, que realizam a gravação em XML, texto e no banco de dados relacional respectivamente.

6.2.3.4 Classes do Módulo Interpretador

As classes utilizadas na interpretação dos movimentos, são invocadas a partir da classe *Principal*, conforme já especificado, e são responsáveis por realizar o tratamento das sentenças.

A classe *AnalizadorBD* tem a função de realizar o tratamento das frases, dividindo-as em palavras. Esta delimitação é feita através do reconhecimento de espaços em branco e vírgulas. Possui os métodos *montaVetorInicial()*, que tira sinais de pontuação e espaços, e também *leGramatica()* que classifica cada palavra em verbos, substantivos, artigos entre outras classes gramaticais.

Após a análise anterior é necessária a chamada de outra classe *ExecutorBD*, que verifica o vetor de palavras na classe *AnalizadorBD*. Em seguida é realizada a busca no banco de dados de todos os movimentos correspondentes. Os movimentos são executados a partir do método *Executa()* da classe *InterpoladorBD*, especificada na subseção 6.2.3.1. A Figura 34 apresenta a sintaxe da chamada do método.

```

while ( i != (palavra[index]).length() )
{
    String pesq="SELECT movLetra.membro, movLetra.eixo, movLetra.angulo FROM movLetra INNER JOIN
    System.out.println("\nVai pegar os movimentos da letra " +palavra[index].charAt(i));

    Statement statement = conexao.createStatement();
    ResultSet resultado = statement.executeQuery ( pesq );

    System.out.println ("\nMontou a query de movimentos de letras!\n");

    while (resultado.next())
    {
        membro = resultado.getInt("membro");
        eixo = resultado.getInt("eixo");
        angulo = resultado.getFloat("angulo");
        interpolador.executar(membro, eixo, angulo);
    } // fim do while

    i++;
    statement.close();
    interpolador.executar(0,0,0);
}

```

Figura 34: Implementação da execução dos movimentos

6.2.3.5 Classes do Módulo de Datilologia

O módulo de Datilologia é composto da classe *Tradutora*, que envia os comandos para a classe *Universo*, montando a seqüência de execuções em que as letras em datilologia devem ser exibidas.

Para a apresentação das letras em datilologia é utilizado o nó *Switch* (*javax.media.j3d.Switch*). Este nó controla o que deve ser renderizado em determinado momento. Os objetos renderizados(mãos) são carregados em tempo de execução utilizando o método *load* do *ObjectFile*. A Figura 35 mostra a criação de um objeto *ObjectFile*.

```
ObjectFile obj =
    new ObjectFile(ObjectFile.RESIZE, (float)(10.0 * Math.PI / 180.0));
try {
    a = obj.load("hands\\A.obj");
    b = obj.load("hands\\B.obj");
    c = obj.load("hands\\C.obj");
    d = obj.load("hands\\D.obj");
    e = obj.load("hands\\E.obj");
    f = obj.load("hands\\F.obj");
    g = obj.load("hands\\G.obj");
    h = obj.load("hands\\H.obj");
    i = obj.load("hands\\I.obj");
    j = obj.load("hands\\J.obj");
    k = obj.load("hands\\K.obj");
    l = obj.load("hands\\L.obj");
    m = obj.load("hands\\M.obj");
    n = obj.load("hands\\N.obj");
}
```

Figura 35: Criação do *ObjectFile*

Após apresentado o modelo de implementação, relacionando as classes que compõem o sistema, será apresentada na próxima seção, a criação e execução de alguns movimentos em LIBRAS.

6.3 Resultados

Este capítulo exemplifica a execução de todos os módulos do sistema. Inicialmente a definição da configuração de *hardware* e *software* utilizados nessas simulações torna-se necessária, visto que será apresentado o levantamento de quadros por segundo apresentados

durante cada simulação. Será analisada a taxa de quadros por segundo, e se esta taxa é aceitável para conseguir obter uma animação e apresentação conveniente do gesto em questão.

O computador utilizado no desenvolvimento e também na simulação dos movimentos é um PC (*Personal Computer*), com processador 3.2GHZ, 512MB RAM, HD 40 GB, com placa de vídeo integrada 128 MB de memória e sistema operacional *Windows XP*.

Alguns movimentos foram criados para a simulação da ferramenta. Inicialmente será apresentada a geração e gravação dos gestos em LIBRAS. Para isso pode-se definir como roteiro de animação, uma seqüência de quadros que mostra parte da animação de forma simplificada.

A seguir será demonstrada a seqüência de quadros utilizados na criação do movimento da palavra “Governador”. A mesma foi escolhida por não possuir um número tão grande de quadros, o que possibilita que o mesmo movimento seja exemplificado em diversos pontos de vista distintos.

Para geração deste movimento foram geradas três posições e definidos três quadros-chave conforme apresentado na Figura 36 .



Figura 36: Exemplificação de movimento, quadros-chave principais

Inicialmente o braço direito foi movido 10° no eixo X e marcado no quadro-chave 2 na Figura 37 pode se verificar este movimento e o detalhamento das posições.

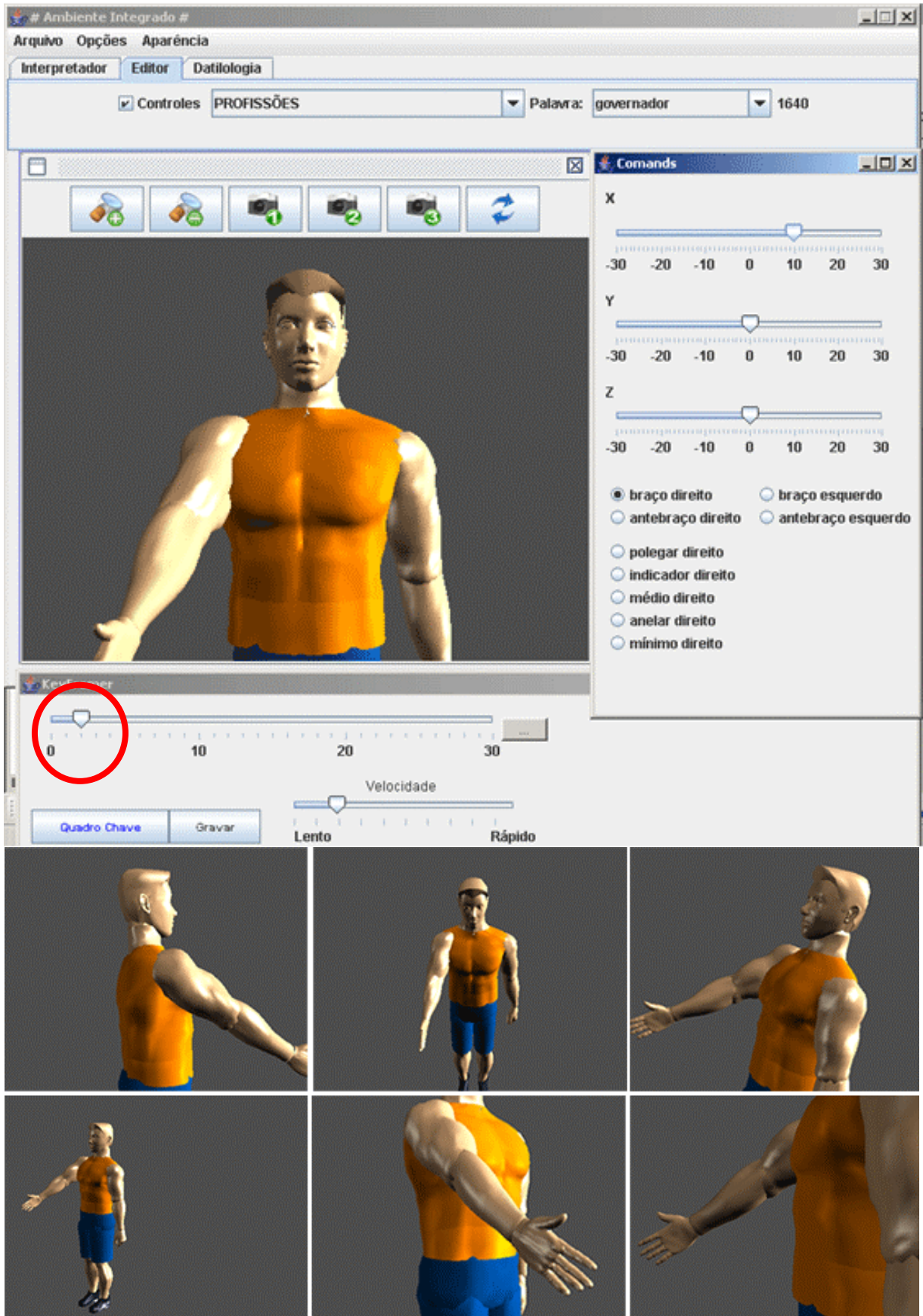


Figura 37: Detalhamento do Movimento: Cena 1

Partindo dessa posição o antebraço direito foi movido no eixo Z em -10° e marcado no quadro-chave 5 conforme detalhado na Figura 38.

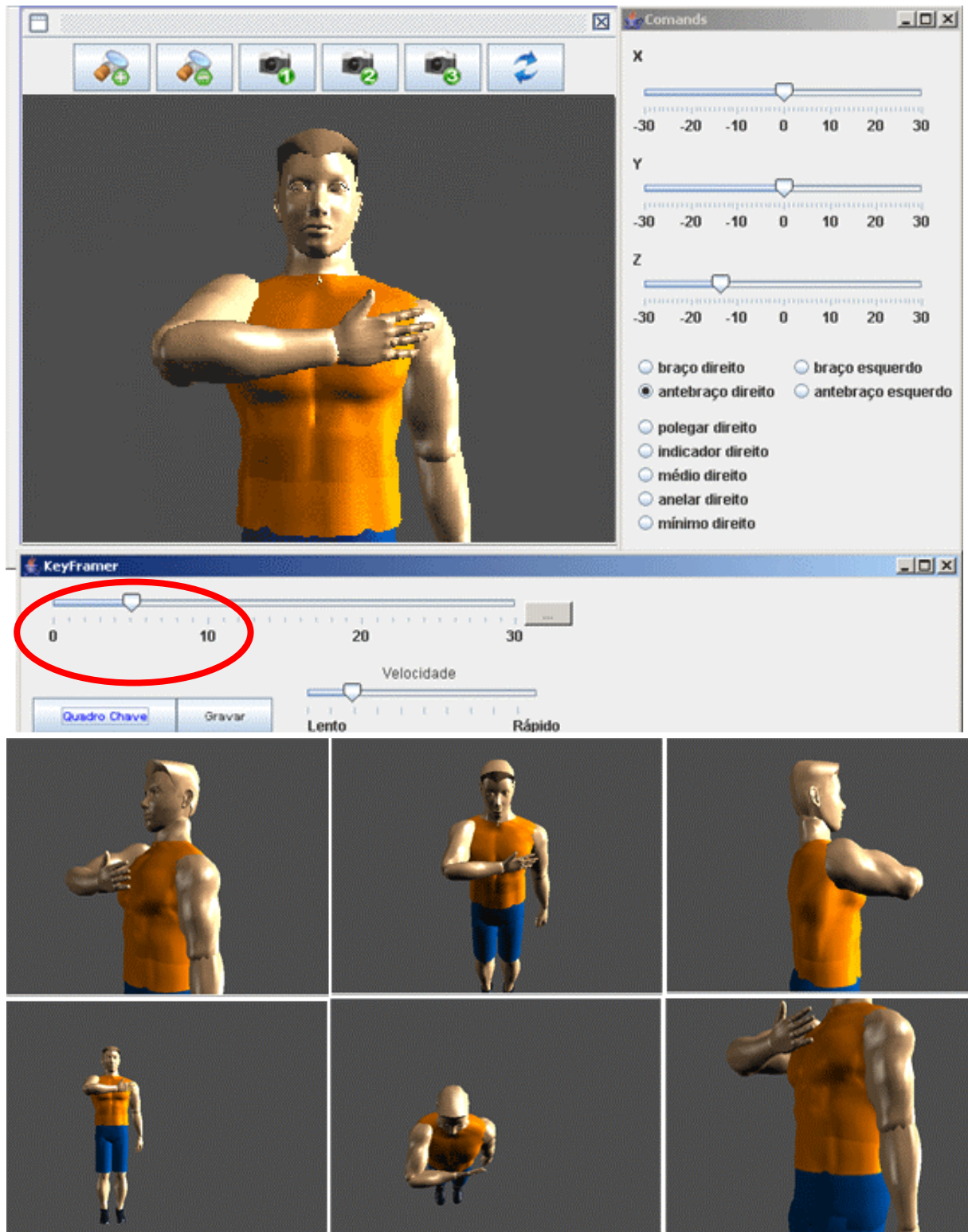


Figura 38: Detalhamento do Movimento: Cena 2

Finalizando o movimento, novamente o braço direito foi movimentado em 20° no eixo Z, sendo gravado no quadro 10 (Figura 39).

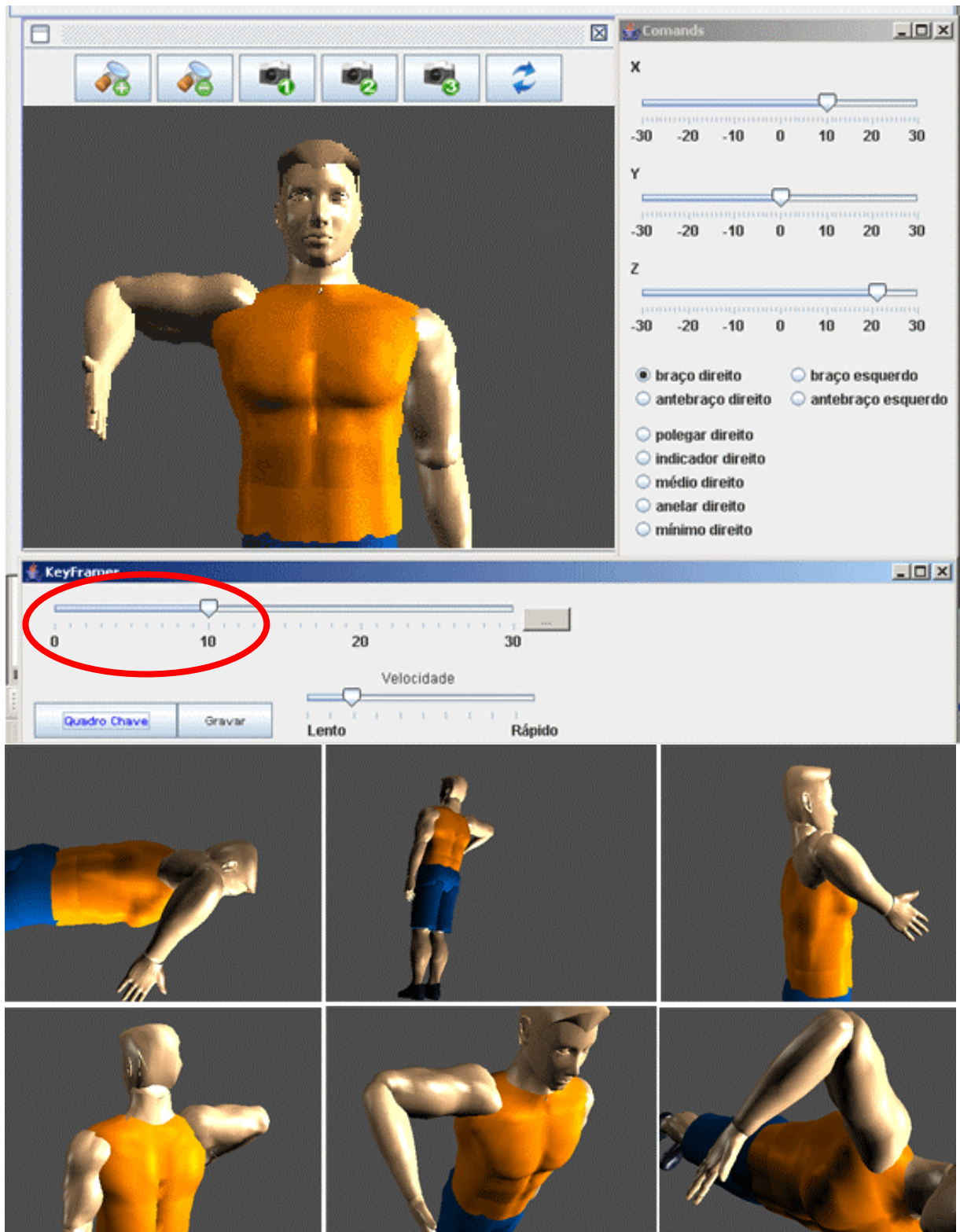


Figura 39: Detalhamento do Movimento: Cena 3

A seqüência de cenas demonstra a forma com que cada posição pode ser observada. Essas alterações são possibilitadas a partir das funções de mudança de ponto de vista, aproximado ou distanciando do Avatar, para um melhor detalhamento da cena e das posições de cada membro.

No momento da simulação descrita foi medida a taxa de quadros por segundo (*Frame Rate*). Esta medida é listada na Tabela 5.

Tabela 5: Quadros por segundo da simulação de criação de movimento

Segmento	Quadros Gerados	Tempo (segundos)	Média de Quadros por Segundo
1	1561	19,25	81,09
2	1561	19,48	80,13
3	1561	19,39	80,51
4	1561	19,67	79,36
5	1561	19,56	79,81
TOTAL	7805	97,35	80,17

Os dados apresentados podem servir como parâmetro geral na criação de movimentos já que na geração de outros movimentos essas taxas variaram na média de 75 a 85 quadros por segundo, valores que são mais do que suficientes para produzir a sensação de movimento.

A simulação também foi realizada no interpretador. Para se ter um intervalo de tempo na simulação que possibilitasse a averiguação da média de quadros durante a execução, foi realizada a interpretação de uma sentença completa, composta por várias palavras (“O soldado é amigo do governador”).

Durante a interpretação, a análise foi feita simulando o Avatar de duas formas. Na primeira execução, representada na Figura 40, apenas a parte correspondente ao tronco e membros superiores foi visualizada. A taxa média foi de 115 quadros por segundo.

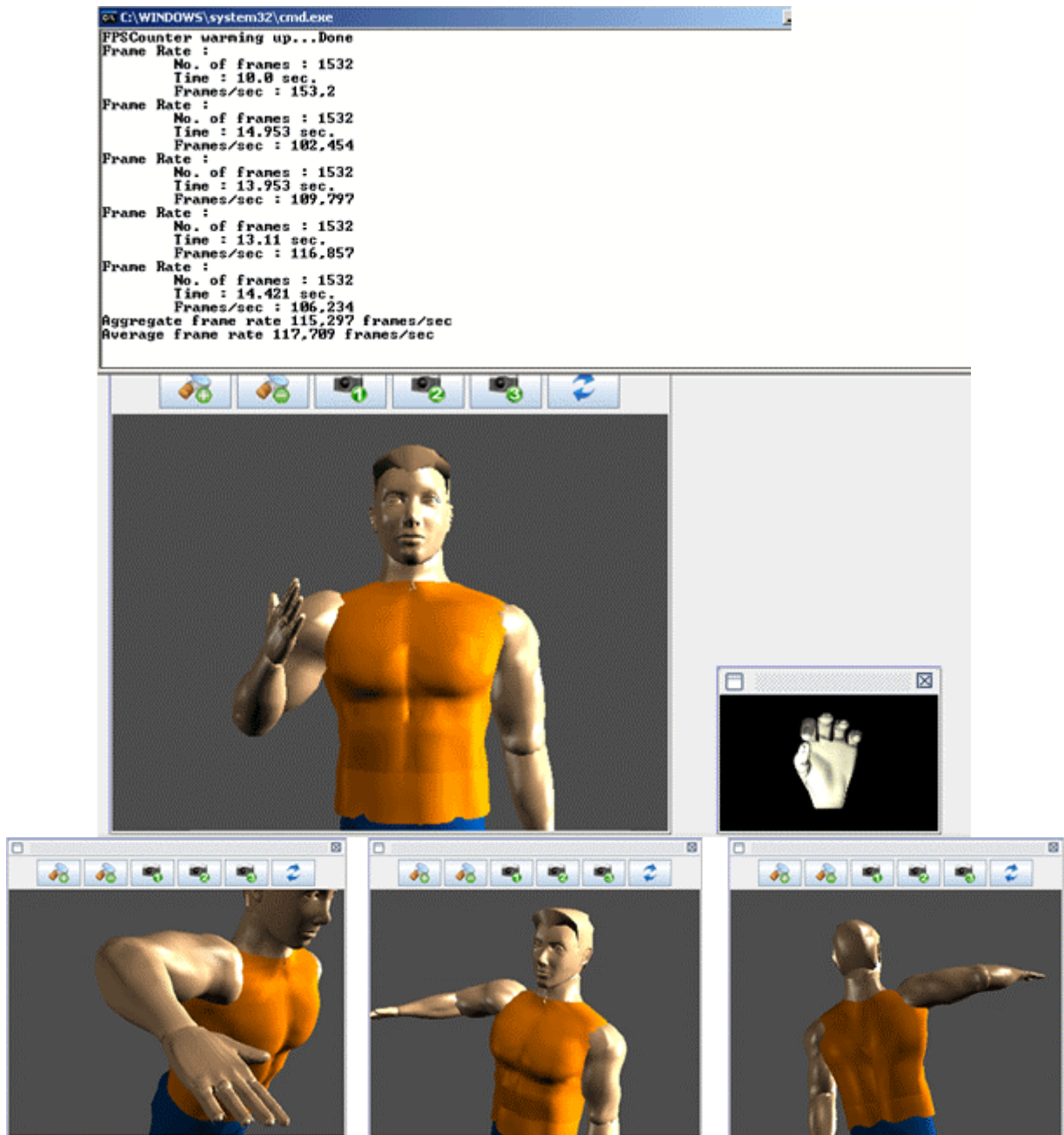


Figura 40: Simulação do interpretador membros superiores

A mesma sentença foi traduzida, mas desta vez o Avatar inteiro foi representado, esta visualização está exposta na Figura 41, neste caso a taxa de quadros por segundo caiu para 76.

A grande diferença ficou evidente na taxa de quadros por segundo. No modelo completo são mais objetos a serem renderizados, isso faz com que a quantidade de quadros

visualizados por segundo diminua significativamente, esta diferença é apresentada no gráfico da Figura 42.

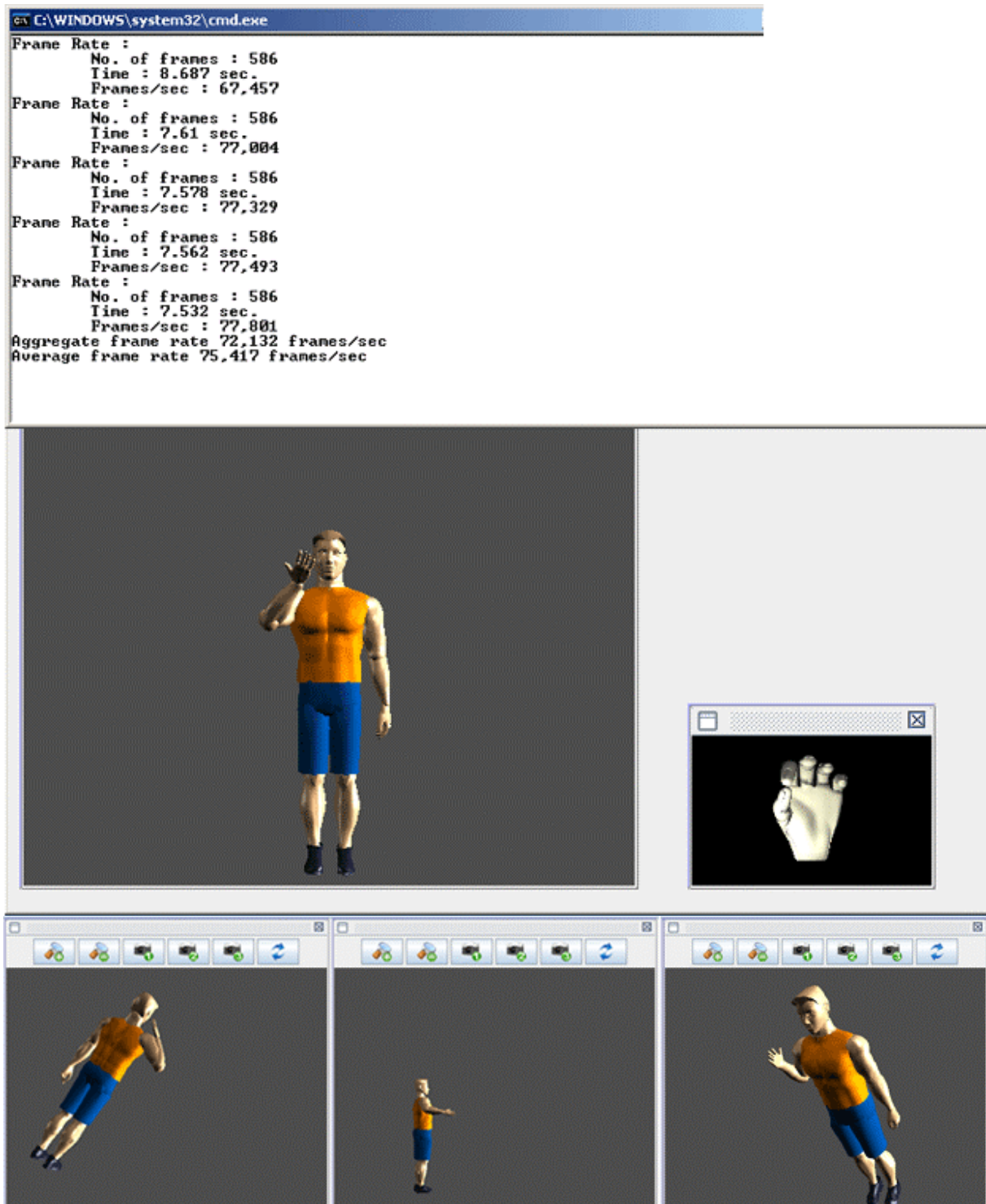


Figura 41: Simulação do interpretador, Avatar completo

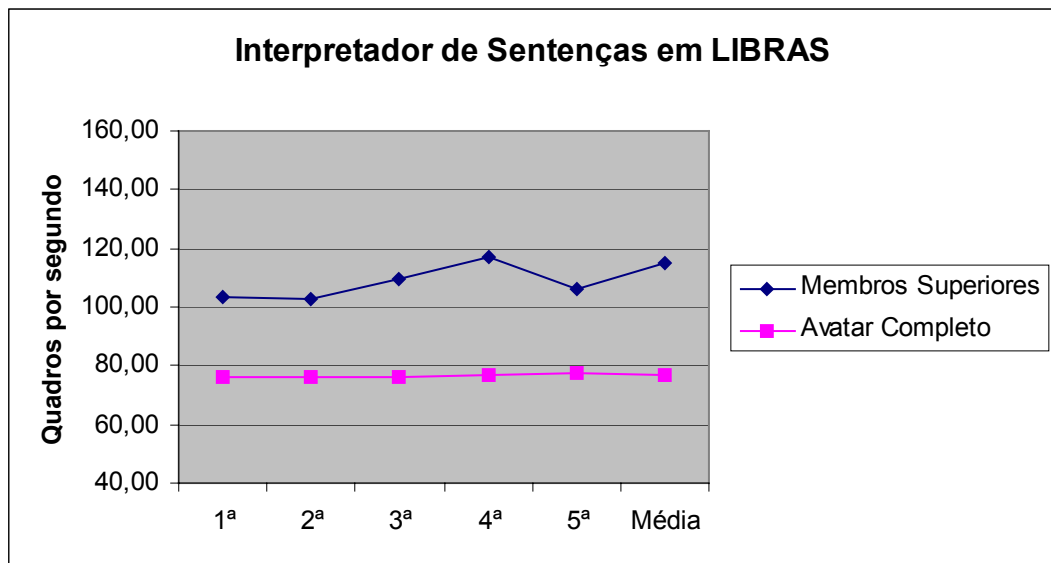


Figura 42: Gráfico comparativo do interpretador de sentenças

Analisando-se o gráfico verifica-se que mesmo a renderização do Avatar completo, tem uma taxa muito boa de 75 quadros por segundo, nas duas simulações o módulo de datilologia foi habilitado, mas os testes realizados com o mesmo desabilitado não apontaram diferença significativa no número de quadros, a média variou de 75 para 77 quadros por segundo, na simulação do Avatar completo.

A Figura 43 possibilita a visualização da representação de datilologia em diversos pontos de vista diferentes, onde a letra pode ser analisada com maior precisão, neste exemplo é apresentada a letra “A”.

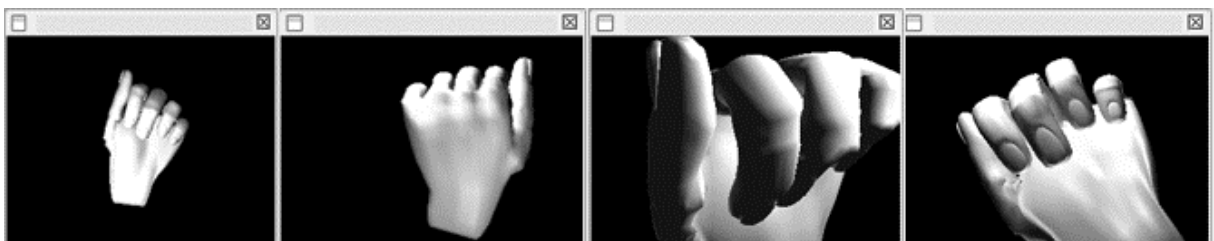


Figura 43: Representação da Datilologia

Este capítulo apresentou toda a estrutura do F-LIBRAS, partindo da interface, especificando classes, finalizando com o levantamento do desempenho da ferramenta. A seguir serão apresentadas as conclusões deste trabalho.

7. CONCLUSÕES

Este trabalho apresentou o F-LIBRAS, uma ferramenta que visa proporcionar aos usuários da LIBRAS um apoio para a difusão desta linguagem. A idéia é proporcionar tal melhoria, com a utilização de tecnologias emergentes e que possibilitaram a integração das diversas ferramentas utilizadas no desenvolvimento.

Os resultados obtidos em termos de renderização dos movimentos foram muito satisfatórios, onde a taxa de quadros por segundo está acima do limite mínimo aceitável em uma aplicação de realidade virtual. Concluindo o objetivo de unificação das diversas ferramentas em um único ambiente, que visou simplificar o acesso e utilização das mesmas.

A análise bibliográfica realizada nos primeiros capítulos deste trabalho é essencial como base na concepção desta ferramenta, apresentando as justificativas, estruturas físicas, lingüísticas e tecnológicas que proporcionaram este desenvolvimento.

7.1 A Implementação

Muito além de simplesmente transformar as diversas ferramentas em um único sistema, diversos desafios envolveram esta implementação. Entre eles destacam-se a integração das classes e o compartilhamento de algumas funções essenciais para a concepção da ferramenta. Ao longo desse processo, puderam-se observar as formas de integrar ferramentas existentes e com implementações diferentes.

Na descrição do desenvolvimento do projeto diversas imagens diagramas e códigos foram apresentados, com o intuito de ilustrar da melhor forma possível todo o processo, facilitando a visualização dos processos necessários para a criação efetiva do F-LIBRAS. Os resultados obtidos foram muito satisfatórios e os objetos responderam com rapidez e precisão às instruções enviadas pelo usuário à ferramenta.

7.2 Formatos de Armazenamento

O armazenamento em três formatos diferentes permite, não apenas armazenar as informações, mas que o formato seja rapidamente estendido e modificado para acomodar novas estruturas que venham a ser incorporadas ao modelo. Além disso, a utilização da linguagem Java, confere à biblioteca de classes desenvolvidas, grande portabilidade e versatilidade para se adaptar a novas situações e novas plataformas.

Desta forma conclui-se que esta aplicação também é uma forma de apoio a difusão da LIBRAS, na utilização de interfaces nos meios tecnológicos. Sendo meio facilitador entre pessoas com certas necessidades especiais, e as pessoas que com elas interagem.

7.3 Trabalhos Futuros

Como trabalhos futuros, novas formas para captação de movimentos poderiam ser adicionadas na criação dos mesmos, utilizando-se de dispositivos não convencionais, como rastreadores de movimentos. Para isso, os arquivos em formato TXT, seriam utilizados na implementação destes rastreadores.

Existe também a possibilidade de criação de um serviço de troca de mensagens em LIBRAS, a idéia seria que o F-LIBRAS possa servir como interface entre duas ou mais pessoas ligadas em rede ou utilizando a internet, onde mensagens seriam enviadas e representadas remotamente. Assim o formato de arquivo em XML poderia agir como base e como padrão de troca dessas mensagens.

Outra função a ser incorporada é a de visualização estereoscópica, utilizando-se de anaglifos. Para isso é necessário implementar alterações nos métodos e propriedades de aparência e material dos objetos em cena.

REFERÊNCIAS

BADLER, Norman et al. **Representing and Parameterizing Agent Behaviors**. University of Pennsylvania, 2002.

BADLER, Norman et al. **Real-Time Virtual Humans**. University of Pennsylvania, 1997.

BICHO, ALESSANDRO L. et al. **Programação Gráfica 3D com OpenGL, Open Inventor e Java 3D**. Revista Eletrônica de Iniciação Científica (REIC), Sociedade Brasileira de Computação, V. II, n. I, Brasil 2002.

BRITO, Lucinda Ferreira. **A Língua de Sinais Brasileira**. 2003. Disponível em: <http://www.ines.org.br/ines_livros/fasc7_principal.htm>. Acesso em: 30 mar. 2006.

BROGDEN, Bill; MINNICK, Chris. **Desenvolvendo E-Commerce com Java, XML e JSP**. São Paulo: Pearson Education do Brasil, 2002.

CAPOVILLA, Fernando C., et al. **Avaliando Compreensão de Sinais da Libras em Escolares Surdos do Ensino Fundamental**. Revista Interação em Psicologia, jul./dez. 2004, p. 159-169.

CONSORTIUM, Web3D. **X3D Documentation**. Disponível em: <<http://www.web3d.org/x3d/>>. Acesso em: 20 mar. 2006.

CONSORTIUM, W3C. **Extensible Markup Language (XML)**. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 16 abr. 2006.

CUSIN, Gislene Marengo. **Ambiente Virtual interpretador de sentenças em português para Língua Brasileira de Sinais**. Dissertação de Mestrado. UNIVEM. Marília, 2004.

DEITEL, H. M.; DEITEL, P.J. **Java, Como Programar**. Tradução: Edson Furmankiewicz. São Paulo: Pearson, 2006.

DEITEL, H. M.; et al. **XML, Como Programar**. Tradução: Luiz Augusto Salgado e Edson Furmankiewicz. Porto Alegre: Bookman, 2003.

DEUSDADO, Leonel Domingues. **Ensino da Língua Gestual Assistido por Personagens 3D Virtuais**. Dissertação de Mestrado. Universidade do Minho. Braga, 2002.

Dicionário Libras. Disponível em <<http://www.dicionariolibras.com.br>>. Acesso em 24. abr. 2006.

DIZEU, Liliane Correia Toscano de Brito; CAPORALI, Sueli Aparecida. **A língua de sinais constituindo o surdo como sujeito**. Educ. Soc. vol.26 n° 91 Campinas. Mai/Ago 2005.

FUSCO, Elvis. **X-LIBRAS: Um Ambiente Virtual para a Língua Brasileira de Sinais**. Dissertação de Mestrado. UNIVEM. Marília, 2004.

HOPKINS, Greg. **Tutorial: The Joy of Java 3D**. Disponível em <<http://www.java3d.org/tutorial.doc>>. Acesso em 14 abr. 2006

HUMANOID ANIMATION WORKING GROUP. **Specification for a standard Humanoid**. 2002. Disponível em <<http://www.h-anim.org/Specifications/H-Anim1.1>>. Acesso em 08 abr. 2006.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo Demográfico 2000**. Disponível em: <http://www.ibge.gov.br/home/estatistica/populacao/censo2000/tabulacao_avancada/tabela_brasil_1.1.3.shtm>

ISDALE, Jerry. **What Is Virtual Reality? A Web-Based Introduction**. 1998. Disponível em <<http://www.isdale.com/jerry/VR/WhatIsVR/frames/WhatIsVR4.1.html>>. Acesso em 08 mar. 2006.

JACOB, LILIANE JACON. **Avaliação de Técnicas de Interação Egocêntricas em Ambientes Virtuais**. Dissertação de Mestrado. Porto Alegre: UFRGS, 2002.

LAMAR, Marcus Vinicius. **Hand Gesture Recognition using T-CombNET**. Tese de Doutorado – Nagoya Institute of Technology. Nagoya, Japan, January 2001.

MACIEL, Anderson. **Modelagem de Articulações para Humanos Virtuais Baseada em Anatomia**. Dissertação de Mestrado. Porto Alegre: UFRGS, 2001.

MANSSOUR, Isabel Harb. **Introdução a Java 3D**. Disponível em <<http://www.inf.pucrs.br/~manssour/Java3D/index.html>>. Acesso em 02 abr. 2006.

MARCATO, Simone Aparecida; et Al. **Ambiente Computacional para a Aprendizagem da Língua de Sinais**. SBC 2000 – XX Congresso da Sociedade Brasileira de Computação, PUCPR – Curitiba, agosto de 2000.

MARTINS, R. B.; MACIEL, A.; NEDEL, L. P. **SkelMod-3D: Um Sistema Livre e Multiplataforma para Modelagem e Visualização de Corpos Articulados**. WSL 2001 - Workshop de Software Livre, 2001. Porto Alegre, 2001.

MINE, Mark R. **Virtual environment interaction techniques**. UNC Chapel Hill Computer Science Tech. Report TR95-018, 1995.

MIRANDA, Andréa da Silva. **Subsídios para Construção de Ambientes Virtuais de Aprendizagem Acessíveis para Deficientes Auditivos**. 2005. Disponível em <<http://www.abed.org.br/congresso2005/por/pdf/193tcc2.pdf>>. Acesso em 01 mar. 2006.

MOURA, M.C. **O surdo: caminhos para uma nova identidade**. Rio de Janeiro: Revinter, 2000.

MODESTO, Fábio Alexandre Caravieri. **Implementação de uma Ferramenta para Geração de Movimentos parametrizados para Humanos Virtuais**. Dissertação de Mestrado. UNIVEM. Marília, 2005.

NARDON, Fabiane Bizinella. **Utilizando XML para representação de informação em Saúde**. Disponível em: <<http://www.tridedalo.com.br/fabiane/publications/XML-SBISNews.pdf>> Acesso em: 20 de mar. 2006.

NEDEL, L. P. **Animação por computador: Evolução e tendências**. Organizado por: Raul Ceretta Nunes VIII Escola de Informática da SBC Sul:ed. 1 ed., Ijuí:, UNIJUÍ, 2000, p. 87-114.

OKUYAMA, Marcelo P.. **Artigo de Realidade Virtual: X3D**. UFSC - Florianópolis, 2004.

PANDZIC, Igor Sunday Et al. **VLNET: A Networked Multimedia 3D Environment with Virtual Humans**. Proc. Multi-MediaModeling 95, Singapura, Nov. 1995

PELLEGRINO, Claudia Negrão. **Estudos em Imagens Falantes: Estimulação do ensino e treinamento de Leitura Labial e Língua de Sinais - LIBRAS VIA CD-ROM em crianças surdas**. Dissertação de Mestrado. UNICAMP. Campinas, 1996.

PEREIRA, Carlos Eduardo Klimick. **Estrangeiros em sua própria cultura**. Revista Espaço. Dez. 2004

PINHO, Márcio Serolli. **Realidade Virtual como Ferramenta de Informática na Educação**. VII Simpósio Brasileiro de Informática na Educação, Belo Horizonte, 1996.

PINHO, Márcio Serolli. **Interação em Ambientes Tridimensionais**. Tutorial. Disponível em: <<http://www.inf.pucrs.br/~pinho/3DInteraction/>>. Acesso em: 20 de mar. 2006. PUCRS. Porto Alegre, 2000.

QUADROS, Ronice Muller de. **A estrutura da frase da língua brasileira de sinais**. II Congresso Nacional da Abralín. Florianópolis, 2000.

QUADROS, Ronice Muller de. **Aspectos da sintaxe e da aquisição da língua de sinais brasileira**. Letras Hoje, Porto Alegre, v.110, p. 125-146, 1997.

QUADROS, Ronice Muller de. **Diversidade e unidade nas línguas de sinais: LIBRAS e ASL**. IN: SKLIAR, Carlos. (Org.) *Atualidade da Educação Bilíngue para Surdos*. Porto Alegre, 1999, v. 2, p. 195-207.

RIBEIRO, Danielle Corrêa et al. **Estudo comparativo das API's JAX-RPC e JAXM na construção de Web Services**. Centro Universitário Luterano de Palmas. Palmas, 2005.

RICKEL, Jeff; JOHNSON, W. Lewis. **Virtual Humans for Team Training in Virtual Reality**. World Conference on AI in Education. Universidade do Sudoeste da Califórnia. Julho de 1999.

ROSA JUNIOR, Onivaldo. **LRVCHAT3D, Desenvolvimento de um Ambiente Virtual Tridimensional Multiusuário para Internet**. UFSC - Florianópolis, 2003.

SOWIZRAL, Henry A.; DEERING, Michael F. **The Java 3D API and Virtual Reality**. IEEE Computer Graphics and Applications. May/June 1999.

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**. XIV Simpósio Brasileiro de Informática na Educação, UFRJ - Rio de Janeiro, novembro de 2003.

SUN, **Introduction to Web Services**. Disponível em <<http://java.sun.com/j2ee/1.3/docs/tutorial/doc/>>. Acessado em 20 abr. 2006.

THALMANN, Daniel. **The Role of Virtual Humans in Virtual Environment Technology and Interfaces**. Computer Graphics Lab, EPFL. Lausanne, Switzerland, 2001.

THALMANN, Nadia Magnenat; KSHIRSAGAR, Sumedha. **Virtual Humans Personified**. MIRALab, CUI. Geneva, Suíça.

TORRES, J. A. R.; MACIEL, A.; NEDEL, L. P. Uma **Arquitetura para Animação de Humanos Virtuais com Raciocínio Cognitivo**. SVR 2002 - SYMPOSIUM ON VIRTUAL REALITY, 2002, Fortaleza - CE Proceedings. Fortaleza: SBC, 2002

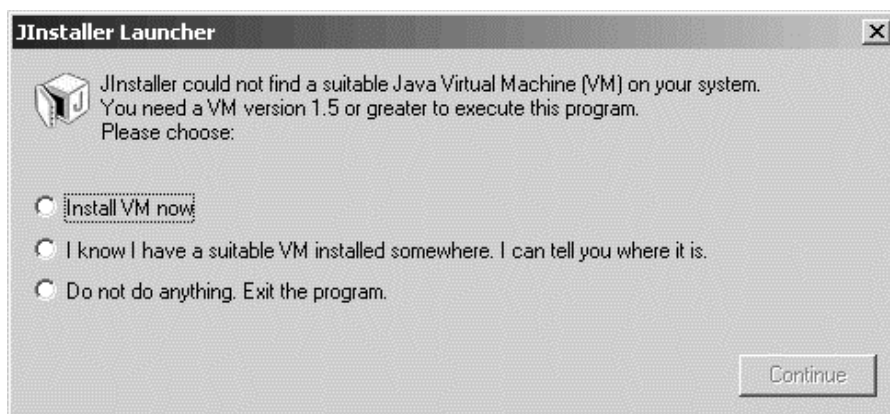
VILELA, Iara Moema Oberg. **Modelos e Aplicações de Humanos Virtuais**. I Workshop de realidade virtual e visualização científica do LAMCE(Laboratório de Métodos Computacionais em Engenharia). UFRJ - Rio de Janeiro, 2002.

VILLAMIL, Marta Becker. **Modelagem Anatômica de Corpos Articulados** Disponível em: <www.inf.unisinos.br/~cglab/equipe/martabv/ufrgs/articula.pdf>. Acesso em: 20 de mar. 2006. UFRGS.

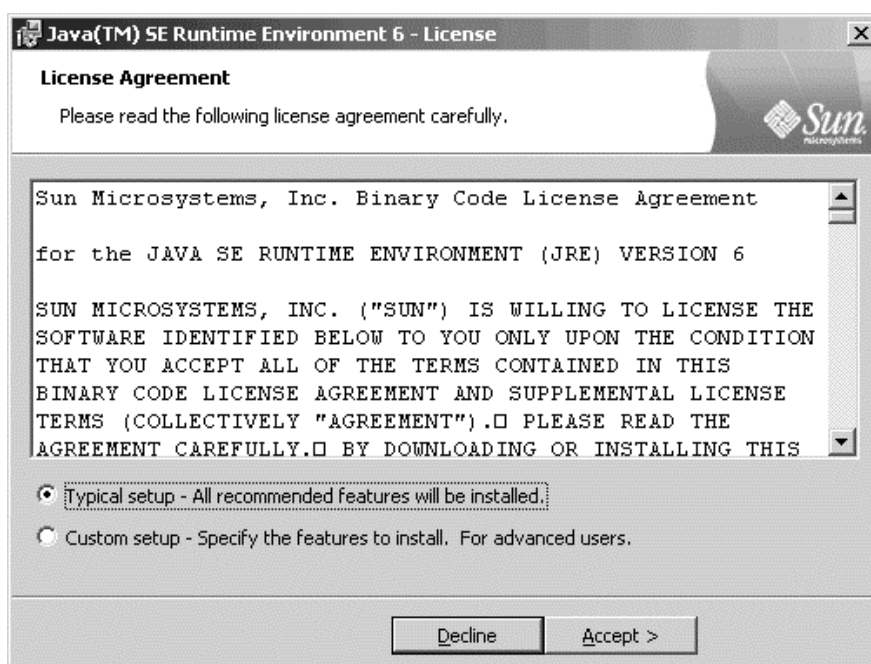
ANEXO A – INSTALAÇÃO DO F-LIBRAS

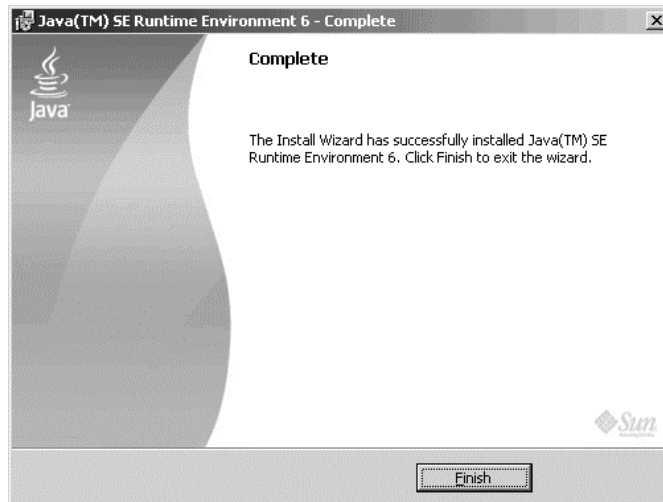
A instalação do F-LIBRAS, foi desenvolvida de forma que simplifique a ação do usuário, a seguir será apresentado o roteiro da instalação.

Inicialmente o instalador do F-LIBRAS, verifica se o usuário possui a Máquina Virtual Java (JVM) em seu sistema operacional, caso não possua, ele apresenta a opção para a instalação do mesmo, conforme a figura a seguir.

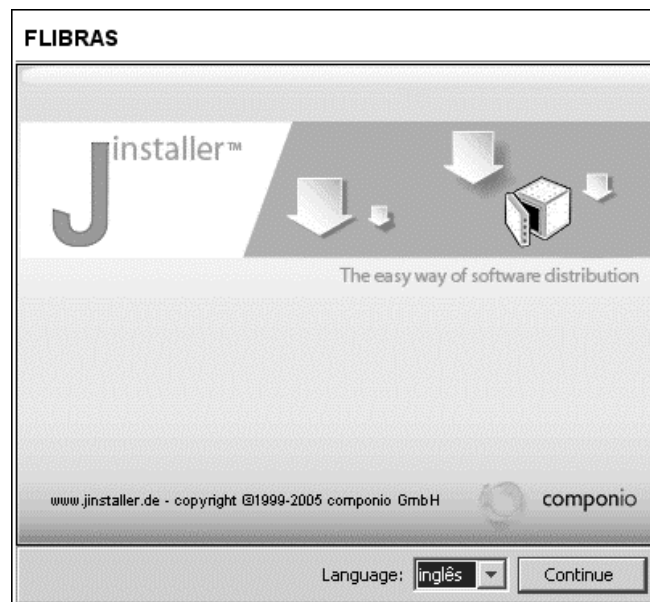


Caso escolhida a opção de instalação da JVM, automaticamente aparecerá a sua instalação.

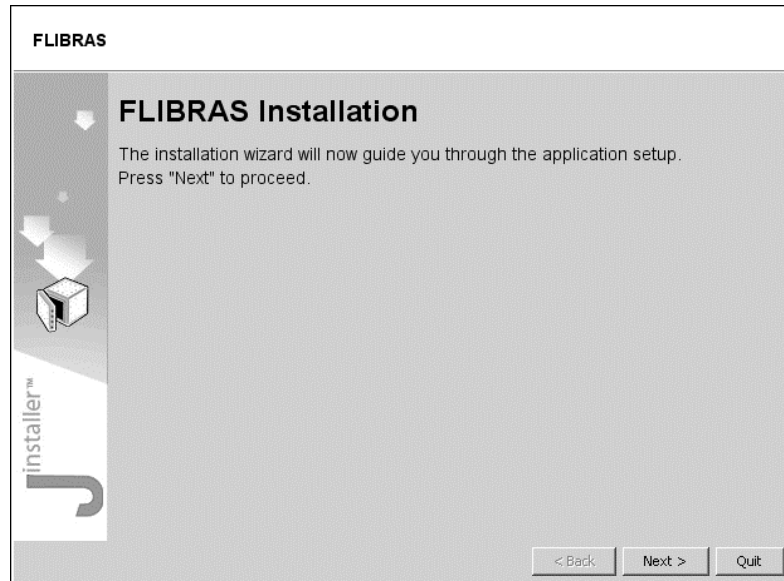




Depois de finalizado o processo de instalação da JVM, irão aparecer as opções para instalação do F-LIBRAS, conforme segue. Ressalta-se que o pacote Java3D, necessário para a execução da ferramenta, já é incluído automaticamente no pacote de instalação.

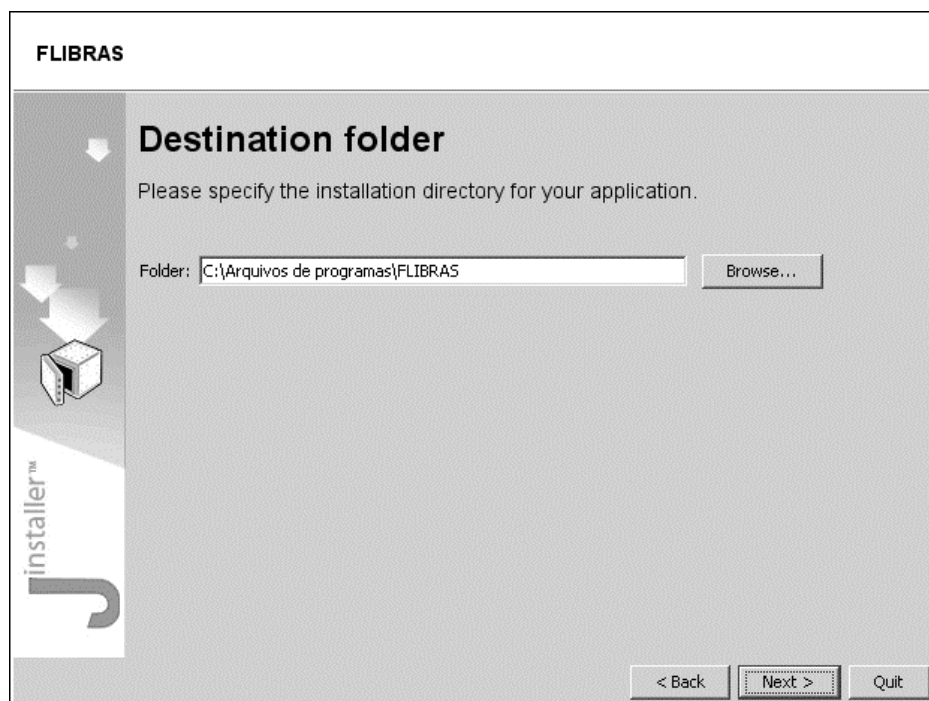


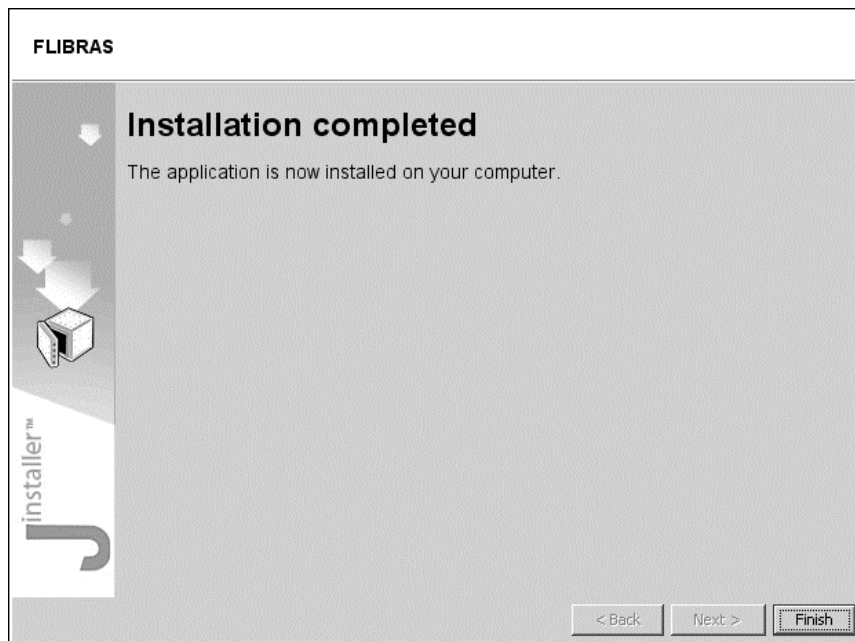
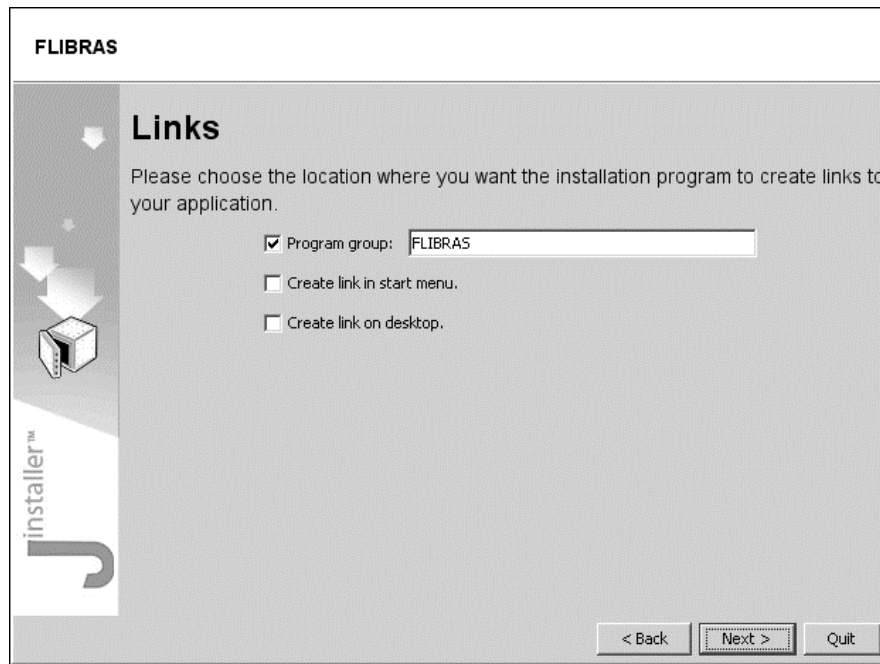
O usuário deve acionar o botão continuar, onde será apresentada a confirmação do início da Instalação, o usuário deve selecionar o botão “Next”;



O próximo passo é selecionar o diretório de instalação e as opções de criação de atalhos no computador.

Após a instalação executada com sucesso, é oferecida ao usuário a tela indicando o processo foi finalizado.





Apresentada a finalização da instalação o F-LIBRAS pode ser acessado, através dos atalhos definidos pelo usuário.